Contents

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

Contribute to the contributor guide

7/7/2021 • 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.

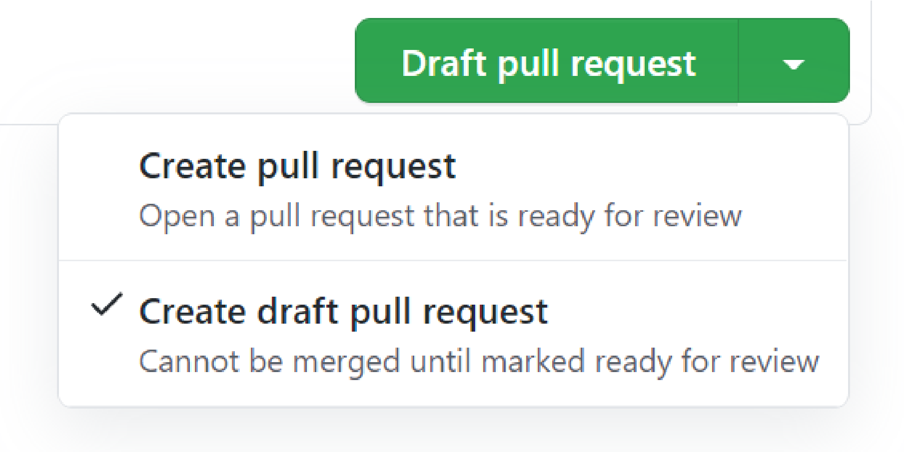
   Contact Jim Parker for an invite to the next session.

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

7/12/2021 • 22 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.

June 2021

ARTICLE	CHANGE
Best practices for contributors	New article detailing best practices for content contributors when working in Visual Studio Code and GitHub.
Visual Studio Code keyboard shortcuts	New article providing the most common keyboard shortcuts that writers use in Visual Studio Code.
Content Health report & score	New article about what Content Health is and how to use the Content Health score.
Coverage Requirements: Introductory Content	New article describing the requirements for introductory content across all business groups.
Art and multimedia gallery guidance	New article providing guidance on what icon and video libraries to use when writing Docs content.
Docs Authoring pack feature: Metadata explorer	New article about how to use the Metadata explorer feature in the Docs Authoring pack.
How to recover a lost GitHub account	New article describing options for recovering your GitHub account if you can't sign in or are locked out.
Resolve a stuck pull request	New providing guidance on how to fix a stalled pull request.
Git and GitHub glossary	New article providing reference for common terminology and concepts that writers and contributors who work with GitHub will come across.
notebook-dependent-schema-failed	New article about build validation rule.

May 2021

ARTICLE	CHANGE
Respond to inappropriate content	Learn what to do about code of conduct violations and other inappropriate content.

ARTICLE	CHANGE
Troubleshooting Git errors	New article with instructions on how to resolve Git-specific errors when updating content.
FAQ best practices	New article providing best practices when writing FAQ content.
Best practices for contributing to docs.microsoft.com	New article outlining important considerations for successfully publishing your content.
Write "best practices" content for using your service	New article providing process guidelines for creating articles about best practices for a service or feature.
SEO onboarding for contributors	New article providing resources to help you incorporate basic SEO into your daily writing practices.
SEO: Checklist for web-friendly docs	New checklist article with guidance for writing web-friendly doc pages.
Get started contributing to Visual Studio documentation	New article providing guidance on how Microsoft employees can contribute to Visual Studio documentation.
Visual Studio docs guidance	New article providing guidance for Microsoft employees who contribute to Visual Studio documentation in visualstudio-docs-pr.
Principles for publishing preview content for Visual Studio and VS Code on docs.microsoft.com	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.
Banners on docs.microsoft.com	New article about how to request promotional banners to be run on docs.microsoft.com pages.
Write a Partner Center announcement	New article about how to write announcements that are published to Partner Center.
Getting data on API documentation	Article moved to the docs contributor guide that provides information on how to get data from API documentation content.
Docs.microsoft.com archive, retire, and removal policy	Article moved to the docs contributor guide that provides an overview for the policies on archiving, retiring, and removing articles on docs.microsoft.com.
Information architecture on Docs	Article moved to the docs contributor guide that introduces the core information architecture components used to structure content on docs.microsoft.com.
Archived content	Article moved to the docs contributor guide that provides guidelines and historical context for archived, previous-version content.
Converting SMS content to markdown	Article moved to the docs contributor guide that provides instructions for converting SMS content to markdown.

ARTICLE	CHANGE
invalid-value	New article about build validation rule.
languages-missing	New article about build validation rule.
path-achievement-not-found	New article about build validation rule.
pivot-definition-not-found	New article about build validation rule.

April 2021

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

ARTICLE	CHANGE
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.
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.

ARTICLE	CHANGE
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.
table-syntax-invalid	New article about build validation rule.

January 2021

ARTICLE	CHANGE
Write how-to guides	New article providing guidance for writing how-to guides.
Write concepts	New article providing guidance for writing concept articles.
Template for how-to guides	New template for writing how-to guides.
Template for concept articles	New template for writing concept articles.

ARTICLE	CHANGE
Contributing to docs.microsoft.com	New article clarifying options for contributing to docs.microsoft.com.
Resolving "stale" GitHub issues	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.
Insert and update content from a Jupyter notebook	New article about the Docs Authoring Pack extension for Jupyter Notebooks.
forbidden-content	New article about build validation rule.
forbidden-link	New article about build validation rule.
module-badge-missing	New article about build validation rule.
module-uid-missing	New article about build validation rule.
token-not-found	New article about build validation rule.

December 2020

ARTICLE	CHANGE
Contributing content to docs.microsoft.com	New article providing an overview and brief description of each high-level phase in the content creation process.
Planning your content	New article listing the tasks of the planning phase of content contribution.
Designing your content	New article listing the tasks of the design phase of content contribution.
Writing your content	New article listing the tasks of the writing phase of content contribution.
Publishing your content	New article listing the tasks of the publishing phase of content contribution.
Maintaining content	New article listing the tasks of the maintenance phase of content contribution.
Create FAQ content	New article about using the structured YAML template to present FAQ content.
Repository and publishing information for the Partner Center business	New article listing Partner Center repositories.
Azure Architecture Center browser	New article about the Azure Architecture Center (AAC) browser.

ARTICLE	CHANGE
New contributor checklist for new content	New article covering the tasks required for publishing a new content set.
Update Azure Databricks content	New article with information about updating Azure Databricks content.
link-redirected	New article about build validation rule.
file-not-found	New article about build validation rule.
sandbox-not-mapped	New article about build validation rule.
redirected-file-not-removed	New article about build validation rule.
quiz-multiple-answers	New article about build validation rule.

November 2020

ARTICLE	CHANGE
UserTesting	New article about the UserTesting research tool.
Publishing for the air-gapped cloud (AGC)	New article about publishing content in air-gapped locations.
Use applies to and appropriate roles in Partner Center	New article about requirements for Partner Center content.
Connect to CGA Kusto data clusters	New article about requesting access to the Kusto data clusters and connecting to them using client applications.
Community Kusto queries	New article listing community-submitted Kusto queries that other contributors have found useful.
Contributor Insights	New article about the Contributor Insights Dashboard
General tagging guidelines	New conceptual article about metadata tagging guidelines
module-child-not-found	New article about build validation rule.
path-achievement-not-trophy	New article about build validation rule.
path-child-not-found	New article about build validation rule.
path-child-not-module	New article about build validation rule.
path-modules-missing	New article about build validation rule.
path-trophy-missing	New article about build validation rule.
path-uid-missing	New article about build validation rule.

ARTICLE	CHANGE
module-achievement-not-badge	New article about build validation rule.
task-type-invalid	New article about build validation rule.
unit-uid-missing	New article about build validation rule.
duplicate-pivot-groups	New article about build validation rule.
duplicate-pivot-ids	New article about build validation rule.
pivot-group-not-found	New article about build validation rule.
pivot-id-not-found	New article about build validation rule.
pivot-id-unused	New article about build validation rule.
pivot-definition-conflict	New article about build validation rule.

October 2020

ARTICLE	CHANGE
Edit an article in a private repo	New article about editing content in a private repository using the web browser.
Breadcrumbs guidelines	New conceptual article about breadcrumbs.
Header navigation governance	New conceptual article about header navigation.
Get page-level metrics in your browser: Microsoft Docs Metrics (PREVIEW)	New article about using MDM to get high-level metrics without leaving the page.
OBSOLETE: What is Content Data Advocacy Group (CDAG)?	New article about the Content Data Advocacy Group.
Taxonomies for Docs	New reference article listing existing metadata taxonomies for Docs.
List of available contributor guides	New reference article listing known contributor guides.
Error documentation not available	New article about build validation rule.
invalid-tab-group	New article about build validation rule.
redirect-url-invalid	New article about build validation rule.

September 2020

ARTICLE	CHANGE
sensitive-language	New article about build validation rule.
PR only validation rules	New article about PR only functionality for build validation.
Docs Local Validation	New article explaining how to perform build validations against your local Docs conceptual or Learn repository.
Monitoring content for Azure services	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.
rule-override-invalid	New article about build validation rule.
Update documentation for a servicing update	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
DocsImageSorter	New article about a prototype tool that uses Azure Cognitive Services to identify issues with images.
preserve-view-not-set	New article about build validation rule.

July 2020

ARTICLE	CHANGE
Unified Content Model FAQ	New article with answers to common questions about the unified content model.
Terminology and inclusive language	Article revised to provide guidelines to address inclusive language in content.
Major update to video guidance	The video guidance has been brought current and consolidated from 4 articles to 1.

June 2020

ARTICLE	CHANGE
Is docs.microsoft.com the right fit for you?	Article updated to remove compliance content from the list of content that does not belong in a docs.microsoft.com article or repository.

ARTICLE	CHANGE
Planning for and managing costs for an Azure service Azure template for a manage costs conceptual article	New article and template for writing a conceptual article about planning for and managing costs for an Azure service.
Metadata attributes	Article updated with end-user-help added as an ms-topic value.

May 2020

ARTICLE	CHANGE
Unified content model	A new article about the Unified Content Model.

April 2020

ARTICLE	CHANGE
Update Azure docs hub page	A new article about how to update the Azure docs hub page.

January 2020

ARTICLE	CHANGE
Write quickstarts	Article updated and clarified.
Base template for quickstart articles	Template updated and clarified to align with guidance updates.
TwitterApp	New article about using TwitterApp for simplified use and management of public-facing twitter accounts.
Create or update a hub page	Update article to add "sample" to itemType list.
Create or update a landing page	Update article to add "sample" to itemType list.
Metadata attributes	Update article to add 'kb-support' as new allowlist valid value for 'ms.topic'
Docs Markdown reference	Markdown guidance consolidated.
Meriwether diff reporting tool guide	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.
How to include code in docs	Article updated with new code snippet markdown extension.

December 2019

ARTICLE	CHANGE
Content and Learning office hours	A new article that lists a current schedule of office hours and guidance for hosting your own office hours.

November 2019

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

September 2019

ARTICLE	CHANGE
Revised table of contents	The contributor guide TOC was reorganized to make content easier to find.
Link to articles from the user interface	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
Create or update a hub page	A new article about a page that collects related docs.
Create or update a landing page	Update to a page for a single docset.

May 2019

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

April 2019

ARTICLE	CHANGE
Create a landing page	Updates to the Learn module section of this article.

March 2019

ARTICLE	CHANGE
Markdown cheat sheet	.pdf file removed because the guidance was outdated.
File name and path guidelines	Article updated with content from *File names and locations for Azure technical articles* which was removed from the guide to eliminate duplication.
Site search enabled for the contributor guide	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.
Identify preview content	A new article that explains how to identify content for private and public preview.

February 2019

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

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

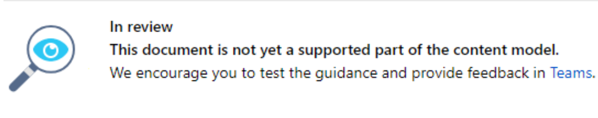
The cadence for the **What's New** page now follows the regular Content & Learning Sprint dates.

ARTICLE	CHANGE
Localization guidance for legal articles	Update process for localizing legal articles.
How to break up large PRs	New simpler instructions for taking a large PR and breaking it up into smaller, easier to review PRs.
How to request a technical focused review	Formal rollout for the Technial Focused review (TFR) process.
GitHub issues and PR reports	Instructions on generating and using the GitHub issue and PR aging reports.
New Azure Dev Ops guidance	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
When to create a docs troubleshooting article	New guidance for when to create troubleshooting content.
Global template for overview articles Global template for quickstart articles Global template for tutorial articles	New templates added for writing standard content types.
In review status	New review status established for contributor guide articles that propose new processes. 
Contribute to the contributor guide	Updated to include info about adding the "In review" label to identify new processes that need testing and feedback.

July 2018

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

June 2018

ARTICLE	CHANGE
How to enable the documentation feedback control	Moved article from OPS documentation to the contributor guide.
Contribute to the contributor guide	New article documenting the process for making contributions to the contributor guide.
Writing content for Content & Learning	Updated to emphasize procedures via the CLI for quickstarts, tutorials, and how-to guides in milestone two.

ARTICLE	CHANGE
Write a quickstart	Updated to clarify that wait time for things like provisioning don't count against the 10-minute limit.
Accessibility guidelines	New article documenting accessibility guidelines for content contributors.
Alt Text	Moved from the style guide to the contributor guide.
TOC creation and management	New article with details about how to create a TOC.
File name and path guidelines	New article with guidance for file and folder names that are included in the public URL on docs.microsoft.com
The 5 writing principles	Updated with guidance to support adding time estimates in procedures when something will take longer than expected.

May 2018

ARTICLE	CHANGE
Rename a product or service in content	New article recommends approach for renaming a product or service in technical documentation.
Install content authoring tools	Replaced/updated Gauntlet information with Docs Authoring Pack info.
Request approval for an exception or experiment	New article documenting the process for requesting an exception to the content model criteria, or an experiment to test variations.
Writing principles checklist	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.
How to include code in docs	Article updated to address content overlap. Update to clarify when brackets are used.
Developer quickstart guidance	Articles updated to address content overlap.
Add in-line code snippets	Deleted as part of the code content overlap resolution.
Subscriptions - Azure subscriptions for writers	Updated Azure internal registration and subscription (AIRS) guidance and added more vendor info.

April 2018

ARTICLE	CHANGE
Contributor's guide	Updated TOC and landing page.

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

March 2018

ARTICLE	CHANGE
Use release branches	Updated with guidance for using release branches

ARTICLE	CHANGE
Write an overview	Updated with guidance to include the customer intent statement as a comment in the metadata section.
Write a quickstart	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.
Write a tutorial	Updated with guidance to include the customer intent statement as a comment in the metadata section. Clarified distinctions between tutorials and how-to articles.
TOC Guidance for Docs Content	Updated the definition of quickstarts and how-to guides.
Create a screenshot	Updated with guidance for using animated GIFs.
Create and publish animated GIFs	Added new article about using animated GIFs in documentation
Text formatting guidelines	Added new article about formatting text (when to use bold, italics, code, etc.)
GitHub for content reviews	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
White papers	Added new article about hosting white papers.
Set up a local Git repository	Updated guidance.
Create a landing page	Updated to align with existing guidance.
Write a quickstart	Revised article to be less Azure specific.
Write a tutorial	Revised article to be less Azure specific.
Request a CFR	Updated with guidance for creating a customer intent statement.
Sovereign clouds	Added new article about sovereign clouds.
Style and grammar	Added new article about style and grammar.
Link blog to article	Added new article about linking blogs to articles.

January 2018

ARTICLE	CHANGE
Install content authoring tools	Updated guidance for installing Git and VS Code.
Create a landing page	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
Write an overview	Updated with guidance about adding "overview" to Page title, Meta description, H1, Intro paragraph
Write a quickstart	Updated with guidance about adding "quickstart" to Page title, Meta description, H1, Intro paragraph
Write a tutorial	Updated with guidance about adding "tutorial" to Page title, Meta description, H1, Intro paragraph

Options for contributing to docs.microsoft.com

7/7/2021 • 3 minutes to read

This article helps you choose the best path for contributing to content in docs.microsoft.com. 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.

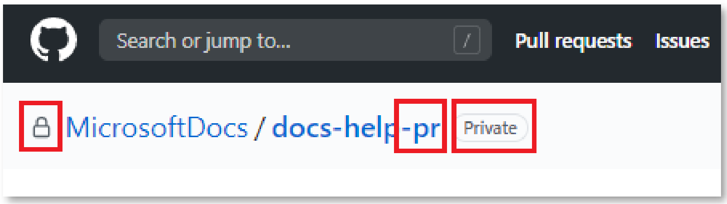
You can also append *-pr* to the repo name in the GitHub URL. **Note,** however, the *-pr* naming convention is standard in *most* cases; not all Microsoft Docs repos follow this convention.

> **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	https://docs.microsoft.com/azure/firewall/overview
Published Docs (linking to *-pr* repo)	https://**review.docs.microsoft.com**/azure/firewall/overview
Public GitHub repo	https://github.com/MicrosoftDocs/**azure-docs**/blob/master/articles/firewall/overview.md
Private GitHub repo	https://github.com/MicrosoftDocs/**azure-docs-pr**/blob/master/articles/firewall/overview.md

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

7/8/2021 • 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 set up GitHub once. It requires performing a few simple tasks, including joining GitHub and linking it to your Microsoft account, joining the MicrosoftDocs organization, and enabling two-factor authentication. Complete those steps here:

☑	TASKS
☐	Set up a GitHub account

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
☐	Make quick edits in a private Microsoft Docs repository using the GitHub WebUI.
☐	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 SpineEdit browser extension. For more information, see SpineEdit.

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
☐	Install content authoring tools	One time setup.
☐	Fork and clone the repository	Required for each repository you want to edit.
☐	Edit an article	Required for each repository you want to edit.
☐	Author an article	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
☐	Create and edit pull request
☐	Address any validation errors. For more information, see Docs Build validation overview.
☐	Add reviewers by @mention their alias in a comment. Use the GitHub contributor alias search.
☐	Interact with Pull request comments and suggestions

☑	TASKS
☐	Review and sign off

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

7/13/2021 • 5 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 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.

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, 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.
7. Verify your organizations: in a browser, go to your own directory listing with this URI, https://repos.opensource.microsoft.com/profile?aspect=organizations.

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 and 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.
```

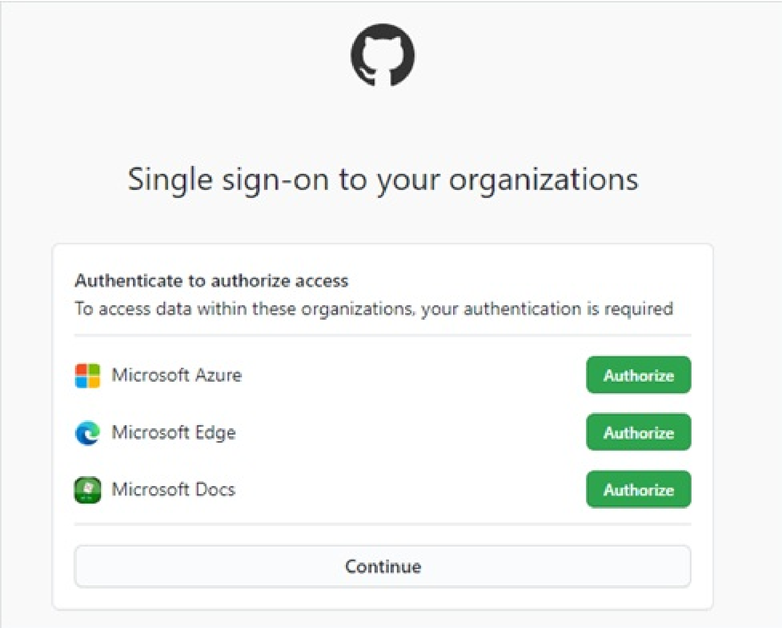
> **WARNING**
>
> Here's an alert box.

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.**

Troubleshooting

If you have any problems with these procedures, use the Docs Support General channel in Teams to search for an answer or post about your issue.

Next steps

Make minor updates Install authoring tools for major updates

Edit an article in a private repo

7/7/2021 • 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

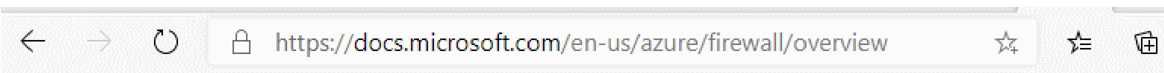
You need a GitHub account to contribute to Microsoft Docs.

- 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 you 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.

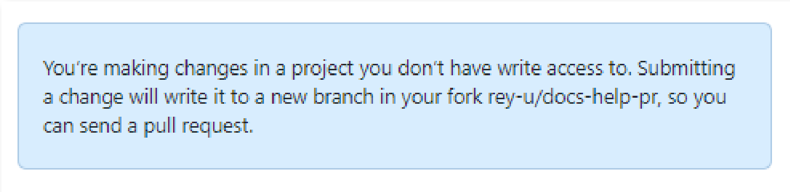
2. Go to the page of the Microsoft Docs article you want to update. To work on the private repo version of the article, add "*review.*" to the beginning of the URL. For more information on private repos, see *Finding private repos*, below.

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 the 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.

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 **#sign-off** in the comment stream, the **ready-to-merge** 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 *isn't* 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 **#hold-off** in a PR comment to remove the **ready-to-merge** label. In the private repo, **#hold-off** assigns the **do-not-merge** 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
#please-open	Authors can reopen a closed pull request.	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

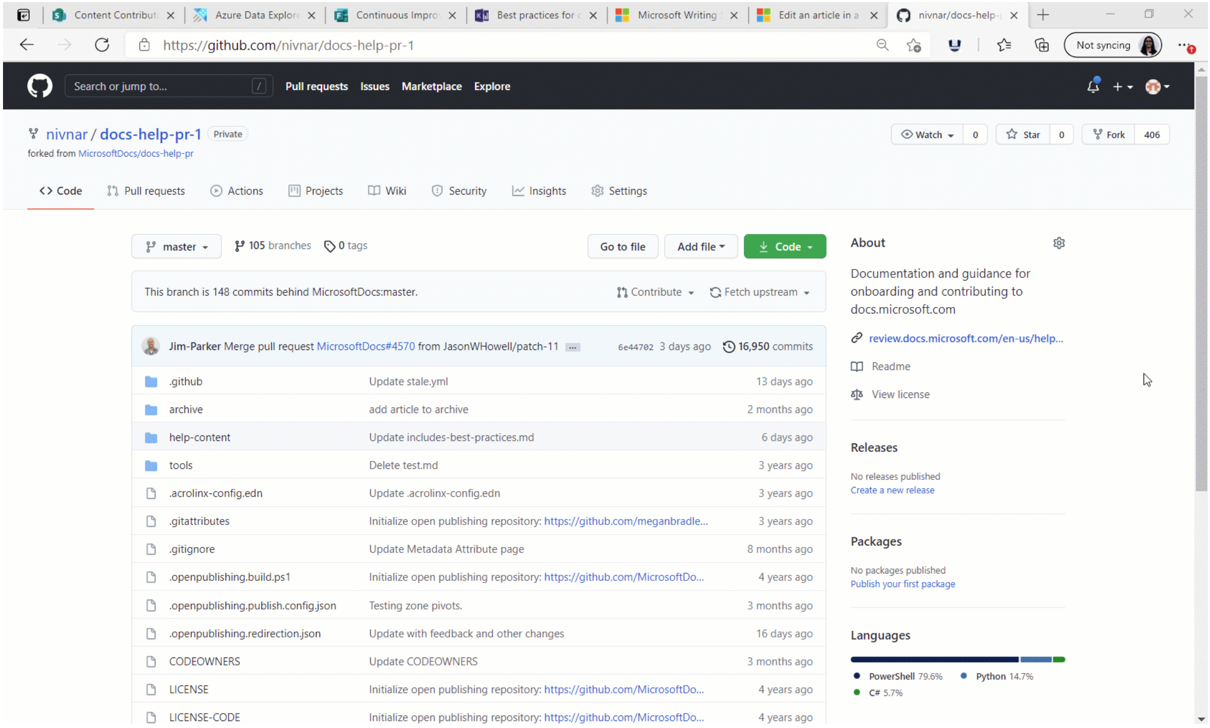
If your fork is not up-to-date with the upstream, you might receive one of the below errors when you edit article and submit a PR:

- 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



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.

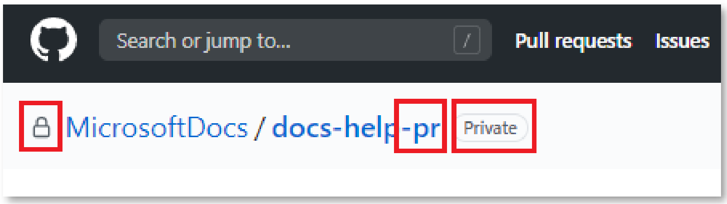
> **TIP**
> For a step-by-step guide on editing in **public** repos, see Edit an article on Docs.

Example: Public vs private repo URLs

SOURCE	URL
Published Docs	https://docs.microsoft.com/azure/firewall/overview
Published Docs (linking to *-pr* repo)	https://**review.docs.microsoft.com**/azure/firewall/overview
Public GitHub repo	https://github.com/MicrosoftDocs/**azure-docs**/blob/master/articles/firewall/overview.md

SOURCE	URL
Private GitHub repo	https://github.com/MicrosoftDocs/**azure-docs-pr**/blob/master/articles/firewall/overview.md

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

7/7/2021 • 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

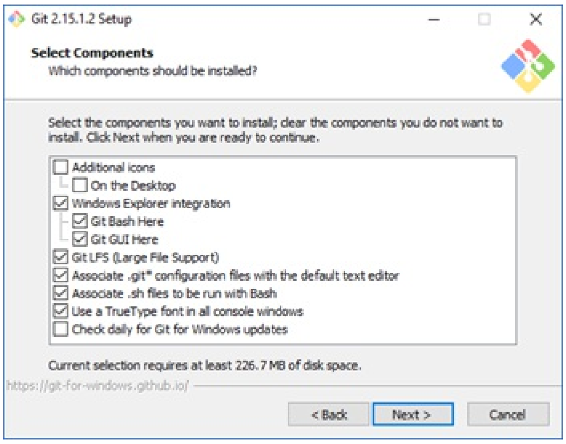
There are essentials tools for both Windows and macOS, that you should install.

Install Git client tools for Windows

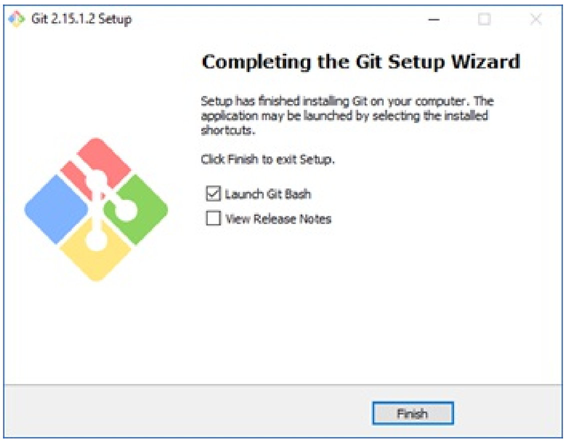
> **NOTE**
>
> - If you prefer a graphical user interface (GUI) over a command-line interface (CLI), Visual Studio Code, Software Freedom Conservancy's available GUI Clients page, and GitHub's GitHub Desktop offer that functionality.

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**.

> **WARNING**
>
> **Git on Windows: Enable long path names**. By default, Git for Windows disables support for long filepaths, which prevents any file with a destination path longer than 255 characters from being cloned. Enable long filepaths to avoid this issue by running the following command **as an administrator**: `git config --global core.longpaths true`

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`)

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.

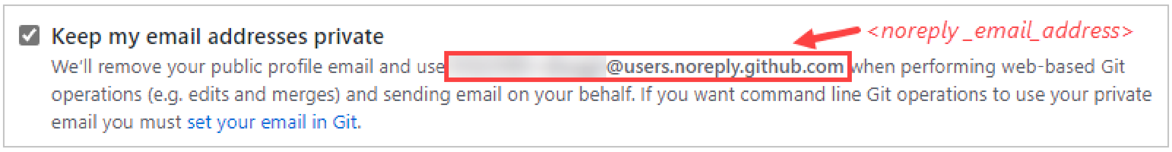
1. Sign in to GitHub using your GitHub account.

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`).



6. 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>"
```

7. 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.

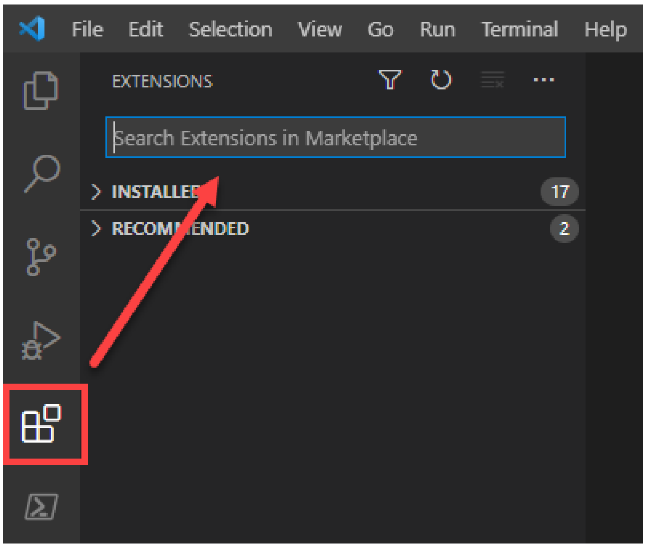
- Run the following command to install Visual Studio Code:

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

Install Visual Studio Code extensions

To install the extensions:

1. Start Visual Studio Code.
2. Select the square **Extensions** icon on the left nav. The **Extensions: Marketplace** pane appears.
3. 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.

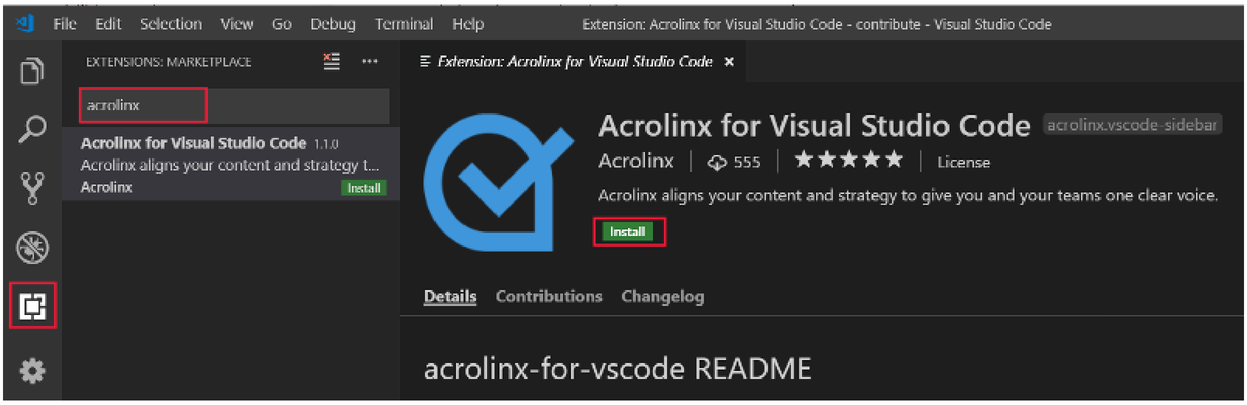
To use the Docs Authoring Pack functionality, press Alt+M in Visual Studio Code. To configure a toolbar to show the functions available, edit the Visual Studio Code settings (Control+comma), and add user setting `"markdown.showToolbar": true`.

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

7/7/2021 • 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.

> **TIP**
> Refer external contributors to this public contributor guide page for similar information. Keep these two pages in sync as things change.

Prerequisites

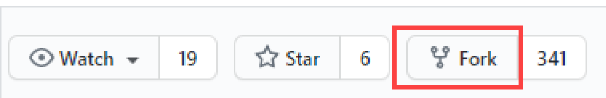
- Identify the GitHub repository that stores your team's documentation
- To set up a GitHub account
- To install authoring tools

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.

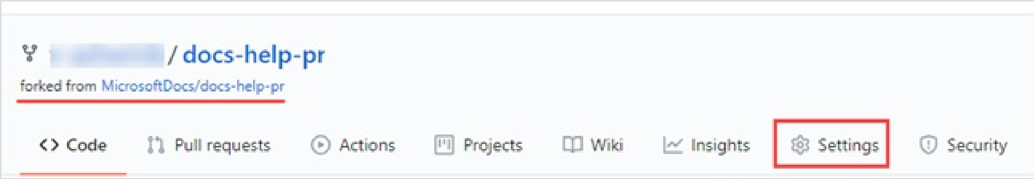


4. In the response to **Where should we fork** , select your account.

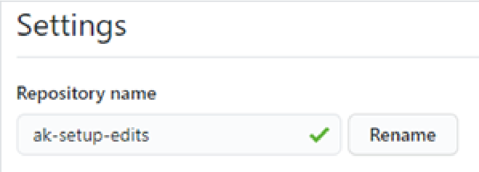
Rename your fork

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**.



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



Clone the repository

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

> **WARNING**
>
> 1. **DO NOT clone local repos into OneDrive-synced folders**. When choosing the folder into which you'll clone the repo, avoid folders that sync with OneDrive (for example, your **Documents** folder). OneDrive can cause unintended merge conflicts when you merge your changes back to the main branch. Instead, it's best to create a short filepath on your `C:\` drive (for example, *C:\GitHub*).
> 2. **Git on Windows: Enable long path names**. By default, Git for Windows disables support for long filepaths, which prevents any file with a destination path longer than 255 characters from being cloned. Enable long filepaths to avoid this issue by running the following command **as an administrator**: `git config --global core.longpaths true`

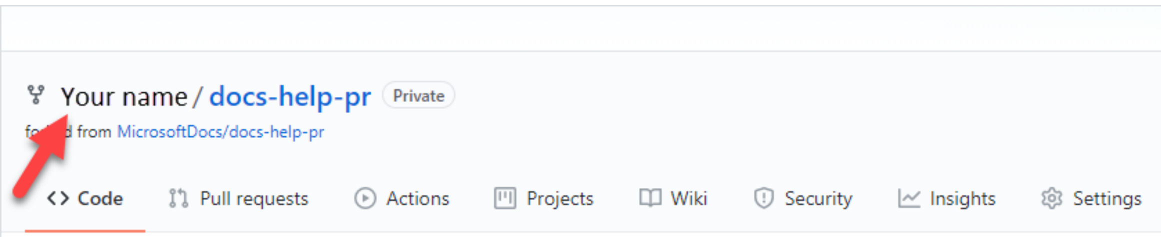
1. Create a `GitHub` folder in your local machine. For example, *C:\GitHub*. (**Don't** clone to a OneDrive synced folder.)

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.

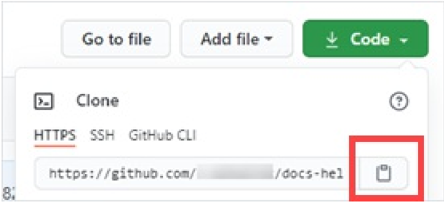
4. Select the repository you want to clone. The repository page of *Your* fork appears.

   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`.

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.



> **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.

2. From top nav menu, select **View** and then select **Terminal**. The terminal view appears in the bottom of the Visual Studio 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.

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 your 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=PMNZGKZDFNZ2GSYLML52HS4DFVNHWC5
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

7/7/2021 • 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 branch-name
   ```

4. Alternatively, you can create a working branch in your GitHub fork, run the git checkout command with a **-u** . For example, `git checkout -b branch-name -u`. 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 branch-name`.

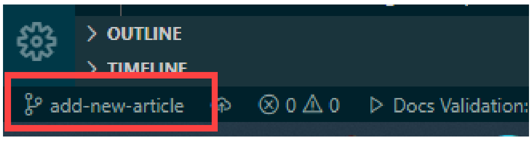
```
git checkout -b branch-name -u
```

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.



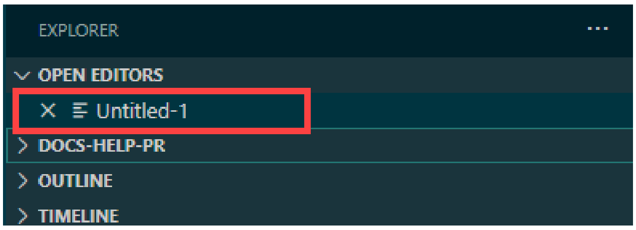
4. If you need to switch to a working branch, run the following command within the VS Code Terminal View:

```
git checkout nameofbranch
```

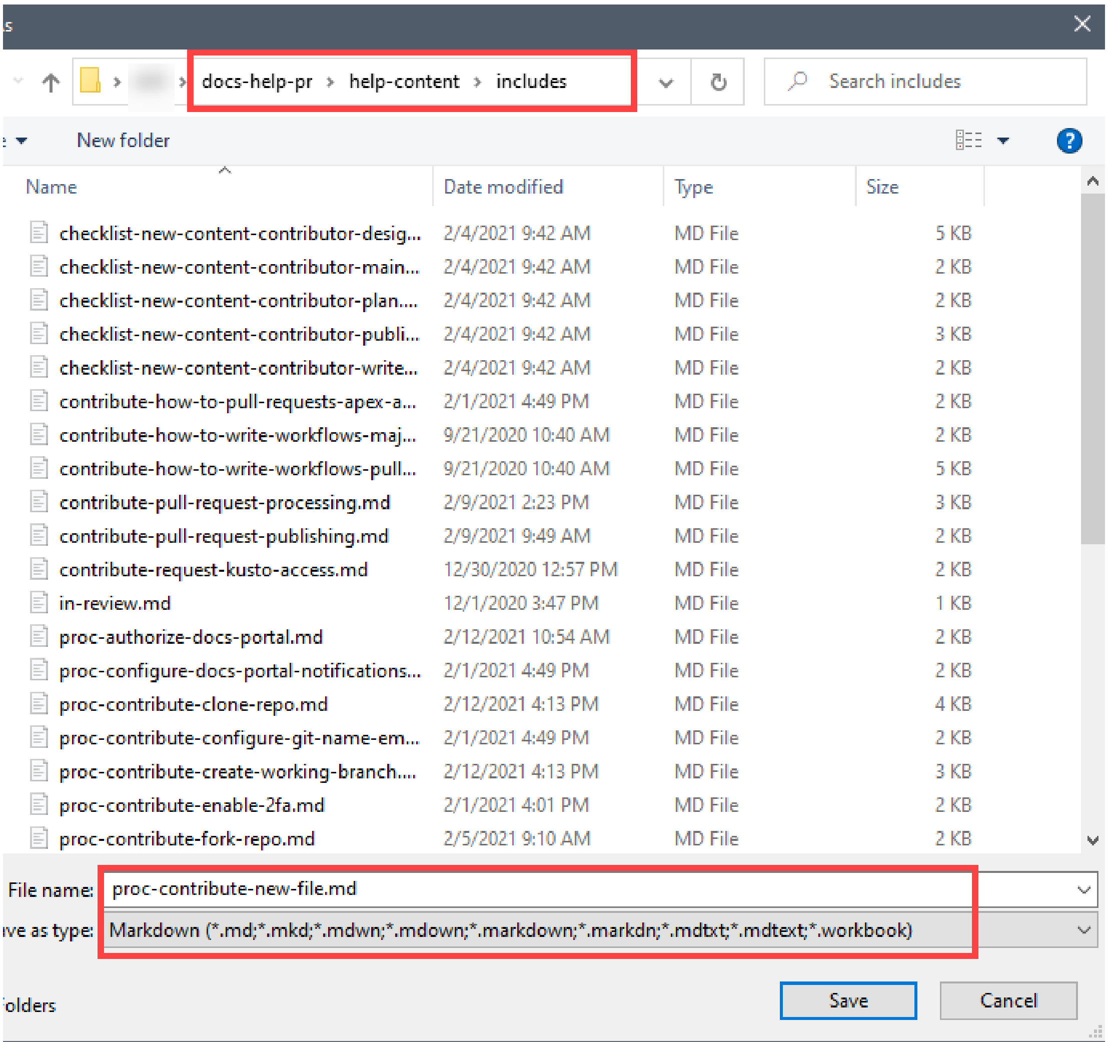
5. Ensure your working branch is current with the master branch, run the following command within the VS Code Terminal View:

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

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.



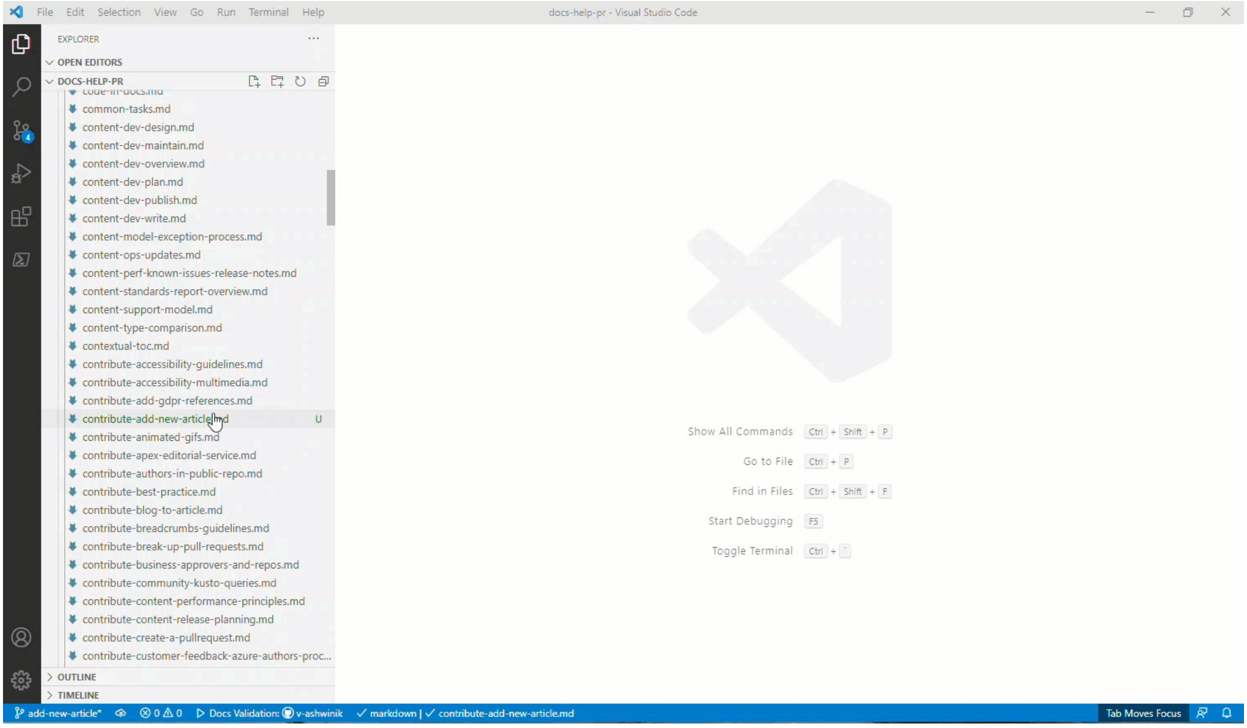
7. To name the file, select **File** and then choose **Save**. The Save As dialog box appears. Navigate to the right location for this file before saving.
8. VS Code defaults to .txt file type, change to .md for markdown. Your new file is now created.



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

7/7/2021 • 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 branch-name
   ```

4. Alternatively, you can create a working branch in your GitHub fork, run the git checkout command with a **-u** . For example, `git checkout -b branch-name -u`. 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 branch-name`.

```
git checkout -b branch-name -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.

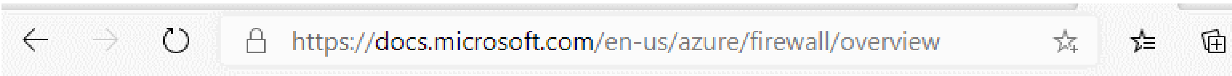
Option 2 - If you're unable to locate the file, visit the article in https://docs.microsoft.com and select the **Edit** link. The relative folder location in the GitHub repository shows in the URL. Here's an example **Edit** link URL:

```
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" />
```

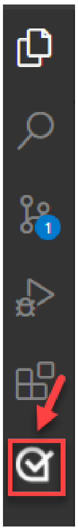
Edit the file

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, select the **Open preview** icon. The preview tab appears next to the article tab.
2. If you have the VS Code legacy toolbar enabled, select 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
```

3. After staging the file, run the `git commit` command followed by commit message to save changes on your local cloned repository.

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

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.

Push the commit

To push the change run the following command:

```
git push origin branch-name
```

If you used *-u* option to create your working branch, just run the following command to push the commit:

```
git push
```

Use the VS Code UI to commit and push

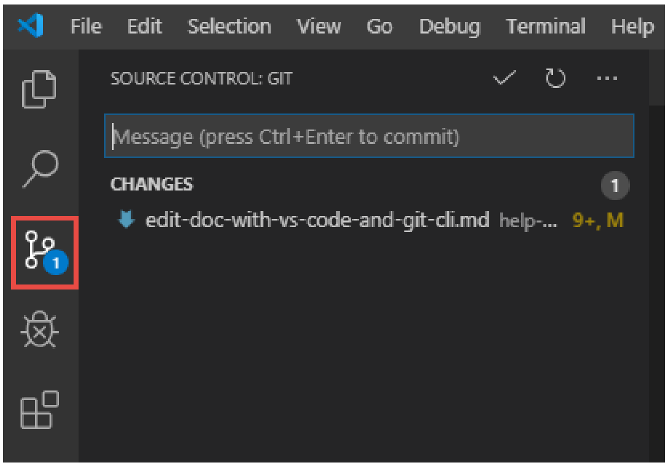
You can also create and push a commit by using Git integration features of Visual Studio Code. Remember to continue your work on the same machine. If you switch machines, you won't have the latest version of the working branch to continue with.

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).

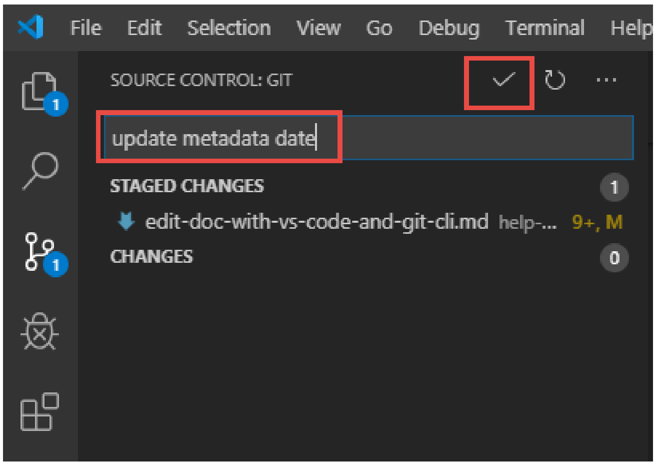
4. In the VS Code left side bar, select the fork tool to see the **Source Control: Git** sidebar.



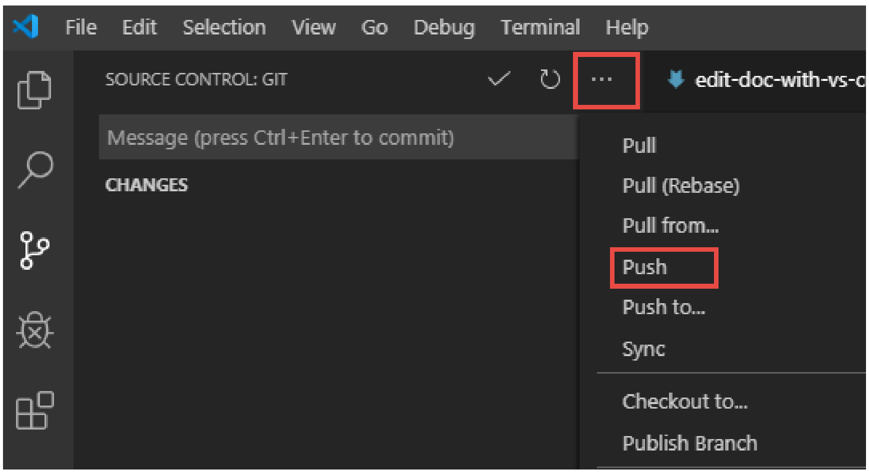
5. The file you edited appears under **Changes** with an **M** on the right to indicate it is modified. New files would have **U** (untracked) and deleted files would have **D**.

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**.

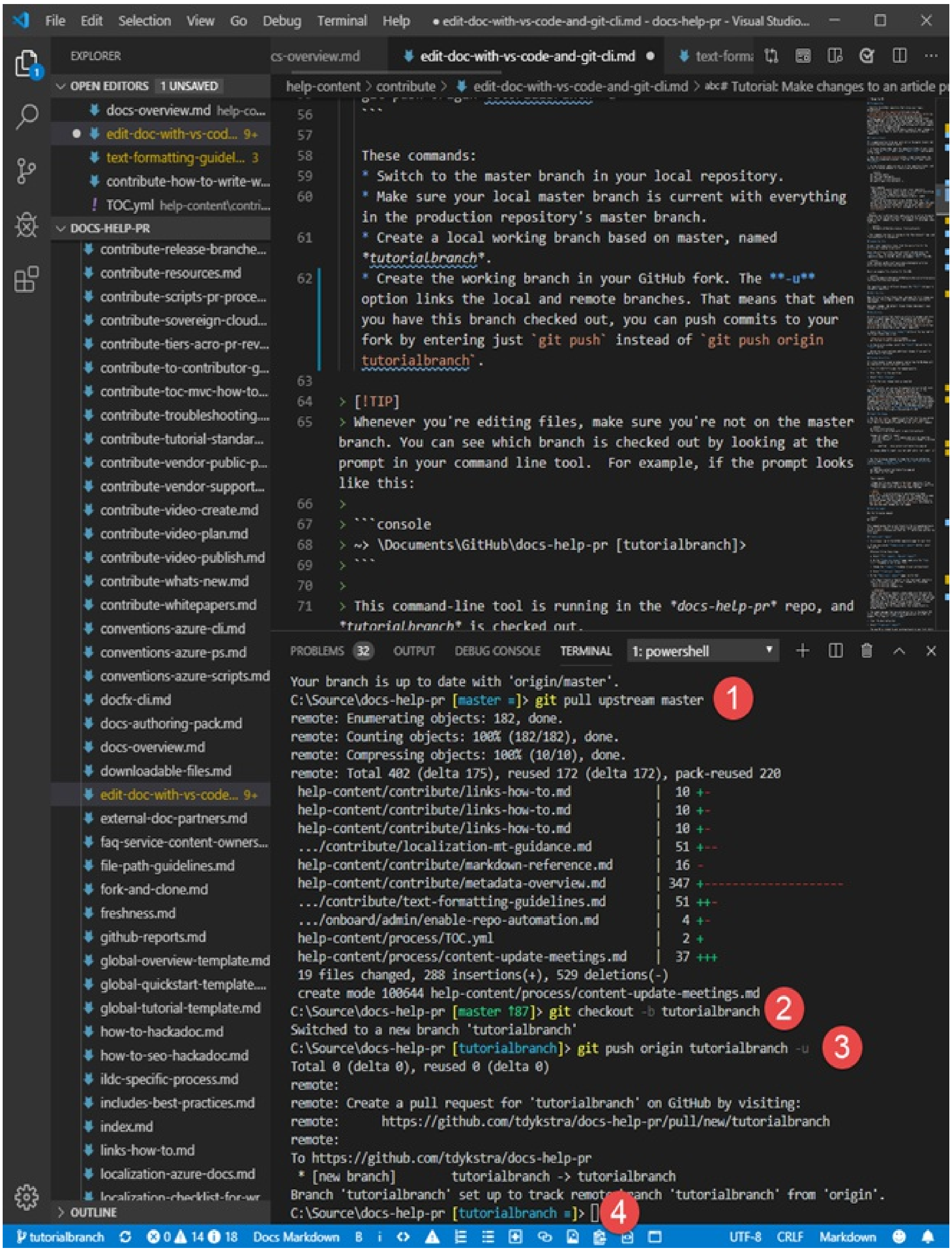


9. Look at the pull request page in GitHub, and you see that there are now two commits. Because you added the commit to the branch specified in the PR, GitHub automatically incorporates it into the PR. In the publishing system, the change triggers a build, and the PR changes are validated and staged again.

10. On the GitHub pull request page, select the **Files** tab. The *diff* display shows both of your changes.

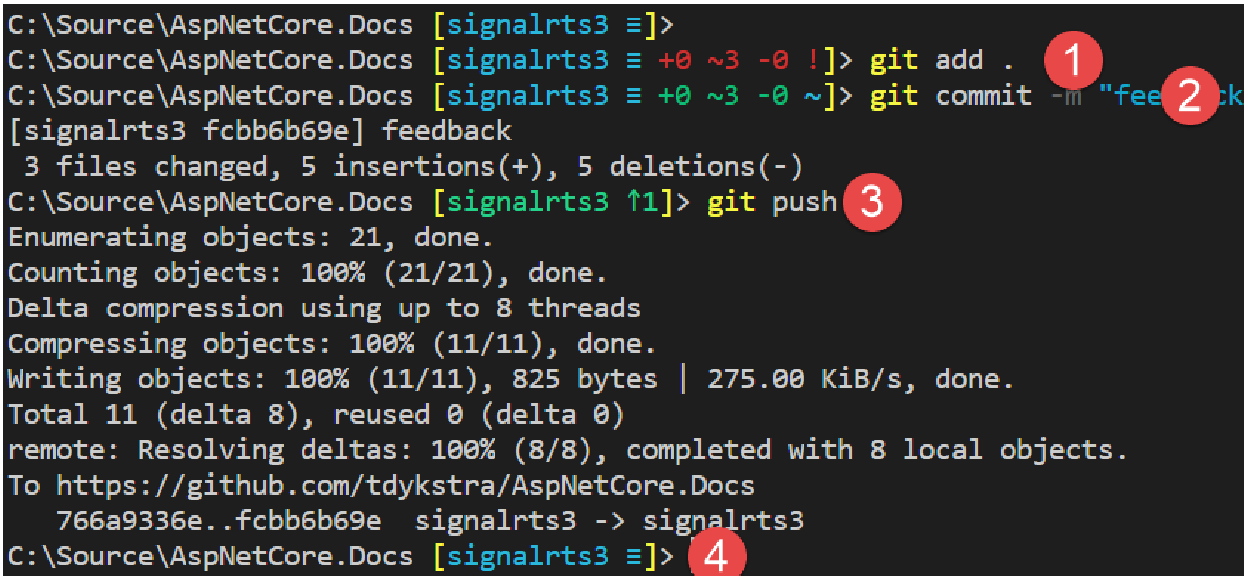
Use PoshGit to review Git status

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:



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.

Next Steps

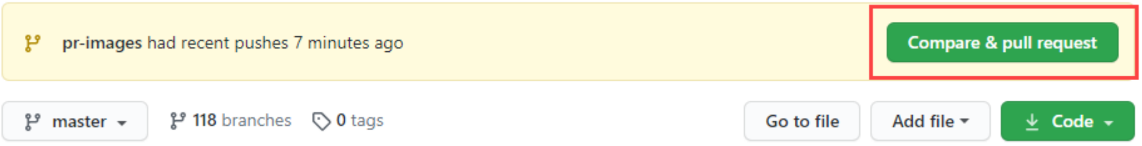
Create your pull request

Create a pull request

7/7/2021 • 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:

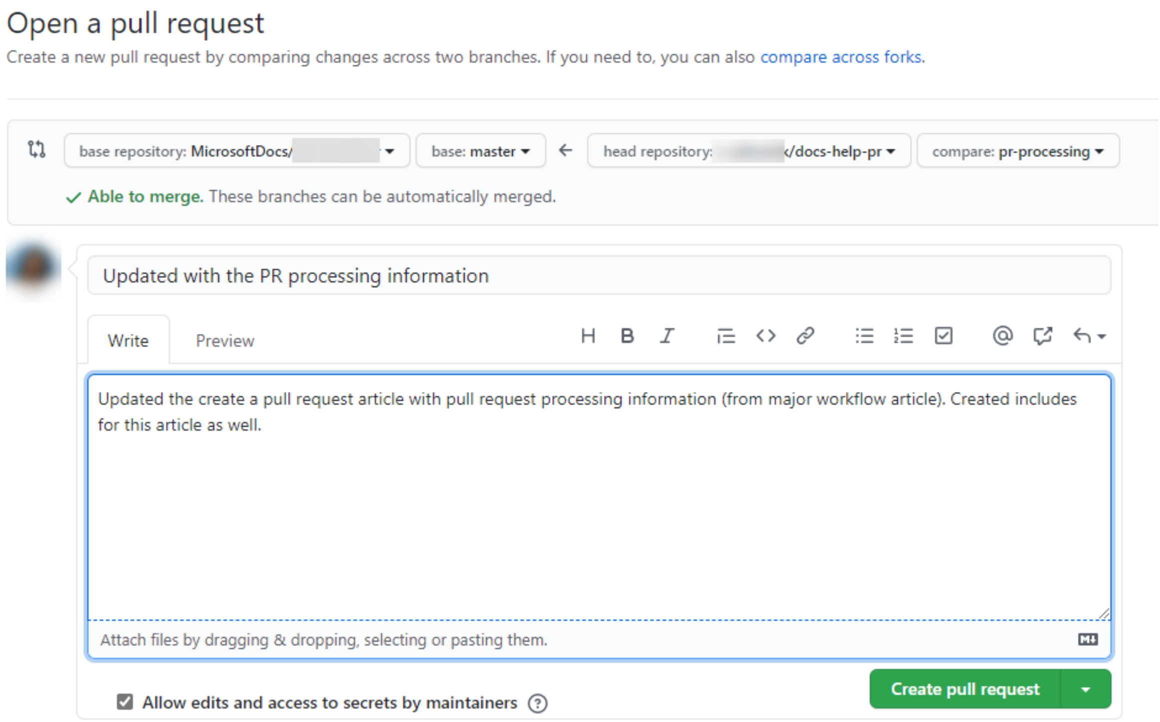
a. Select **Pull requests > New pull request**.
b. On the **Comparing changes** page, make sure the **head fork:** drop-down is set to your fork.
c. Change the **compare:** drop-down to your working branch.
d. Select **Create pull request**.

> **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.



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.
- **Automerge**: 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; for example: *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

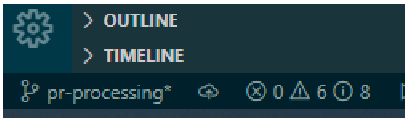
After all PR processing is completed, you should review the results (PR comments, preview URLs, and validation) to determine if more changes are required before you sign-off for merging. If a PR reviewer has reviewed your pull request, they can also provide feedback through comments if there are outstanding 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 **#sign-off** in the comment stream, the **ready-to-merge** 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 *isn't* 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 **#hold-off** in a PR comment to remove the **ready-to-merge** label. In the private repo, **#hold-off** assigns the **do-not-merge** label.	Public and private
#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
#please-open	Authors can reopen a closed pull request.	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

7/8/2021 • 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
- ms.reviewer: 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 ms.reviewer 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.

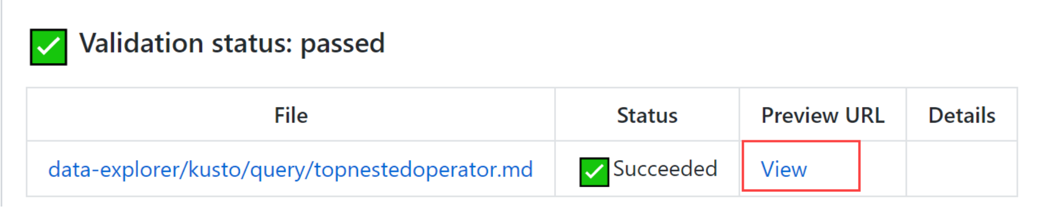
> **NOTE**
>
> To enable Acrolinx in Visual Studio Code you must open the entire data-explorer folder in your workspace, and not a single file.

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



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

Comment responsibly

Docs supports HTML comments, but you should avoid using them. Docs carries HTML comments through to the published HTML that goes public to customers. While HTML comments are invisible to the reader's eye, they are exposed in the HTML underneath.

If you must use a comment, make sure it does not contain private or sensitive information.

An HTML comment looks like this:

```
<!-- Here's my comment -->
```

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 best 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)

6/29/2021 • 2 minutes to read

Listed below are the resources available to contributors to docs.microsoft.com for support and guidance on using out 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 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 **visible to customers**	Sitehelp
An internal tool isn't working correctly	Sitehelp
specific problem with your branch or repo	Contributor Guide
specific problem with your branch & you **already** searched the contributors guide	Docs Support
Builds are very slow	Docs Support
Request a docs feature	Docs Feature
Need a new repo	Publish-on-docs
Need to set a repo live	Publish-on-docs
Need to learn how to publish on docs	Docs Training
GitHub isn't working correctly	Docs Support
Need to set up public contributions	Publish-on-docs
Need to change base URL or priority	Publish-on-docs
Want help with improving content performance	Docs Training
Need a release branch	Publish-on-docs
Need to set a repo live the first time	Publish-on-docs
Need help with generating redirections	Publish-on-docs
Need help with setting up conceptual versioning	Publish-on-docs
Need to make a config change	Publish-on-docs

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

6/28/2021 • 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	Configure your GitHub account
Make a small change (for example, fix a typo)	Click the **Edit** button on a published article; change the *repo name* by editing the URL. *For example*, from *docs-help* to docs-help-pr or from *azure-docs* to azure-docs-pr and submit fixes. The SpineEdit browser extension can make this process even easier.
Locate an author of a specific article	Click the **Edit** button at the top of a published article, the article will open within the GitHub user interface; click on the **Raw** button; find the GitHub username of the author from the `author` value or the domain alias from the `ms.author` 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. *For example*, docs-help-pr or azure-docs-pr from your fork*; *be sure to @ mention the doc author in the PR*
Create a new branch	While on the main/default branch such as docs-help-pr or azure-docs-pr, run: `git checkout -B <BRANCH_NAME>`. This will create a local branch on your local fork of the remote repository. Note that remote branches originating from `azure-docs-pr` are release branches, and are managed.
Sync your branch with latest changes	While on your branch, run: `git pull upstream master`. For this command to work you need to have a remote named *upstream* that points to the private repo such as, docs-help-pr or azure-docs-pr.
Copy branch to GitHub for a PR	Run: `git push origin <BRANCH_NAME>`. Once the branch is copied to GitHub, you can navigate to the web app to create a PR.
Request to publish a PR	Add `#sign-off` to a comment in your PR
Find out when my merged update will be live	Check the PR review and publishing schedule for your repo. You can also read about publishing from a public repo.
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. *For example*, docs-help-pr or azure-docs-pr

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 makes 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.

- Create a video

Overview of the content-development process

7/7/2021 • 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 in 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

6/28/2021 • 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
☐	Identify the lead PM, sponsor, and subject matter experts for the new content.
☐	Identify the expected timeline for the release.
☐	Identify if a repository for your need exists. If not, request the creation of a content repo.

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

Next steps

Design your content

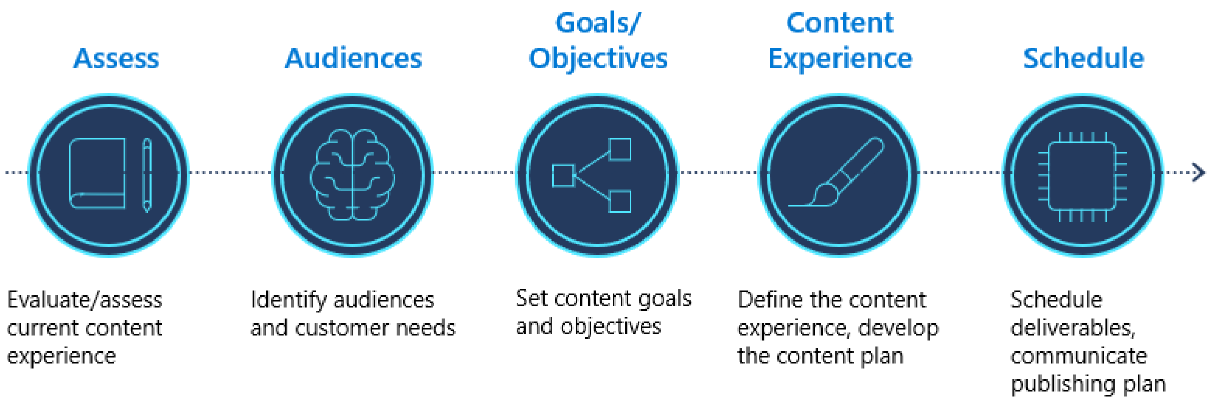
Technical Content Planning Framework

7/9/2021 • 9 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 Technical Content Planning Framework consists of a set of worksheets, guidance, and examples. These worksheets are designed to guide you in the development of a comprehensive content plan that assesses existing content, considers audience needs, defines content goals and objectives, delves into the overall content experience, and, lastly, drafts a publishing schedule.



Each worksheet builds upon previous worksheets. Apply the framework to a content set, subset, or a single article. Apply it to existing content or use it to plan new content. Use the entire framework, or select parts of the framework. And, customize the worksheets based on your specific technologies and content areas.

Benefits of using the framework

The benefits of applying this framework when planning an article or new content set include:

- Think more holistically about the consumers of your content and their needs
- Gain a greater appreciation of customer needs and identify new needs
- Shorten the time it takes to write comprehensive content
- Minimize content redundancy
- Create the right content the first time
- Focus on highest priority content

Additionally, benefits of applying this framework to refresh existing content include:

- Approach content with fresh eyes
- Increase content performance and customer satisfaction
- More easily identify missing content elements

Content planning

When developing content plans, writers must consider many factors, many listed below. The craft of technical writing occurs by addressing these various factors in a skillful and timely manner.

Information needs and stakeholder goals

- audience needs
- audience skills
- audience trust
- accuracy
- completeness
- content objectives
- diverse audiences and stakeholders
- learning styles
- organization goals
- product goals

Content experience and content journey

- address customer verbatims
- build on existing content
- content elements
- content experience
- content journey
- cross-product integration
- cross-service integration
- identify content gaps
- scenario-based content

Metrics and KPIs

- article freshness
- content health
- metadata
- page views
- bounce rate
- dwell time

Standards and best practices

- good introductions that clarify audience and benefits
- minimal content footprint
- modern voice
- SEO
- single article or a suite of articles?

Communicating plans and tracking dependencies

- communicating plans across teams
- promoting published content
- scheduling deliverables
- tracking dependencies
- working with subject matter experts (SMEs)

General guidance

General guidance is to capture in a word document the following elements.

1. If you have existing content, Perform a content assessment.
2. Identify audience needs.
3. Define content goals and objectives.
4. Develop the content experience and content plan.
5. Schedule publishing plan based on the content plan.

The TCPF Outline provides a template as a starting point. And, the Content Plan for Technical Content Planning Framework provides an end-to-end example.

Develop the most comprehensive plan possible. Use priorities to determine what is most important and inform the sequence of publishing articles. You may only develop half the content in a first phase and return subsequently to work on lower priorities.

> **TIP**
>
> When working through the worksheets, you may go back and forth from one step to another. Feel free to adapt and customize the worksheets and guidance to support your specific content areas.

Perform a content assessment

You can assess the content of a single article, set of articles, or entire content set. Having a strategy for assessing content can save time and support a focus.

Periodic assessment of existing content supports several goals:

- Improves content health
- Ensures content accuracy and completeness
- Meets current goals and objectives
- Refreshes the content experience within the existing larger content set
- Consolidates content to better support select audiences

Worksheet and guidance

- Content assessment
- Content assessment, examples

Identify audience needs

Understanding the various audiences and their information requirements is critical to developing new content and improving existing content. Recommended actions include performing one or all of the following tasks:

- Identify primary and secondary audiences and their primary and secondary information needs
- Identify each audience segment that will read the content.
- Review the information needs, both conceptual and instructional, that each audience segment needs.
- Assess information needs against audience current skills.
- Review with customers, PMs, and additional audiences to discover their primary and secondary needs. -
- State audience needs in customer language. Identify key words used by customers.
- Perform a review of competitor content to assess their approaches and what can be learned. Augment audience and business needs with key learnings.
- Specify calls-to-action (CTA) relevant to audience needs.

Worksheet and guidance

- Audience needs analysis
- Audience analysis, examples

Define content goals and objectives

Building upon the audience information analysis, you define a set of content goals and objectives. You align the objectives that meet audience and customer needs with business and organization goals.

In the process, you consider the following factors:

- Prioritize goals and objectives so that the most relevant ones are met first
- Content coverage, what types of content will meet content objectives
- Content rationale, how specific objectives meet information needs
- Data analytics, product analytics, and customer verbatims to support coverage and prioritization
- How new content fits within the existing content set(s).

Worksheet and guidance

- Content goals and objectives
- Content goals and objectives, examples

Develop the content experience and content plan

The next phase of the framework delves into the specific details that define the overall content experience. Each article is comprised on several content elements, such as concepts, tasks, images, tables, tips, notes, and more. By mapping the information needs to their content elements, you begin to identify the components that will comprise the content plan.

You can apply this method to existing content or developing new content. By breaking down the content needs into the component parts, you can then build up one or more articles that contain the right information targeted to the right audiences.

The steps involved in this phase include:

- Determine the granular content experiences that help meet the content objectives.
- Identify select content elements—conceptual images, tables, videos, gifs, etc.—for supporting audience understanding.:
- Group content experiences into a set of articles that support specific audiences and scenarios.
- Identify section heads to support consumability and findability.
- Prioritize articles and develop structured content plan and TOC.
- Consider the various ways users will find the content and ensure it is easy to find through search and relevant links.
- Identify appropriate ways to promote new or enhanced content to ensure it's discoverability.
- Ensure next steps and related content is surfaced in a way that is easily consumed and helpful to the reader.
- Consider if there are ways to personalize the content and work to incorporate that level.

Worksheet, guidance, and examples

- Content experience
- Content plan
- TCPF-ContentPlan-Security-Refresh
- Content Experience and Plan-Query Refresh

Schedule publishing plan

The last step in the framework is to identify the set of deliverables to be published and their priority for delivery. In this step, we recommend using Azure Boards to define a set of user stories parented to a feature. You can then assign each story to a content contributor and assign a target date or sprint for delivery.

Lastly, either through a query or Delivery Plan, you can communicate your plan with key stakeholders and track dependencies.

During this step, you want to exercise the following:

- Estimate the time and resources required to publish each deliverable.
- Set priorities for each deliverable.
- Publish content incrementally as needed to meet target deadlines.
- Communicate the plan with key stakeholders and get feedback.
- Ensure that the expected timeline is well communicated and understood.
- Effectively execute on the plan and publish in a timely manner.
- Effectively execute on a content promotion plan through social media or other means.

Worksheet, guidance, and examples

- Schedule deliverables

How long does it take to use the framework?

Estimated times vary depending on several conditions:

- Whether you are applying the strategy to a single article, a few articles or an entire content set.
- Whether you are starting from scratch on new content or refreshing existing content.
- How familiar you are with the product or technology you're documenting.
- How responsive your product team is.

The thing to note, is that the more comprehensive and detailed your content plan is, the easier it is to write the final content. You may also be better positioned to farm out sections of content to other contributors.

You may want to use only one or two of the worksheets.

Estimated time

Task and worksheet

1 article

3 to 5 articles

Doc set

1. Perform an Audience needs analysis

1 to 2 hours

1 to 2 hours

3 to 4 hours

2. Define Content goals and objectives

1 to 2 hours

3 hours

4 to 5 hours

3. Deconstruct the Content experience and reconstruct the Content plan

1 to 2 hours

2 to 3 hours

4 to 5 hours

4. Define the Schedule deliverables

10 minutes

30 minutes

1 hour

Framework definitions

Term

Definition

Audience analysis

An assessment of the various consumers of an article or content set that identifies the information needs, both conceptual and instructional, that each audience segment needs.

Audience segments

Audience segmentation is a process of dividing people into homogeneous subgroups based upon defined criterion such as product usage, demographics, psychographics, communication behaviors, and media use.

Content Coverage

From the Content Coverage Initiative - Overview.docx: Content Coverage ensures we're delivering the content our customers need in the format they want for their learning moment. By taking a holistic view of our content across Product Docs, Solutions, and Learn, we will identify the gaps in our overall content portfolio that need to be filled by content developers, part-time contributors, or external contributors.

Content assessment

The steps a writer or team take to review and determine how best to improve or revise an article, a group of articles, or an entire content set to better meet customer needs and business goals.

Content element

Any piece of content that appears within an article. Such elements include concepts, tasks, procedures as well as images, tables, FAQs, tips, and more. Taken together, they define the content experience for a specific article.

Content experience

The sum total of actions and information a customer encounters when they search for information about a product, feature, or how to accomplish a task. From this perspective, it touches on the search capabilities, content structure, findability, specific content elements, and any personalization or integrated help provided.

> *A content experience is the environment in which your content lives, how it's structured, and how it compels your prospects and customers to engage with your company. - Uberflip*

Content health

From the Content Health initiative: Reflects the quality of the content using various worksheets and validation tests. Content Health includes issues that negatively impact the customer experience such as writing quality, broken links, accessibility, terminology, and freshness/technical accuracy. Providing the right worksheets and tests throughout the authoring and publishing process becomes important as we increase the number of the part time and external contributors who may or may not be professional writers.

Content goal

Identifies specific outcomes to be achieved by the content that align with larger business and organization goals as well as meet customer needs for information.

Content journey

To be provided.

Content objective

A specific and potentially measurable result to achieve. A content objective indicates the actions to take to meet specific content goals.

Content plan

Defines the articles, high-level outlines, and publishing schedule for a content set. Also, it may include the research and strategy that informs the plan. A framework-generated content plan includes all the completed framework worksheets and notes captured while developing the plan. For an example content plan, see TCPF-ContentPlan word document.

Content strategy

Addresses the goals, objectives, and actions to take to assess and execute on the technical content needs to support customer, business, and organization goals.

Content assessments

7/9/2021 • 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.

Prior to writing new content or updating existing content, you want to assess the content you have. Content assessments can be made against a single article or an entire content set. In general, assessing the content set is recommended.

Application: A content assessment is a valid exercise when:

- Content hasn't been updated for a long period of time, 18 months or more
- Content has primarily been written by non-technical communicators
- Content has grown up around a bunch of features, but not thought through about audiences, content footprint, and clear calls to actions

Benefits:

- Identify easy updates you can make to improve the content experience
- A clear content strategy to refresh an article or content set
- Question how well the content meets current goals and objectives

General guidance

When assessing existing content, you want to consider the following factors:

- Content health metrics
- Audience needs
- Content goals and objectives
- Current organization and business goals
- Overall content experience
- Customer's content experience that may spread across our docs, technical communities, blogs, Stackoverflow, and more

Assess an individual article

General guidance for assessing an individual article:

1. Review Content Performance metrics and take action as outlined in How to troubleshoot lower-performing articles.
2. Review and address customer verbatims.
3. Review the article with fresh eyes.
    - What's changed since the article was first written?
    - Is consolidation in order.

- Are their background operations that need to be surfaced?
- Are examples clear and to the point?
- Have changes been made to the user interface or other functional operations?

4. Review the article with respect to the reader's experience:
   - Is the content easy to find?
   - Is it clear in it's intent, are there clear calls-to-actions (CTAs)?
   - Is it easy to get started for new comers?
   - Can a new user understand at a glance the supported features?
   - Is it clear who the audience is?
   - Is it easy to read? Engaging?
   - Are images, gifs, tables used appropriately, effectively?
   - Are there lots of other articles that address similar content? If so, how would a reader know which to read to figure out what applies to them?
   - Are choices clear and user guided to make the proper choice
5. Review the article with regard to these red flags or technology-specific areas:
   - Does the article build off other content that is foundational?
   - How hard are you making the reader work to gain the understanding they need?
   - Have you considered enterprise customers that work across products or services?
   - Are you telling the integration story that the product or enterprise supports?
6. Are you overloading the reader with too much information. Does the information need to be better segmented?

Assess a set of articles

Whereas an assessment can be made article-by-article, here we want to determine if another method might be better employed. The idea is to assess the content set from these points of view:

1. How well it addresses audience needs

2. How well it meets content goals and objectives

3. What can be said about the content experience?

   - Is the entire content set clear
   - Can customers find the content they need easily
   - Are choices clearly outlined and guidance provided
   - Are links between the content properly surfaced
4. When assessing a content set, consider:

   - Are there opportunities to consolidate content bases on similar tasks, objectives, scenarios

Assess a content set

Assessing a content set requires a journey across different content types, and requires a more subjective approach based on cross-group collaboration and experience.

To assess a content set, the recommended guidance is to start from scratch and apply the Technical Content Planning Framework to the set of articles. The reasons are:

- Address existing audiences and their needs.
- Review content goals and objectives based on current customer needs and business goals.
- Perform a Content Experience breakdown to help reveal missing content elements.

- Your better position to assess the existing content against an indepth understanding of what current content goals and objectives .

Next steps

Audience needs analysis

Examples

- Content assessment, examples

Audience needs analysis

7/9/2021 • 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.

Use the **Audience Needs Analysis** worksheet and guidance to identify each audience segment that will read your content. Identify existing information requirements as well as new requirements, both conceptual and instructional, that each audience needs. Assess information needs against audience current skills. Identify calls to action. And lastly, assign priorities to the audiences and their needs.

Application:

- Can apply to a content set, subset, or single article.
- Identify the customer audience(s) that the content will address, their primary and secondary information needs based on concepts and scenarios-job-tasks they need to become proficient in
- This is a deconstruction technique that will help inform the Content goals and objectives.

Benefits:

- Think more holistically about all the audiences that will consume the content and how the content will support them
- Think beyond the obvious information requirements and consider secondary or less obvious needs
- Identify the underlying goals of consumers of content and how to express those in a set of calls to action

Pre-work

Review the following articles:

- Writing content for C+AI
- The planning phase for contributing to customer facing content

Worksheet

Use the **Audience Needs Analysis** worksheet to identify the various readers or consumers of the content and the information they need to be successful. Readers access product documentation to perform a task, to get something done, or to understand why the product isn't working as they expect. You want to capture the most important information needs and identify any secondary needs. Also, noting the prerequisite knowledge or skills a reader has can indicate content elements to include.

The main columns of this worksheet are indicated in the following table: Audience & Priority, Primary information needs, Secondary information needs, and Knowledge prerequisites.

TIP

For a Word version of the worksheet, open the TCPF Outline.

Audience Needs Analysis worksheet

Audience	**Primary needs**	**Secondary needs**	**Knowledge Prerequisites**	**Skills to develop**	**Notes**

Fill out the worksheet

NOTE

Not all cells need to be filled in.

1. **Audience & Priority**: Audiences are based on the product area or technology area.
   - Identify key audiences such as Project Managers, Developers, Stakeholders, Administrators, etc.
   - Talk to the key stakeholders for your content—program managers, subject matter experts, marketing leads—to help with this list.
   - Consider internal audiences such as Sales, Support, Developer Advocates and how they might use the documentation.
   - See also the Audience segmentation list for ideas.
   - Assign a priority—1, 2, or 3—based on how large the audience is and how importance it is that they succeed in consuming the content.
2. **Primary needs**:
   - For each Audience, list one or more information needs, tasks, or goals that the audience needs to be successful.
   - Review Customer verbatims for primary and secondary needs
   - Assign a priority—1, 2, or 3—to the various information needs.
3. **Secondary needs**:
   - Consider edge cases, cross-service and cross platform content needs or goals.
   - Review stack overflow and Developer Community for primary and secondary needs
   - Consider needs to understand what isn't supported as well as what is - and present in a simple, direct way
   - Assign a priority—1, 2, or 3—to the various information needs.
4. **Knowledge Prerequisites**:
   - Indicate any knowledge or skills audience must meet in order to be successful.
5. (Optional) **Skills, Knowledge to develop**:
   - List knowledge or skills to develop as a result of consuming the content.
6. (Optional) **Notes**:
   - Any additional thoughts or questions about the audience and it's needs.

Calls to action

A second step in the audience needs analysis is to identify specific calls to action (CTAs).

> A call to action is an invitation for a user to take some desired action. You often see call to action examples in persuasive writing. Once a brand has made its case in a blog post or video, for instance, they'll often include a call to action at the end.

Most technical content is organized around a task or scenario that the reader wants to accomplish. As such, most CTAs represent a task or scenario. The best call to action phrases are brief and use strong verbs. Prior to performing a task or scenario, readers may need to have greater understanding by reviewing conceptual content. The CTA for conceptual content generally is around understanding.

Identifying a clear set of CTAs can help organize content and ensure you meet the main tasks and scenarios that meet your audience needs. Here is an example of CTAs for semantic search grouped by audience.

Stakeholders

- Find work items **Contributors**
- Search projects
- Search boards
- Search repos **Developers**
- Search artifacts
- Search code
- Search test plans
- Search repos

Administrators

- Search users, groups
- Search settings

Customer Support

- Understand how semantic search works
- Help customers troubleshoot search

Audience segmentation

While some products may have one or two main audiences, others—such as Azure DevOps—address several customers with varying responsibilities and knowledge. Identifying the list of audiences is the first step to ensure the content is useful to all audiences.

Also, such a list may be of use to the Content Coverage initiative where we may want to tag content based on the primary audience segments served.

> **NOTE**
> **To be completed**: This list should be reviewed by all content leads across C&L and augmented to capture their specific audiences.

Audience segment

	Knowledge level	Key responsibilities & core skills
Developers	Beginners to Expert	Primary consumers of products and services that support developing software.
DevOps engineers	Beginners to Expert	Primary consumers of products and services that support building and deploying software products and services.
Product Managers	Beginners to Expert	Responsible for interfacing with their customers and defining product features, prioritizing work, and interacting with product teams to ensure the most important work is being worked on first.
Stakeholders (Azure DevOps)	Nonspecialists	Limited access to Azure DevOps, primarily to support project planning.
Administrators (Cloud, Azure DevOps)	Technician	Responsible for managing organization settings, policies, and security.
Administrators (on-premises, Azure DevOps)	Technician	Responsible for managing on-premises servers, install, upgrade, settings, policies, and security.
Data Scientists	Expert	Responsible for generating reports and mining data to gain insights to impact business decisions.
Decision Makers	Executives	Responsible for evaluating major product or service purchases and making purchasing decisions.
Support Engineers WW Communities ALM + DevOps	Technician	Responsible for supporting product or service customers to use the product to meet their business goals.

Next steps

Content goals and objectives

Examples

- Audience analysis, examples

Content goals and objectives

7/9/2021 • 5 minutes to read

Use the **Content goals and objectives** worksheet and guidance to define the high-level goals for the content to achieve that align with audience needs as well as business and organization goals. Determine the specific actions you'll take to meet each goal. Identify specific and potentially measurable results to achieve.

Application

- Can apply to a content set, subset, or single article
- Builds from the audience needs analysis and begins to shape the required main sections of the content

Benefits:

- Ensures you scope the content project appropriately
- Identify and work the highest priority goals and objectives first
- Meet both audience needs as well as business and organization needs

Goal versus objective

The following image illustrates the main differences between a goal and an objective.

	GOAL	OBJECTIVE
DEFINITION	The direction and overall destination of your company that helps you realize your vision	The exact actions and steps your company must take to reach its goals
SPECIFICITY	General intention or direction	Specific, precise
PLAN	Broad in scope	Narrow in scope
SIZE	Large in size, the whole	Small chunks, part of the whole
EXAMPLE	Increase international customer base	Acquire 28% more customers from the UK, Australia, and Germany in next 3 months
ACTIONS	A general outcome	Specific actions and measurable steps
MEASUREMENT	Difficult; goals are usually intangible and may not be strictly measurable	Easy; it must be measurable and tangible
TIMEFRAME	Long-term	Medium- to short-term

Pre-work

Review the following articles:

- Audience needs analysis
- The planning phase for contributing to customer facing content

Worksheet

The **Content goals and objectives worksheet** defines why the content should be written and the specific actions to write the content.

> **TIP**
>
> For a Word version of the worksheet, open the TCPF Outline.

Content goals and objectives worksheet

Goals	Objectives	Notes, deliverables

> **NOTE**
>
> Not all cells need to be filled in.

Fill out the worksheet

When filling out the worksheet, consider how the content will meet C&L organization and product business goals.

1. **Goals**:
   - Working from the Audience needs analysis worksheet, identify a set of goals and objectives for the new or updated content. This establishes the scope of work to be done.
   - Identify the high-level goals to be met. Think about the existing content set if any, and specify goals that will address existing problems.
   - Identify goals that address specific audiences and that build on each other and audience needs.
   - Review the needs and how they relate to various Content coverage areas and to meet the following needs:
     - Get started content for beginners
     - Integration with other services
     - Meet larger goals other than addressing a single feature or worksheet
     - Review product team goals and identify content goals that support them.
     - Review C&L organization goals and identify content goals that support them.
   - Organize the goals with most important at the top of the worksheet, followed by next most important, and so on.
2. **Objectives**:
   - For each goal, identify specific objectives that will help meet that goal.
   - Consider the different Content Coverage and Rationale.

- Organize the objectives in order of importance to meet each goal.
- Assign a priority—1, 2, or 3—to each objective. Consider what is minimally viable content and assign it a 1. Assign a 2 for content that may increase customer understanding or satisfaction but isn't absolutely required.

3. **Notes, deliverables**:
   - Where it makes sense, add information that meets the listed objectives.

Additional notes

- If you have existing content and metrics, identify specific goals or objectives you want to meet to address improving content metrics. See Documentation and reference report
- Consider the need for a **Get started** section to help new users get oriented quickly and easily.
- Consider how to minimize the content footprint, what are the core content groupings that address the goals and objectives
- Consider creating a matrix to compare tasks, scenarios, features to help group content into the smallest footprint
- Identify the key conceptual content to deliver and how that supports calls-to-actions (CTAs).
- For scenario-based content, look at organizing all information—concepts, tasks, reference, scenarios, troubleshooting—within the same group.

Content coverage areas

The following tables describe various coverage areas to consider when defining goals and objectives. Based on the audience needs, one or more of these areas should be included in the content objectives.

Most common coverage areas

Area	Description	Examples
Get started	Content that supports and orients new users to the basic functions and usage of a service or sub-service. Get started sections often involve several articles that are targeted to new users to quickly get them up and running.	Start using Azure Boards Sign up for Azure Boards
FAQs	Provides Q&A content to address frequently asked questions.	Azure Boards FAQs Semantic search FAQs
Job-task	Quickstart, how-to, tutorial content that addresses how to accomplish a specific task or set of tasks.	Filter backlogs, boards, queries, and plans
Overview	Content that introduces a content set or subset to guide a user to the content that's available. This content should work to use common or industry language that will be well understood by the readers.	

Understand what you get with Azure Boards
About managed queries

Scenario

Content that addresses an end-to-end scenario to accomplish a complex set of procedures.

Define iteration paths and configure team iterations
Manage requirements, Agile methods

Troubleshooting

Content that supports user in resolving issues or errors that can come up when using the product.

Troubleshoot access and permission issues
Fix display, reordering, and nesting issues

Additional coverage areas

Consider including one or more of the following coverage areas as time and priority permit.

Area	Description	Examples
Best practices	Provides guidance in using the product or feature to be most efficient.	Best practices for Agile project management Guidance to create high-performing work item queries
Cross-product	Addresses information that supports a scenario that entails two or more products.	Connect a project to a GitHub repository
Cross-service	Addresses information that supports a scenario that entails two or more services within a product.	End-to-end traceability
Index	Content that provides a quick reference index to help users find information of interest. Entries may use alternate labels to those used in the user interface.	Quick index to work item tasks and concepts Index of work item fields
Process map	A process map is a picture of steps, decisions, and other elements of either a work or mechanical operation.	See Process mapping overview

Content rationale

A content rationale indicates the main rationale for including the content or making it a priority. Where did the need arise from? How was this need verified? The following list is suggested from the Content Coverage Initiative - Overview.docx and subject to revision.

- **Business Priority**: Addresses a business priority, re how to sign up for a service or how to connect one service to another.
- **Content Developer Identify**: Content Developer identified need.
- **Content Model Compliance**: Content required to be in compliance.
- **Content Request - Customer**: A specific request came up via a Developer Community page.
- **Content Request - Internal**: The product team, customer support, Sales, or other internal Microsoft team identified a need for content.
- **Language Parity**: Addresses covering various developer languages such as Python, Ruby, .NET, and so on.
- **Product**: Basic content to address product usage.
- **Role**: Content needed to support a specific customer role.
- **Solution/Scenario**: Addresses a high priority solution or scenario.

Next steps

Content experience

Examples

- Content goals and objectives, examples

Content experience

7/9/2021 • 16 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.

Use the **Content Experience** worksheet and guidance to identify the various content elements to deliver that make up the customer's content experience. By breaking down the information requirements into distinct elements, you can then create a content plan that you build from the component parts. You craft each article to contain the right information targeted to the right audience(s).

This is essentially a deconstruct/reconstruct method that you can use to construct the **Content Plan**.

Application:

- Can be easily applied to a single article, small group of articles, or an entire content set
- The main strategy is to review the existing content and identify how it contributes to the content experience and identify additional content needed to supplement of add to the experience.
- Identify content that may be out dated, not relevant, or should be moved elsewhere.

Benefits:

- Think through the various content elements that help tell the story you want to tell
- Helps identify missing elements
- Quickly organizes existing content into component parts

Pre-work

Review the following articles:

- The planning phase for contributing to customer facing content
- The design phase for contributing to customer-facing content
- TOC Structure
- Content goals and objectives
- Diagrams for technical communicators

Worksheet

The **Content experience worksheet** defines the individual content elements to include within the content plan. At this stage, your focus should be identifying the information elements that will meet the content objectives. Later, you'll organize the content into one or more articles.

TIP

For a Word version of the worksheet, open the TCPF Outline.

Content experience worksheet

Content elements	Content requirements	Candidate articles	Notes
Best practices			
Choice guidance			
Client support			
Command Line tool(s)			
Concepts			
Conceptual images			
Content roadmap			
Cross service features/Integrated features			
Examples			
Extensions			
FAQs			
Features at a glance Other elements at a glance			
Get started			
Keyboard shortcuts			
Navigational guidance			
Prerequisites			
Quick reference			
Reference content			
Related articles			
REST API			
Tasks, primary			
Tasks, secondary			
Tips			
Troubleshooting			
Video			

Content element	Information requirements (Priority)	Candidate articles
Best practices		
Choice guidance		
Client support		
Command Line worksheet(s)		

Concepts
Conceptual images
Content roadmap
Cross service features Integrated features
Key tasks/scenarios
Examples
Extensions
FAQs
Features at a glance
Get started
Navigational guidance

Prerequisites

Quick reference

Reference content

Related articles

REST API

Tasks, primary

Tasks, secondary

Tips

Troubleshooting

Fill out the worksheet

> **NOTE**
>
> Not all cells need to be filled in.

1. **Content elements**: Identify the various content elements that apply to your content set.
   - If you have existing content, conduct an audit of the various content elements that are already used in your article or content set.
   - Review the content for your product area and add content elements that are frequently used in your content set.
   - Review the Content elements, examples, and guidance for detail descriptions and examples.
2. **Content requirements**:
   - For each content element, determine the content requirements needed to meet the **Content objectives** identified in the Content goals and objectives worksheet.
   - List the content already met for each element.
   - Work to identify missing elements and list the content requirements associated with those elements.
   - Assign a priority—1, 2, or 3—to each element or set of elements. Consider what is minimally viable content and assign it a 1. Assign a 2 for content that may increase customer understanding or satisfaction but isn't absolutely required.
3. **Candidate articles**:
   - Start to specify the Article(s) in which the content requirements should appear and in parenthesis the targeted content type.
   - Group information into the smallest footprint as possible while considering the target audience(s), end-to-end scenario, and overall reading time; keep reading times to a reasonable size for your content set.
   - Minimize the amount of content sharing across articles.
4. **Notes**:
   - Capture any additional information in an optional **Notes** column.

Limit shared content across articles

You can share content within several articles using the Markdown INCLUDE syntax. In general, you want to limit the use of INCLUDEs according to the guidance provided in Include reusable content in articles.

Consider sharing blocks of text within several articles when:

- The INCLUDE content addresses concepts that are fundamental to conceptual overview content
- Not including the content would disrupt the flow of the content experience you're building

Not recommended

- Don't include the same Q&As in multiple articles, instead provide a single, targeted FAQ.yml file that you point readers to

Documenting tasks supported by multiple methods

Sometimes, there are multiple methods provided by a service to perform the same task. Each method may support the same options, or support slightly different options. Based on time and resources, you may only document a single method, but should at a minimum note that other methods are supported.

When different options are supported with different methods, we recommend that these options are fully documented. In particular, document how to perform tasks that are supported through the user interface and through a command line interface. You can do this using Tabbed Conceptual Markdown.

Here are a few examples that illustrate documentation of multiple methods:

- Add users or groups to a team or project
- View work items
- Run a query

Content elements quick reference

Here's a quick reference of the various content elements that go into creating the content experience. Your content areas may contain other elements.

Feel free to expand and adapt this list to meet your content needs.

Topic types

Content elements

Images and more

- Overview
- Quickstart
- Tutorial
- Sequence of tutorials
- How-to
- Conceptual
- Samples/Examples
- Troubleshooting
- FAQs
- Reference
- Index
- TOC

- Best practices
- Choice guidance
- Client support
- Command Line tools
- Concepts
- Conceptual images
- Content experience
- Content roadmap
- Cross service features
- Diagrams, Diagram as code
- Examples/samples
- Extensions
- FAQs
- Features at a glance

- Get started
- Integrated features

- Key tasks/scenarios
- Keyboard shortcuts
- Navigational guidance
- Other elements at-a-glance
- Prerequisites
- Process overview
- Quick reference
- Reference content
- Related articles
- Rest API
- Secondary tasks
- Tips
- Troubelshooting

- Conceptual images
- Diagrams
- Gifs
- Metadata
- Process maps
- Screenshots
- Tables
- TOC
- SEO keywords
- Videos

Content elements, examples, and guidance

A content element is any identifiable piece of content that appears within an article or a TOC. Such elements include concepts, tasks, procedures as well as images, tables, FAQs, tips, and more. TOC elements include Get started sections, Troubleshooting, Examples, and more. Taken together, they define the content experience for a specific article or set of articles.

The following list of content elements was developed based on Azure DevOps content. For your product, review the various content experiences you support and add them to the matrix. Remove those elements that don't apply.

Then, think beyond that, consider what your competitors might document. Add content elements that support your product or service and meet customer needs.

Element

Description

Guidance and examples

Best practices

Provides guidance for using features or functions to accomplish specific tasks.

Provide guidance that will support optimal usage or when users need to navigate several worksheets to accomplish one or more goals. Examples:

- Best practices for creating high performing queries
- Refine your backlog
- Best practices for Agile project management

Choice guidance

Defines options and guidance for choosing the best option based on customer needs.

Include when a user must make a choice between one or more alternatives. Identify the pros and cons of each choice or benefits of one choice over another. Examples:

- Choose a process
- Choose the query type
- Choose the authentication method

Client support

List the clients that support the product services and features.

Begin with a list of supported clients. Later, indicate which clients will be documented within Quickstarts, Tutorials, How-to-Guides, or may not be documented due to lack of resources. Examples:

- Web portal
- az boards CLI
- Team Explorer
- Eclipse, Team Explorer Everywhere
- Visual Studio
- Visual Studio Community
- Microsoft Excel
- REST API

Command Line Interface

Example for exercising a CLI to include in a task-focused topic.

Include when available. Typically using Conceptual Tabs in task-focused content, or in a command-line worksheet specific article. Examples:

- az boards query

Concepts

List of main concepts users should become familiar with to be effective in navigating and using the product or service.

Consider placing all major concepts within one or more conceptual topic, however divide up as necessary if different audiences are being addressed. Examples:

- About permissions, access, & security groups
- About security, authentication, and authorization
- About work items

Conceptual images

Images that will support reader understanding of concepts being addressed.

Consider how an image can help tell the story. Consider mixing conceptual images with screenshots as needed.

- Work item types and hierarchy

Content roadmap

A graphic or table with links to most relevant content grouped and annotated, providing the reader with an at-a-glance view of content supported for the product/features.

Consider including when there is a sequence in which articles should be read to get up and running. Examples:

- About Boards and Kanban
- Define work items and create your backlog

Cross service features/Integrated features

List of features within the service that are related to the primary service being documented.

Add information as needed to an Overview or Conceptual topic. Examples:

- Cross-service and enhanced query operations

Examples

List of examples or sample code to include to support understanding and use of a feature or service.

Examples that illustrate how to do something are highly useful content elements. May provide more than one example per article. Examples:

- Query by title, ID, or description
- Deploy to a Linux Virtual Machine
- Bug trends sample report

Extensions

List of extensions that customers should be aware of as they support additional features.

Consider including a description and links to extensions in an overview or conceptual article. Examples:

- Azure Boards extensions developed by Microsoft
- WIQL Editor
- Azure DevOps Open in Excel

FAQs

List of FAQs that will support the reader in understanding what is and isn't possible, guide them to where they want to go. Determine if 1 or more FAQ articles are needed.

Author in YAML format. Best if provided in it's own article versus included within another article. See also FAQ best practices. Examples:

- Work across projects FAQs
- Azure Boards FAQs
- FAQs for acquiring Azure DevOps extensions

Features at a glance
Other elements at a glance

A matrix that lists supported features, providing additional notes and links to content.

One of our main tasks as content developers is to provide guidance in making choices. However, sometimes the decision making is spread across several articles. When several options are available - providing a matrix that outlines those choices is recommended.

These matrices can be structured to show a single version or provide checkmarks to show which features are supported for different versions. The latter may be warranted to help support users make the case to upgrade to a later version. Examples:

- Filter backlogs, boards, queries, and plans
- Query features
- Default permissions and access

Get started

A section designed to orient new users to understand key concepts and use the ey features. Typically contains one or more conceptual topics, quickstarts, or tutorials.

Particularly useful for a major service - Such as Azure Boards, Azure Repos - but also for other service areas that have a number of features and applicable to a number of audience types, such as Security and Queries. Review any Page View metrics you have and consider including the high-page view articles in this section. Examples:

- Security Get Started section
- Get started with Azure Boards
- Query Get Started section

Keyboard shortcuts

List of supported keyboard shortcuts that support a service or feature.

Provide to support efficient navigation and accessibility. For accessibility, we should also provide keyboard shortcuts within each article where they can be used. Examples:

- Keyboard shortcuts for Azure Boards
- Keboard shortcuts (Dynamics 365
- Keyboard shortcuts in Power Query

Navigational guidance

A section that provides guidance on use of the icons and features supported for a product page.

Specify when there are many UI elements on the page that support interaction. Examples:

- Web portal navigation
- Query directory, query folders, and breadcrumbs

Prerequisites

List of requirements to be met before a user can perform a task or scenario.

Include to reduce customer frustration who try to exercise a task for which they don't have permissions or rights to. Examples include:

- Permissions
- Access level
- Configured resources
- Installed extensions

Process map

A process map is a picture of steps, decisions, and other elements of either a work or mechanical operation. This article provides worksheets for defining and summarizing a process.

See Process mapping overview. Examples:

- Process map

Quick reference

A list of links to content grouped by concepts, tasks, scenarios, or examples.

For a large content set that may include several tasks or concepts that reside across several articles, it may be difficult for a reader to locate what they are interested in. Build out from the various subsections in each article. Consider alternate verb choices or keywords used. Examples:

- Security quick reference
- Work item quick reference
- Dashboards, charts, widgets quick reference

Reference

Details functional elements, such as commands, operations, and more.

When possible, organize the content so that it is easily scanned. See also:

- SDK & reference content guide
- Document your Azure client library
- Write an Azure client library Quickstart
- Write .NET docs
- Write CLI docs
- Write Powershell docs
- Tag code examples
  Examples:
- Query data types, operators, and macros
- Link type reference
- Azure DevOps CLI

Reference content to link to

List of reference content that should be included and how it will be grouped in each article.

Reference content that relates to the main article's content. Examples:

- Reference articles
- REST API content
- CLI content
- Cross-service content

Related articles

Strategy for linking to other articles.

Most articles benefit from having links to related information. Think about what set of links will support each article. Always consider how each article builds from existing content or other articles within the doc set or across doc sets. Examples:

- Quickstarts & Tutorials, include Next steps
- Conceptual topics, include links to other conceptual topics and key task-oriented articles

---

REST API

List of relevant REST API articles. Strategy for including links to REST API content. Identification of features supported by REST API that may not be accomplished by UI.

Specify when REST API support the product area. Consider how to link appropriately to these articles. Investigate if there is information that needs to be explained which isn't correctly covered in the REST API. Also, does the REST API support functions not available through the UI. Examples:

- Azure DevOps Services REST API Reference
- Queries
- Work item query language
- Work item query language (WIQL) syntax
- Fetch work items with queries programmatically

Tasks, primary

List of Key tasks or scenarios to be developed, often written as quickstarts or Tutorials.

As a first step, be as granular as possible in your list. When developing your content plan, you'll look at grouping tasks into one or more articles. See also, Documenting tasks supported by multiple methods. Examples:

- Open, run queries
- Manage queries
- Define queries
- Edit a query
- Save a query
- Rename a query
- Delete a query
- Favorite/unfavorite a query
- Triage a list of work items defined in a query
- Filter a query
- Email a query
- Share a query URL
- Query across projects

Tasks, secondary

List of secondary tasks or scenarios, often written as How-to guides.

In the content plan you'll group each of these secondary tasks into one or more articles. Examples:

- Export/import queries
- Define hyperlink query
- Set query-level permissions
- Define query folders
- Rename query folders
- Organize queries
- Add a query to a dashboard

- Add a query chart to a dashboard
- Share queries

Tips, Notes, Warnings

List the tips, notes, and warnings to include in the content.

Typically Tips should be included within a section of an article, but can also be added to their own article. Tips shouldn't be labeled as such if they are FAQs. Examples:

- Useful tips when working with a tree list

Troubleshooting

Structured set of tasks to help troubleshoot areas.

See When to create a Docs troubleshooting article. Examples:

- Troubleshoot access and permission issues
- Unable to connect

Video

Illustrates one or more features, benefits, scenarios, or how-to tasks.

Video types include conceptual, how-to, sample showcase, design, or promotional. For how to, see Create and Publish Video. Examples:

- How to use the Azure Cognitive Services .NET Speech SDK for recognition
- What is cloud computing?

Diagram elements

Including useful illustrations is always welcome in technical content. To learn more about each of these elements, see the Diagram cookbook.

Element	Description
Class diagram	Describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.
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.
Component diagram	Depicts how components are wired together to form larger components or software systems.
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).

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.

Entity relationship diagram

An entity relationship model, describes interrelated things of interest in a specific domain of knowledge.

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.

Package diagram

A package diagram in the Unified Modeling Language depicts the organization and dependencies between various elements (packages) that make up a model.

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.

Venn diagram

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.

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.

Activity diagram (flowchart)

Activity diagrams are graphical workflow representations of stepwise activities and actions with support for choice, iteration, and concurrency.

Business Modeling Process Notation diagram (flowchart)

Business Process Model and Notation (BPMN) is a standard way to represent business processes graphically.

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.

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.

Next steps

Content plan

Examples

- Content Experience and Plan-Query Refresh
- Azure DevOps Search refresh
- Azure DevOps Security refresh

Content plan

7/9/2021 • 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.

Use the guidance and worksheets provided in this article to build off the **Content Experience** worksheet to develop your content plan. The **Content Plan** worksheet identifies each article, grouped within a TOC (Table of Contents), and their basic outline. If you have an existing content set, you can start with that and then annotate as needed.

Application:

- Can be easily applied to a single article, small group of articles, or an entire doc set
- Working from the **Content Plan** worksheet, generate comprehensive outlines for each article to be refreshed or written

Benefits:

- Develop a comprehensive blueprint of the content to be written or updated
- Identify a prioritized set of deliverables

Pre-work

- Review The planning phase for contributing to customer facing content
- Review The design phase for contributing to customer-facing content
- Review TOC Structure
- Complete the Content Experience worksheet.

Worksheet

The **Content plan worksheet** defines the individual articles and high-level outline for each. Your focus is to organize the content elements into the set of articles based on audience needs and topic type.

TIP

For a Word version of the worksheet, open the TCPF Outline.

Content plan worksheet

Article	Content Type & Status (New/Update/No Change)	Audience	Coverage Area/Rationale	Priority	Draft Outline	Notes/Promotion Plan

Fill out the worksheet

1. **Article**:
   - Working from the **Content experience worksheet**, identify a set of candidate articles. Organize the articles into the order they should appear in the TOC.
   - When updating a current content set, start with the existing TOC and articles.
   - Identify TOC groupings and place articles under the appropriate groupings.
2. **Topic Type (Priority)**:
   - Specify the topic type based on those listed in Choose the correct topic type for your article.
   - In parenthesis, indicate the priority for developing the content (1, 2, or 3). 1 being highest and 3 being lowest. This will help support the execution plan.
3. **Draft Outline**:
   - Develop an initial Outline for the article, identifying the main H2 section heads.
   - Indicate any sharing of topics across the overall content set.
   - (Optional) Assign a priority—1, 2, or 3—to each H2, where 1 indicates must publish with first draft; 2 can be considered for publishing later.
4. **Objective(s) met**:
   - List the objectives met by each article.
   - At a minimum, indicate the audience and rationale for including the article.
5. (Optional) **Promotion Plan**:
   - If an article lends itself to being promoted, indicate the promotion plan. For example:
     - Social media tweets
     - Blogs
     - Email to DLs
     - Presentation, DevOps SHOTs
6. **Final review**:
   - When finished, review the worksheet against the Audience needs analysis, Content goals and objectives, and Content experience.
   - Has anything been dropped?
   - Has content been grouped to minimize the content footprint?
   - Are the priorities correct?

Promotion plans

Additional guidance on promotion plans to be provided in future versions of this or other article.

Next steps

Schedule deliverables

7/9/2021 • 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.

Once you've developed your Content Plan, you're ready to develop a publishing plan. The publishing plan should contain a list of all deliverables, the priority for their development, target dates for their publishing, and sufficient information to communicate to stakeholders.

Application:

- Can be applied to an individual article, small group of articles, or an entire content set.
- Define an Agile plan of work by identifying a prioritized set of articles to write or update
- Estimate times for executing each deliverable, assign work to a content contributor, and assign work to a sprint

Benefits:

- Plan that can be reviewed with stakeholders
- Plan that can be executed by one or more content contributors
- Ability to track dependencies
- Method for tracking updates and capture discussion across stakeholders

Define deliverables

Working from the Azure Boards user interface defined for your team, perform the following tasks. For example, Azure DevOps uses Azure DevOps Backlog.

1. Create a work item for each new article or article that needs significant updates. Optionally group the user stories under a Feature.
2. Estimate the number of days each work item will take and specify in Story Points field. Include time for review and incorporating feedback.
3. Assign a Priority from 0 to 4. Order the work items in the Priority order that makes sense to work on them. Identify what you can publish without publishing the entire set.
4. Assign each work item to a sprint.
5. Assign each work item to a content contributor, if other writers will work on the deliverables.
6. Identify sub-tasks if that is helpful and to distinguish work that others will do.
7. (Optional) Tag work items to easily query on them as a group of related work.

The following image illustrates the query of work items defined for a Query refresh project.

ID	Priority	Work Item...	Title
1810852	1	Feature	Azure Boards Queries - Content Strategy Refresh
1806943	1	User Story	Define a query (Was Create and save managed queries) | /azure/devops/bo...
1807774	2	User Story	About managed queries second updates [UPDATE]
1807775	1	User Story	Query FAQs [NEW]
1808424	1	User Story	About managed queries
1808426	3	User Story	Query support for cross-service, integration, and extensions [New}
1808427	1	User Story	View, open, run a query (Was View, run, or email a query)
1808428	1	User Story	Manage queries & query folders (Was Organize queries with folders)
1810864	1	User Story	Semantic search (Was Ad hoc search) [UPDATE]
1810865	1	User Story	Query quick reference [UPDATE]
1818846	2	User Story	Bulk import and update work items using CSV files

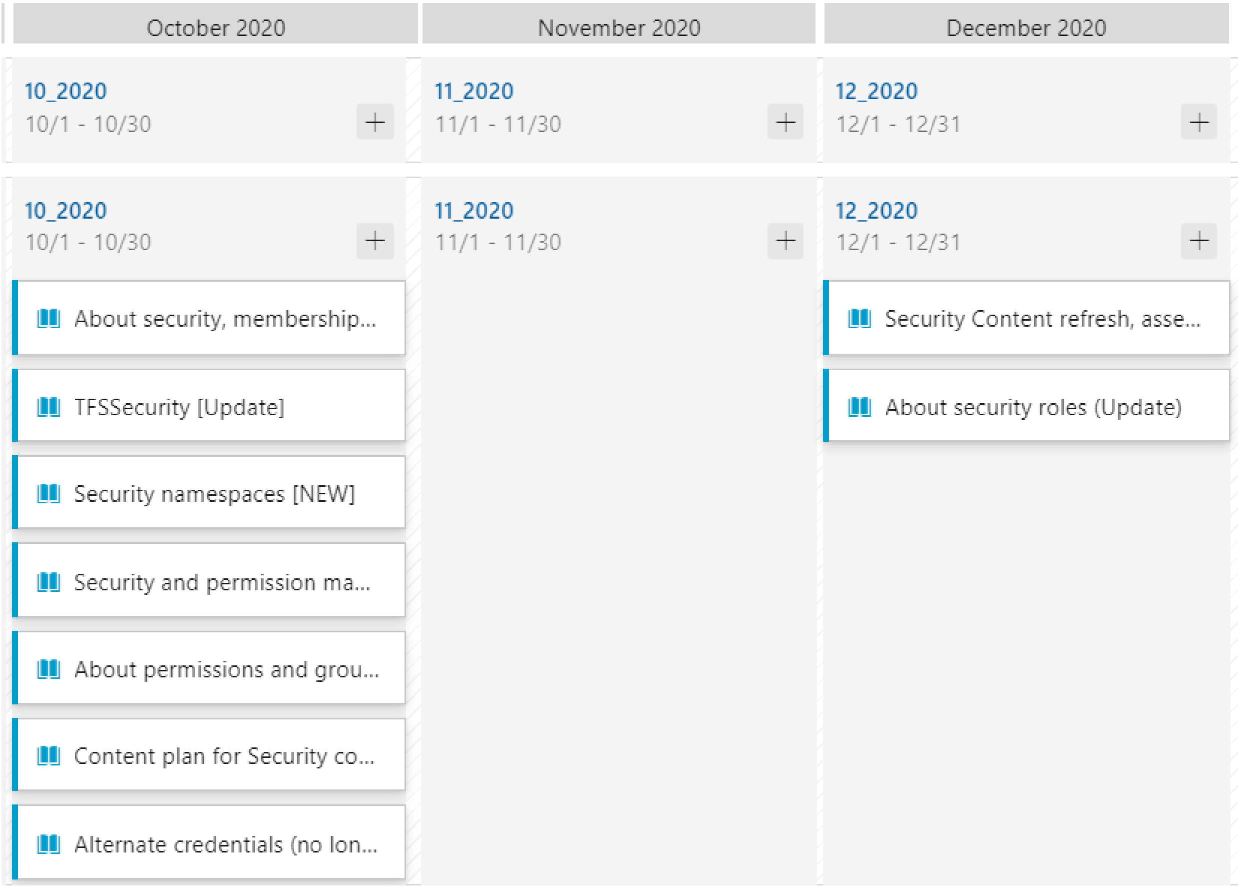
Prioritize work

Identifying priority is only important if the work will take more than one sprint to complete. Sometimes the priority will be clear. However, here is some guidance to help decide priority:

1. Identify the task-based scenario topics that are high priority to complete.
2. Identify the conceptual topics needed to support customer understanding.
3. Prioritize get started content.

Delivery plan

Delivery plans show a calendar view of deliverables which can support quarterly planning or review of publishing plans with stakeholders. The following image shows a screenshot of the delivery plan for the Security Refresh deliverables. To learn more about defining delivery plans, see Review team Delivery Plans.



Track dependencies

You can track dependencies of various work items by adding Predecessor and Successor links to work items. Using delivery plans, you can track dependencies and identify when issues arise. To learn more, see Track dependencies using Delivery Plans.

Content assessment, examples

7/9/2021 • 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.

Several examples are provided to illustrate ways to assess a single article and showcase some before and after rewrites. Examples address some of these red flags:

- Introductory paragraphs aren't engaging
- Introductory paragraphs don't build off foundational articles
- Story being told isn't engaging or well-targeted

As time and resources permit, additional examples will be provided.

Conceptual overview article introduction

The following example illustrates an introductory paragraph for the About pipeline resources article.

Before

After

A resource is anything used by a pipeline that lives outside the pipeline. Resources are defined at one place and can be consumed anywhere in your pipeline. Resources can be protected or open.

Azure Pipelines uses resources to support these tasks:

- Share resources, such as agent pools, to provide the infrastructure to run pipeline jobs
- Reuse across pipelines defined within a project
- Support automation of CI/CD
- Securely store information that needs to be reused by pipelines
- Connect to external and remote services to execute pipeline tasks

A table that includes a linked-list of resources, short descriptions, and other specifics would be more helpful.

Story being told
There are lots of resources used in pipelines.

Story being told
Resources support several pipeline tasks to share, reuse, support, and connect.

Assessment

- Doesn't engage me in why I should know about resources.
- I'm new and don't know what protected or open means.

- Doesn't build upon previous content that user may need to be familiar with
- Doesn't indicate that Resources can be shared within pipelines defined in the same project.
- Provides a linked-list of resources, but not a brief description. I have to click through several articles to get that information.
- Doesn't distinguish which resources are required and which are optional. Which are protected, which are opened.

How-to article introduction

The following example illustrates an introductory paragraph for the Define, capture, triage, and manage bugs article.

Before

After

How do you track and manage defects in your code? How do you make sure software problems and customer feedback get addressed in a timely manner to support high-quality software deployments? And, how do you do this while making good progress on new features?
At a minimum, you need a way to capture your software issues, prioritize them, assign them, and track progress. Moreover, you'll want to manage your bugs in ways that align with your Agile practices.
In a nutshell, you manage bugs through the following tasks:

- Capture information using the bug work item type
- Triage bugs by assigning a priority
- Update bug status throughout the bug lifecycle
- Monitor bug assignments and trends

Each team can choose how bugs show up on their backlogs and boards.

How do you track and manage defects in your code? How do you make sure software problems and customer feedback get addressed quickly to support high-quality software deployments? And, how do you do make good progress on new features while addressing your technical debt?

At a minimum, you need a way to capture your software issues, stack rank them, assign them to a team member, and track progress. You'll want to manage your code defects in ways that align with your Agile practices.

To support these scenarios, Azure Boards provides a Bug work item type. The Bug work item type shares all the standard features of other work item types with a few more. For an overview of standard features, see Track work with user stories, issues, bugs, features, and epics.

Extra features for managing bugs include the following:

- Option for each team to choose how they want to track bugs
- Test worksheets to capture bugs
- Built-in integration across Azure DevOps to track bugs linked to builds, releases, and tests

Story being told
Here are some things you might want to know about managing bugs.

Story being told

- How the Bug work item type provides additional features over those used by other work item types.
- Azure DevOps has a robust integration story when it comes to bug management.

Assessment

- Does address the key tasks of interest around bug management.
- Doesn't relate to how the Bug work item type DIFFERS from the other work item types. Doesn't BUILD on this particular information.
- Doesn't highlight the integration story that Azure DevOps supports with regard to bug management
- Use of the word "nutshell" isn't recommended for localization purposes

Related articles

- Content assessments

Audience needs analysis, examples

7/9/2021 • 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.

Examples are provided that illustrate use of the Audience needs analysis worksheet.

Azure DevOps Search content set refresh

Background notes: The original content was written by the product managers. They wrote content to support the search box feature based on searching for Code, Work Items, Wiki content, and Packages or Artifacts. 85% of the semantic search features are shared across all four areas.

Refresh focus: Consider all search needs for Azure DevOps and address both semantic search, how users use semantic search, and other search features supported by Azure DevOps.

For a review of the audience needs analysis, see Azure DevOps Search refresh

Azure DevOps Security content set refresh

Background notes: The original content was written over several years and updated as new features and functions were introduced. Security content addresses 18+ audiences, 30+ concepts, and 100+ tasks.

Refresh focus: Segment content to address specific audiences. Develop a get started section. Consider end user goals and how security content meets those goals.

For a review of the audience needs analysis, see Azure DevOps Security refresh

Calls to action: Security

- How do I (Individual Contributor) ...
  - View permissions, get elevated permissions
  - Understand permissions granted to me
  - Create a personal access token or other authentication credential
- How do I (Project Collection Administrator) manage users and user access
  - Manage users and user access
  - Manage users and user access with Azure Active Directory or AD
  - Change access levels (on premises Admin)
  - Add users to teams, team projects
- How do I (Project Administrator and Project Collection Administrator) restrict or grant access or permissions:
  - Azure Boards features
  - Repositories and branches
  - Pipelines and Pipeline resources [Security roles]
- How do I (Individual Contributor, Project Administrator)...

- Get my application authenticated
- Manage authentication policies
- How do I secure the following....
  - A project
  - An organization
  - Azure Boards, work tracking
  - Azure Repos, code, repositories, branches (Developers)
  - Azure Pipelines (Developers, Pipeline Admins)
  - Azure Test Plans
  - Azure Artifacts
  - An on-premises deployment

Content goals and objectives, examples

7/9/2021 • 4 minutes to read

Examples are provided that illustrate use of the Content goals and objectives worksheet.
Word versions of these examples are provided here:

- Azure DevOps Search refresh
- Azure DevOps Security refresh

Azure DevOps Search content set refresh

The worksheet for this content set refresh builds off the Audience needs analysis applied to Azure DevOps Security content set.

Goals

Ojectives

Notes, deliverables

1. Address the major calls to action, scope the conceptual and how to tasks towards those CTAs. Organize content around scenario-based concepts and tasks and examples. Present conceptual information targeted to appropriate audiences.

- Specify the key scenarios to address based on primary scenarios and CTAs
- Present conceptual information, tasks, reference, examples, and FAQs targeted to appropriate audiences and scenarios.
- Identify all Search conceptual information

Organize content around these major CTAs:

- How do I do a basic search, quick?
- How do I do a command-level search?
- How do I search code?
- How do I search work items?
- How do I search artifacts or packages?
- How do I search wiki?

2. Ensure that content is easily found. Make hi-pri topics more discoverable, improve SEO

- Provide a quick reference to Search tasks and concepts and examples, based on audience

- Search basics - conceptual = "Get started with Search"
- Links to all articles in table
- Delete Index page and redirect to Overview article
- Apply keywords in metadata

3. Identify information that is out of date and update it. Remove most references to TFS.

- Apply articles to all versions and apply monikers

- Combine overview with Navigation topic
- Remove most TFS refs

---

4. Review the content for how it fits in the overall Azure DevOps content set, Azure environment, and C&L resources.

- Identify how Search content shows up across Azure DevOps content · Identify articles to link to

- Add links to and from other articles

---

5. Decrease repetition/duplication and find and close any content gaps.

- Identify repetitive content
- Identify content gaps

- Combine "basic" and "advanced" searches into related single articles
- Combine index, overview, and navigation topic info
- Write new article about indexing

Azure DevOps Security content set refresh

The worksheet for this content set refresh builds off the Audience needs analysis applied to Azure DevOps Security content set refresh.

Goals

Ojectives

Notes, deliverables

A. Address the major calls to action, scope the conceptual and how to tasks towards those CTAs. Organize content around scenario-based concepts and tasks. Present conceptual information targeted to appropriate audiences.

1. Document the key scenarios to address based on primary scenarios and CTAs
2. Present conceptual information, tasks, reference, and troubleshooting content targeted to appropriate audiences and scenarios.
3. Regroup reference and troubleshooting tasks under the primary scenarios
4. Identify all Security conceptual information
5. Segment the conceptual information based on key audiences

Organize content around these major CTAs:

- Get started
- Add & manage user access
- Authentication and secure applications
- Secure projects
- Secure organizations
- Secure Azure Boards & Analytics
- Secure Azure Repos
- Secure Azure Pipelines
- Secure Azure Test Plans
- Secure on-premises deployments

- Secure networks and data

B. Provide an easily consumable get started section targeted to the primary audiences

1. Identify the key concepts and tasks for Stakeholders through Organization Owner to get started on security elements.
2. Outline a set of articles that meet get started needs.

Key concepts

- Permissions, security roles, access, authorization
- Default permissions and access (include roles)
  Tasks
- View permissions (overview)
- Troubleshoot permissions
- Ask to get elevated permissions **Articles: Get started **
- About permissions, access, and authorization
- Default permissions and access
- Trace permissions
- Increase your permission levels
- Security quick reference

C. Where multiple choices for doing something are present, ensure appropriate guidance for making the choice is provided.

1. Review conceptual information for guidance in making smart choices
2. Review FAQs and determine additional FAQs

Choices – guidance:

- Assign access level - ✔
- Grant or restrict access
- Choose authentication method (needs work)
- Assign security role

D. Ensure that content is easily found.

1. Provide a quick reference to Secure tasks and concepts based on audience For lookup guide, Change over HTML table format to MD syntax format.
2. Make sure that the Permission lookup guide is up to date.

For lookup guide, Change over HTML table format to MD syntax format.

E. Identify information that is out of date and update it. Remove most references to TFS.

1. Update Azure Artifacts security content, apply monikers
2. Run a search against the md files for instances of TFS, and Team Foundation Server and look at removing.

F. Review the content for how it fits in the overall Azure DevOps content set, Azure environment, and Microsoft docs.

1. Identify how security content shows up across Azure DevOps content
2. Identify Azure Environment security articles to link to

- Security content shows up in **Prerequistes** in most task-based articles. Identify topics missing Prerequisites and create user stories to add it.
- Some security content is shared across Security articles and service-specific articles. Review and make this as consistent and complete as possible. **Potential Learn links:**
- Explore DevOps Continuous Security and Continuous Operations
- Secure and isolate access to Azure resources by using network security groups and service endpoints
- Top 5 security items to consider before pushing to production
- Improve your cloud security posture with Azure Security Center
- Security documentation

Design your content

6/29/2021 • 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:

☑	TASKS
☐	*If a repo doesn't exist*: 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. For example, *virtual-machines* is the service slug for Azure Virtual Machines and is consistently used between the Docs site (https://docs.microsoft.com/azure/virtual-machines/) and the ACOM site (https://azure.microsoft.com/services/virtual-machines/).
	If a repo already exists: Work with the sponsor to coordinate your new content.
☐	Create a new release branch for your new content.
☐	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. For example, this folder matching the service slug is used for the Azure Virtual Machines articles: https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/virtual-machines.
☐	Create the new metadata value. Use the new ms.service value request form to add the new metadata to the list. Be sure to select **Yes** to include your new Docs metrics in the content engagement reports. The ms.service metadata property typically uses the same value as the service slug.
☐	Get the new .svg icon from the marketing representative for your release or product.
☐	Based on your content needs, list the articles that you need to create. Review the recommended article types for Docs.
☐	Based on your article list, define the TOC structure 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 TOC checklist spreadsheet to draft your TOC.
☐	Start thinking about landing pages and hub pages 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.
☐	Add your content set to the main breadcrumb file: \bread\toc.yml. Review the breadcrumb guidelines.

☑	TASKS
☐	If you have a product forum URL, add it to the docset's docfx.json file: MicrosoftDocs/docs-help-pr/docfx.json. Under `feedback_product_url`, add your folder path and feedback URL. (See other line items as examples.) Review the article How to enable the documentation feedback control.
☐	Submit a request 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

6/28/2021 • 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
☐	Create a release branch)	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 *release-* prefix followed by feature or content name.
☐	Write code	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.
☐	Write first draft of content	Configure your authoring environment, set your remote repositories, and create a local branch. Then, create your Markdown files for your content.
☐	Create rich media	Create images and media effectively and accessibly. Save your media in a `media` folder, and reference it with correct Markdown syntax.
☐	Address style errors	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

6/28/2021 • 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.

☑	TASKS
☐	When your content is ready, understand the pull-request (PR) submission recommendations and create a PR to receive feedback from SMEs.
☐	Check the build validation, and address any issues.
☐	Check the Acrolinx results, if available. See Acrolinx coverage.
☐	Invite reviewers to provide feedback on the new content.
☐	After you incorporate the reviews and feedback, sign off (**#sign-off**) 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.
☐	Schedule your release branch merge with the main branch.
☐	Request Go Live.
☐	Generate a forward link (FWLink) to your landing page so your product team can link to the content from their product's user interface.
☐	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 PLR sign-off for articles.
☐	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

6/28/2021 • 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
☐	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
☐	Review content performance.
☐	Address any GitHub issues.
☐	Retire old content.

Next steps

Contribute to Docs

Writing content for C+AI using the MVC content strategy

7/7/2021 • 4 minutes to read

The Minimum Viable Content (MVC) content strategy exists to support content quality and consistency on docs.microsoft.com. It starts with direct collaboration between engineering and content partners to define what customers want to do with the service/product and continues through five milestones.

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

TASK	DELIVERABLE	REVIEWERS
Milestone 1: Top tasks 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, Engineering partners
Milestone 2: Plan your content and your TOC Build your content plan. It may help to use the TOC checklist spreadsheet. Include the tasks and a customer intent statement for each article. 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. 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 definition for what type of content belongs in each section of the TOC is in the Guidance for TOC structure article.**	TOC checklist	Writing team M1 manager

TASK	DELIVERABLE	REVIEWERS
Milestone 3: Create your first articles 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 & Learning writing guidelines. Build your landing page 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. When CLI and PowerShell scripts can be written to automate common tasks, write them and store them in the repo.	Content review	Writing team, Engineering partners
Milestone 4: Create all articles for a release 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. Only write articles where the platform story is complete. Apply feedback from Milestone 3 review as appropriate to these articles. Update the TOC and landing page.	Release branch corresponds to approved content plan	Writing team, Engineering partners
Milestone 5: Publish 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

MVC content set requirements

TOPIC TYPE	WHAT IS IT	WHEN TO USE IT
Overview	**Required:** One article explaining the service from a technical point of view. **Optional:** A second article comparing features across related services/projects.	
Quickstarts	**Recommended:** Fundamental day-1 instructions for new customers to quickly try out a specific product/service. 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.	
Tutorials	**Required:** Scenario-based procedures for the top customer tasks identified in milestone one. The focus is on showing the customer how to do the task, not on helping them set up their own environment.	
Samples	**Recommended:** Example scripts that align to the most common tasks. The scripts should be short, simple, and quick to complete.	
Concepts	**Optional:** In-depth explanation of functionality related to a service(s) that are fundamental to understanding and use.	

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

Choose the correct topic type for your article

7/7/2021 • 2 minutes to read

Use 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	An article explaining the service from a technical point of view. And/or an article comparing features across related services/projects.	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	Fundamental day-1 instructions for new customers to quickly try out a specific product/service. 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.	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	Scenario-based procedures for top customer tasks. The focus is on showing the customer how to do the task, not on helping them set up their own environment.	To show the single best procedure for completing a customer task.
Samples	Example scripts that align to the most common tasks. The scripts should be short, simple, and quick to complete.	To provide the most common scriptable tasks.
Concepts	In-depth explanation of functionality related to a service(s) that are fundamental to understanding and use.	When you want to: - explain what something is and define its critical functions. - describe how something works or what happens when something is done. - provide guidelines for completing a job task in varied situations.
How-to guides	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.

TOPIC TYPE	WHAT IS IT	WHEN TO USE IT
Troubleshooting articles	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	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	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

7/7/2021 • 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.

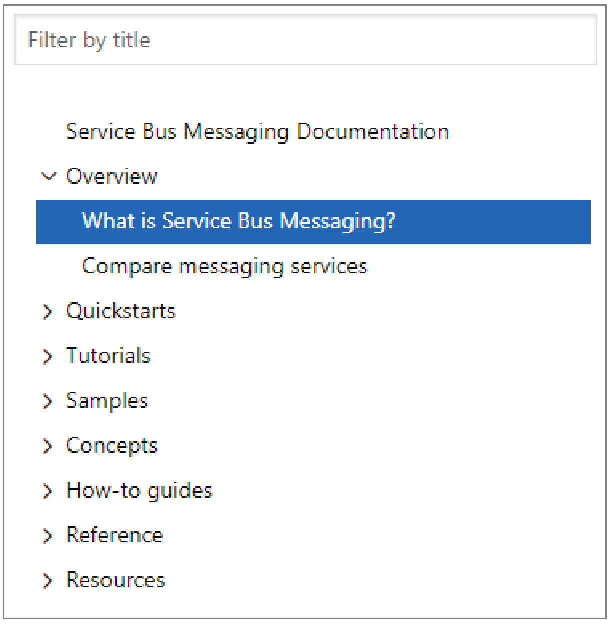
You may also create a second article that compares features across related services. For example, see Comparing hosting options.

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.



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

7/7/2021 • 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.

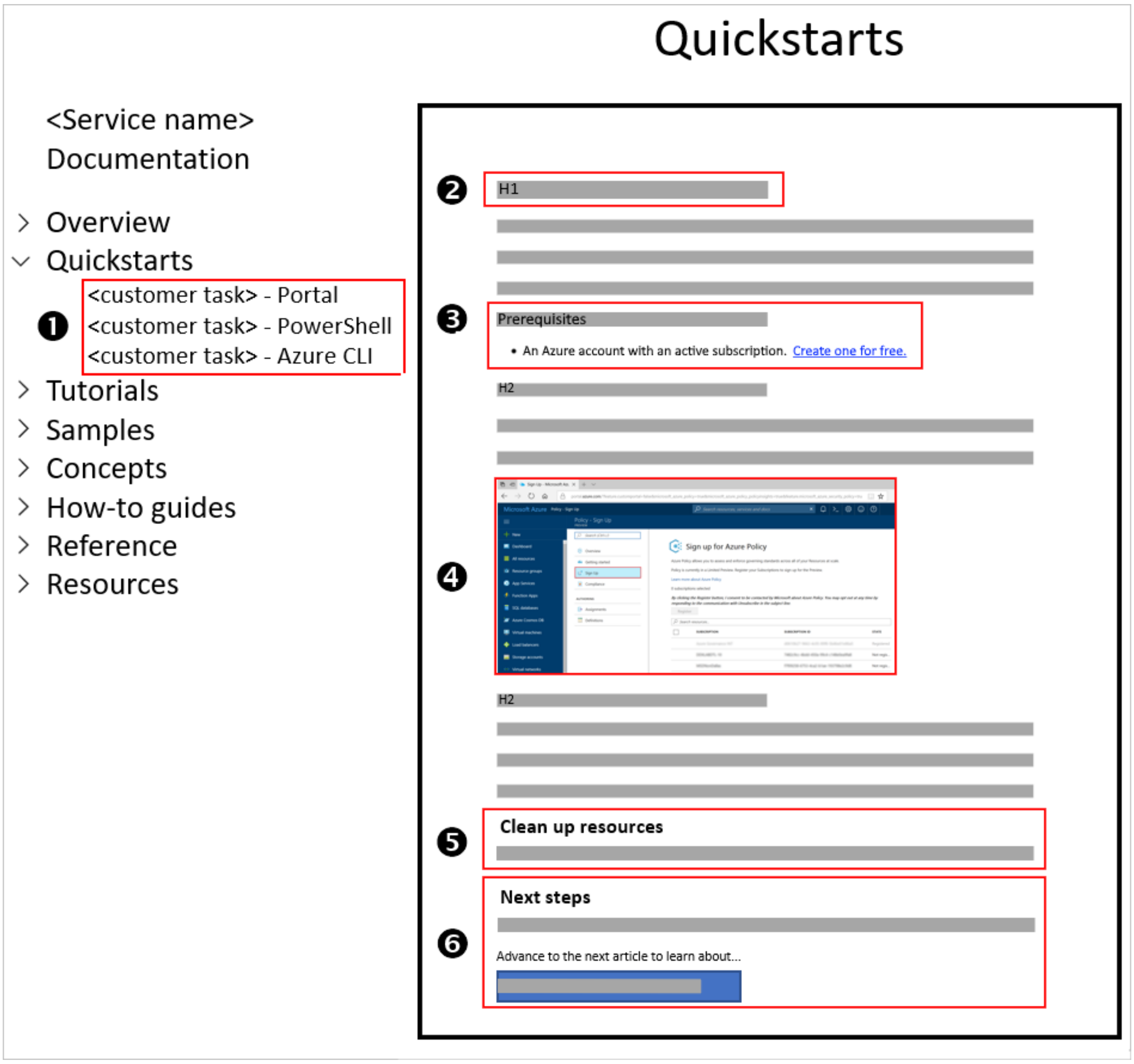
Use this template when writing a new quickstart article.

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.



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).
  ```

  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 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

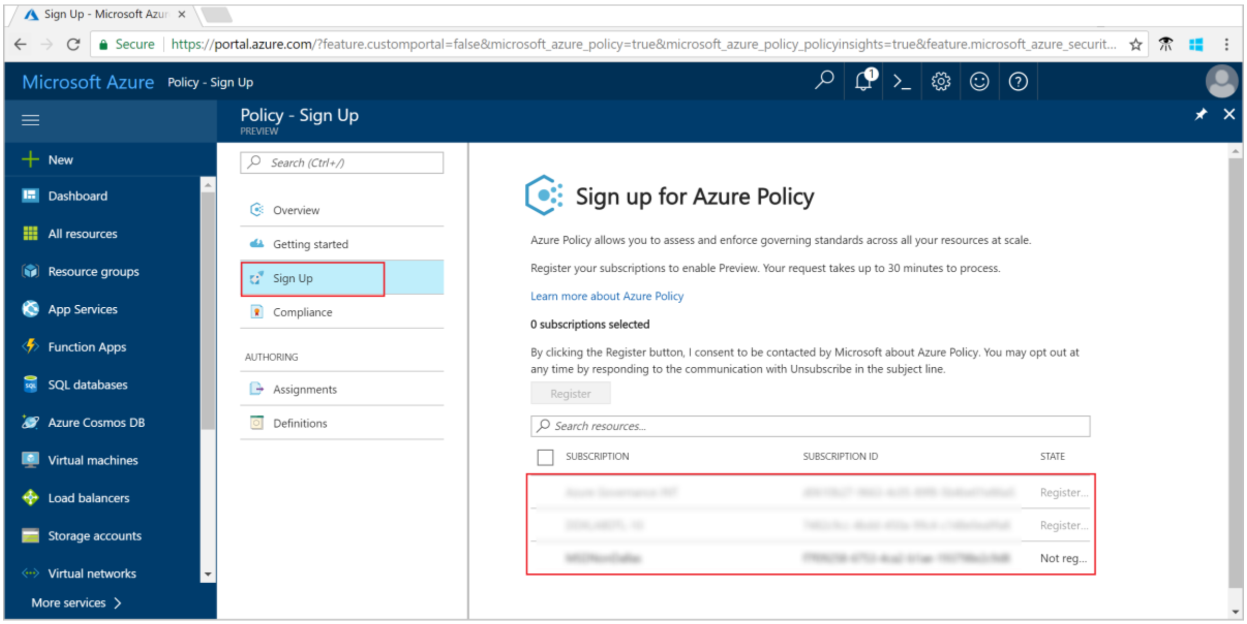
  ```
  - [Azure CLI 2.0.76+]( /cli/azure/install-azure-cli).
  ```

- 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:

```
> [!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

6/24/2021 • 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][readme-template] 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][sdk-guide-terminology] 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 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][readme-template] 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 >

contributor-guide-style-guide-hierarchy [style-guide-msft]: https://docs.microsoft.com/style-guide/capitalization [style-guide-cloud]: https://styleguides.azurewebsites.net/StyleGuide/Read?id=2696&topicid=25357 [sdk-guide-terminology]: https://azuresdkspecs.z5.web.core.windows.net/DesignGuidelines.html#general-terminology [readme-template]: https://github.com/Azure/azure-sdk/blob/master/docs/README-TEMPLATE.md

Write a quickstart about deploying Azure services by using ARM templates

6/24/2021 • 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.

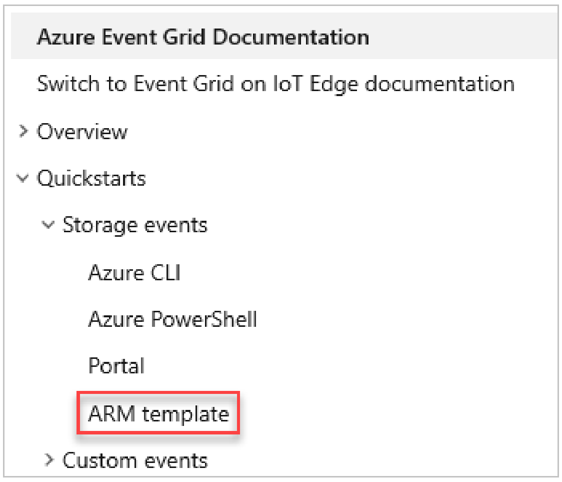
Write an ARM template quickstart article for your service

To begin writing your ARM template quickstart article use use this Markdown template.

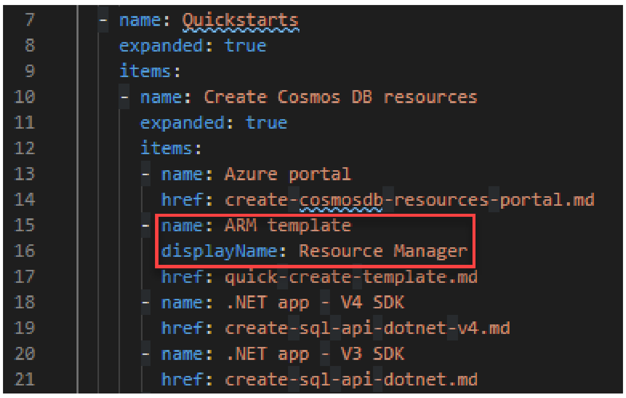
> **NOTE**
>
> Remove all the comments that are enclosed by `<!-- -->` from the template file before your quickstart article is pushed to GitHub and published.

TOC

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



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



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"
  $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 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

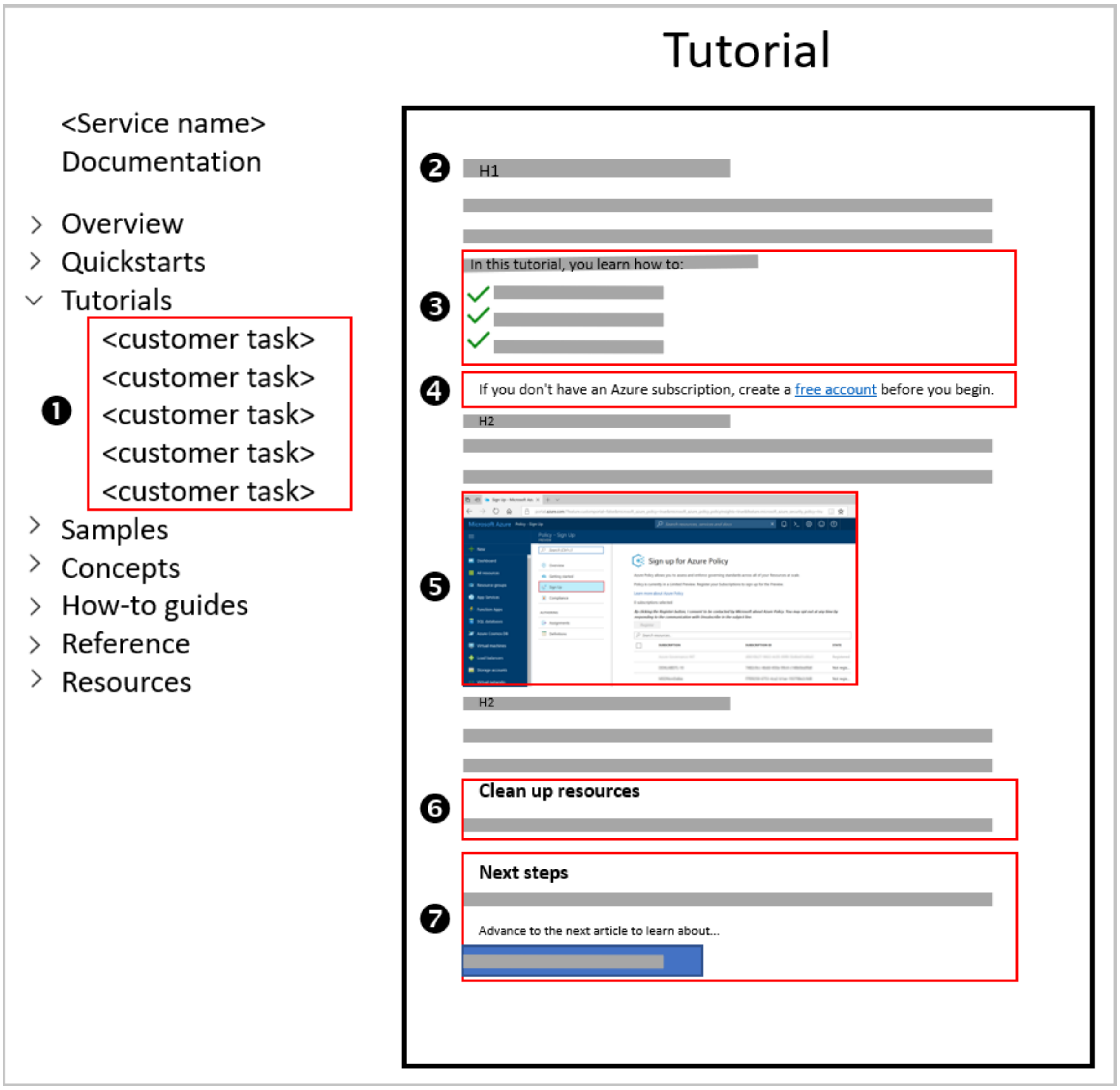
7/7/2021 • 8 minutes to read

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



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 *green* 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:

```
> [!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) 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

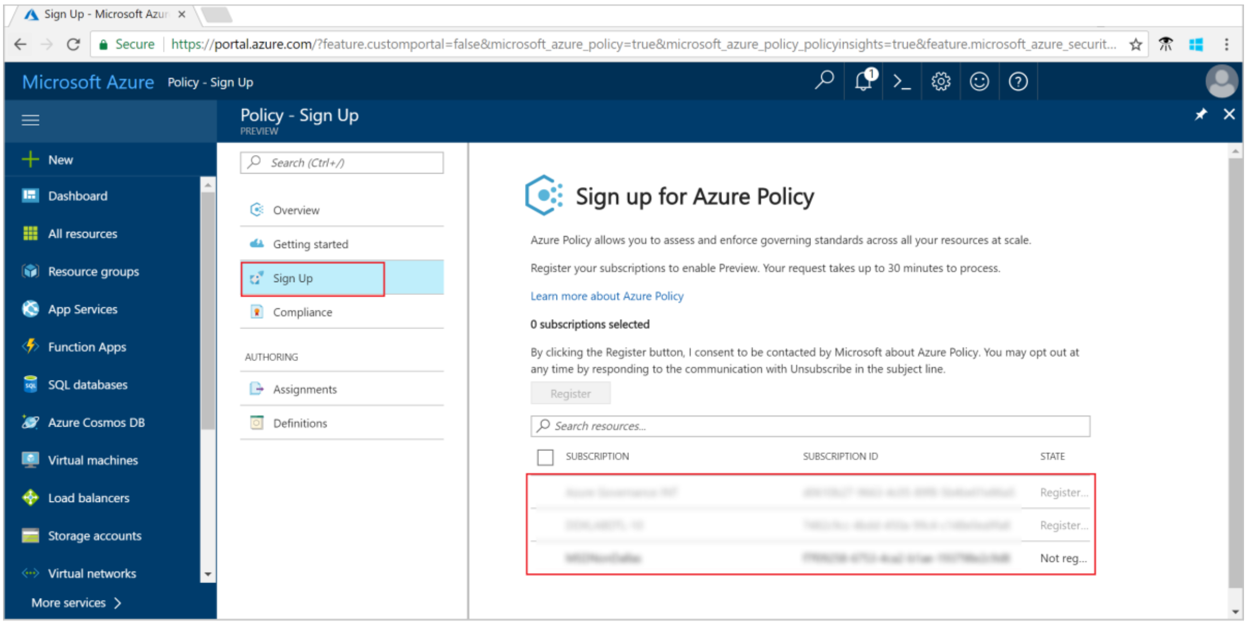
  This tutorial requires that you run the Azure CLI locally. You must have the Azure CLI version <version # (for example 2.0.4)> or later installed. Run `az --version` to find the version. If you need to install or upgrade the CLI, see Install Azure CLI 2.0.

- 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:

```
> [!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

6/28/2021 • 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.

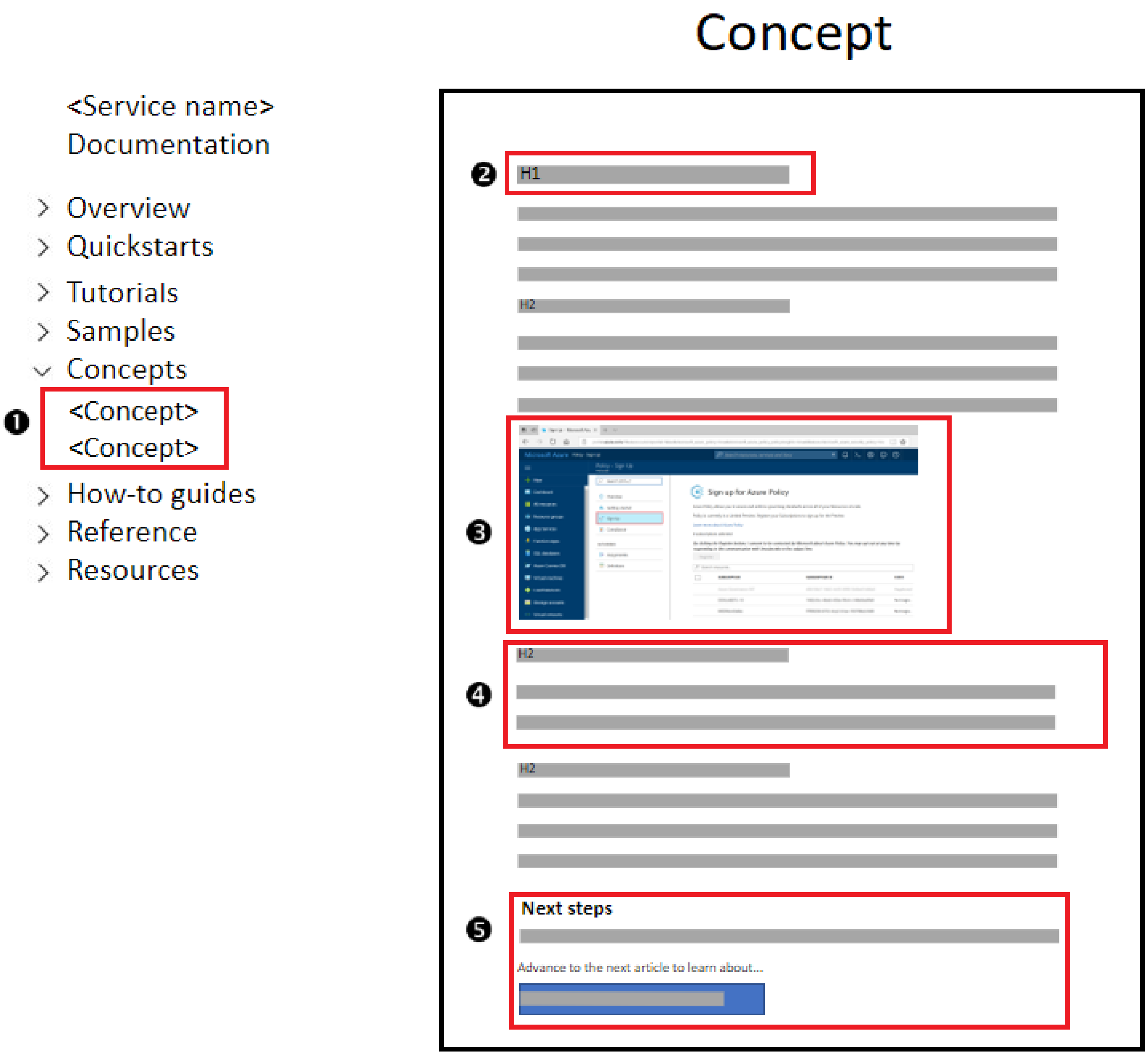
Why would I use a concept article?

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



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 *(OR)* 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.

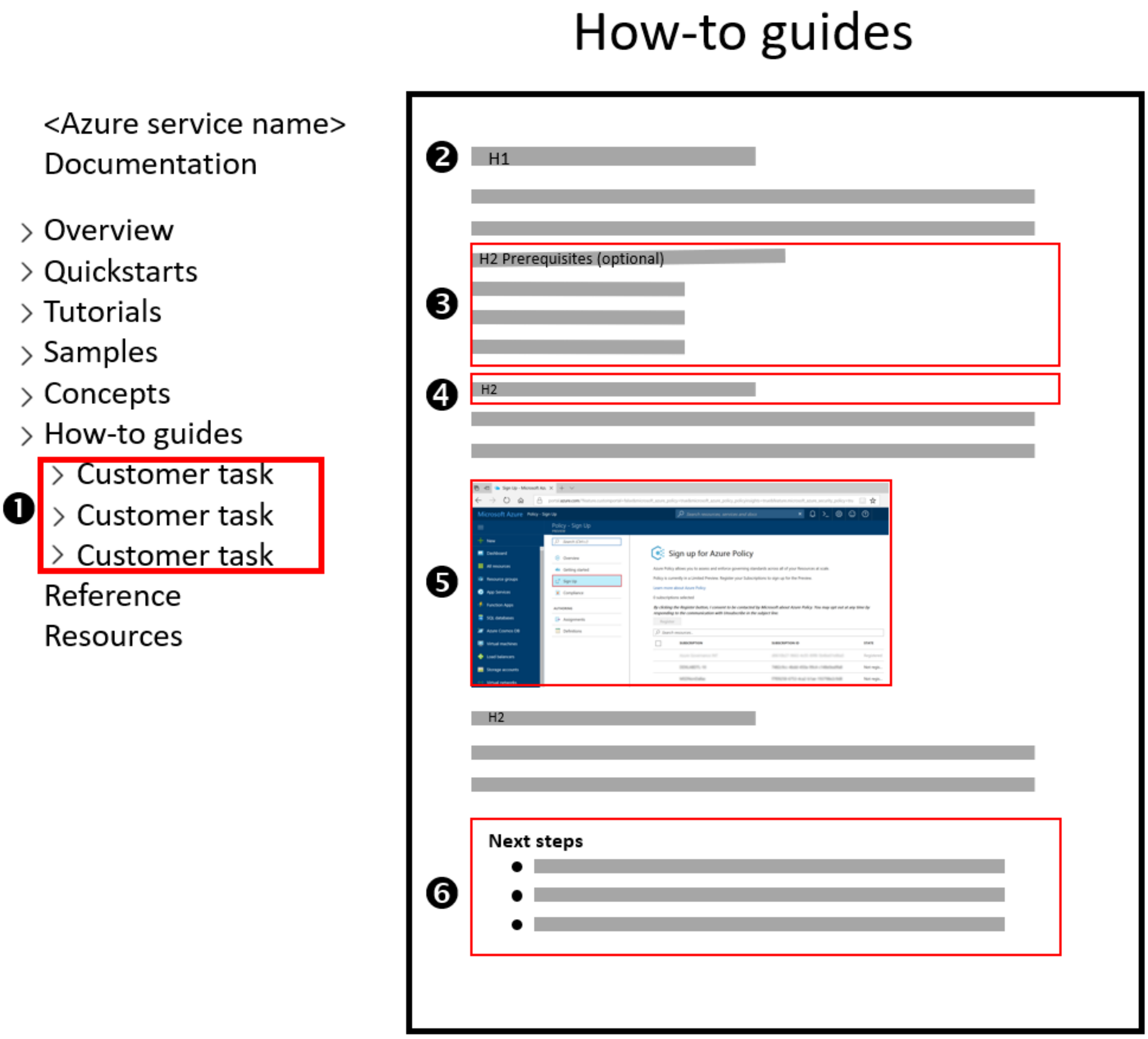
Write how-to guides

6/24/2021 • 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



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.

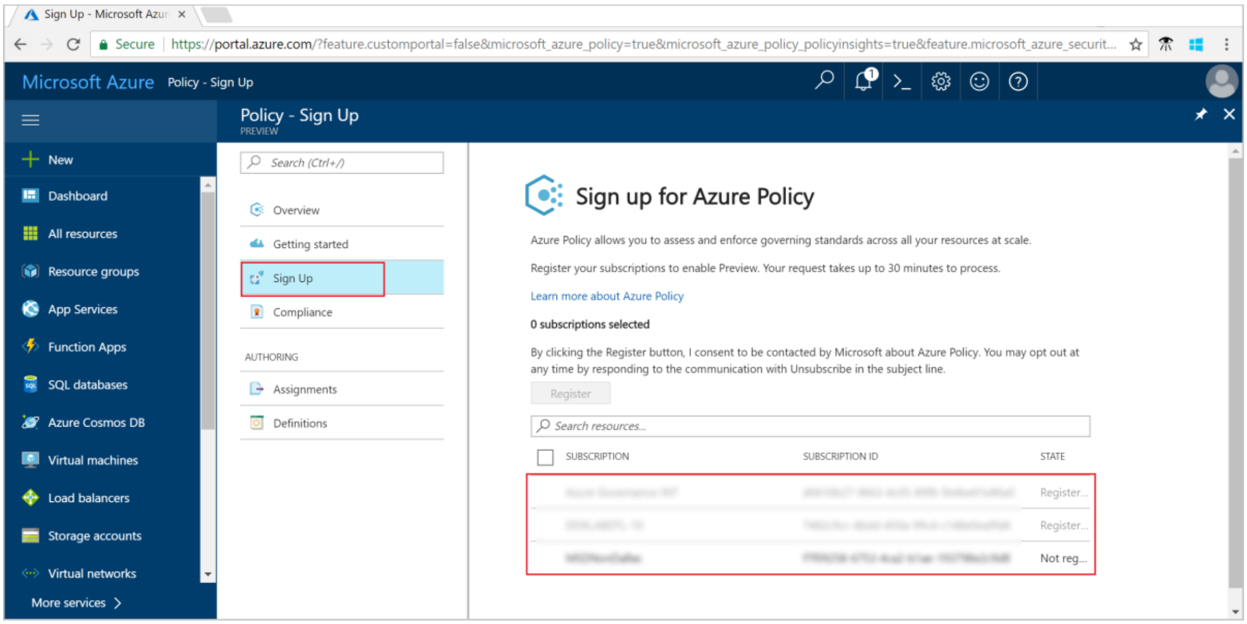
Introduction

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

6/28/2021 • 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 betters 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

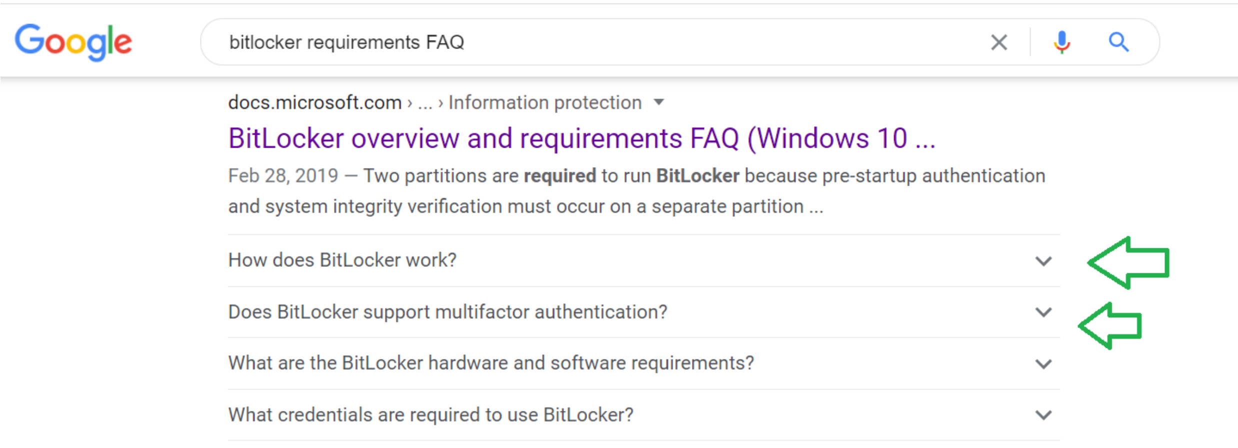
6/21/2021 • 3 minutes to read

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.

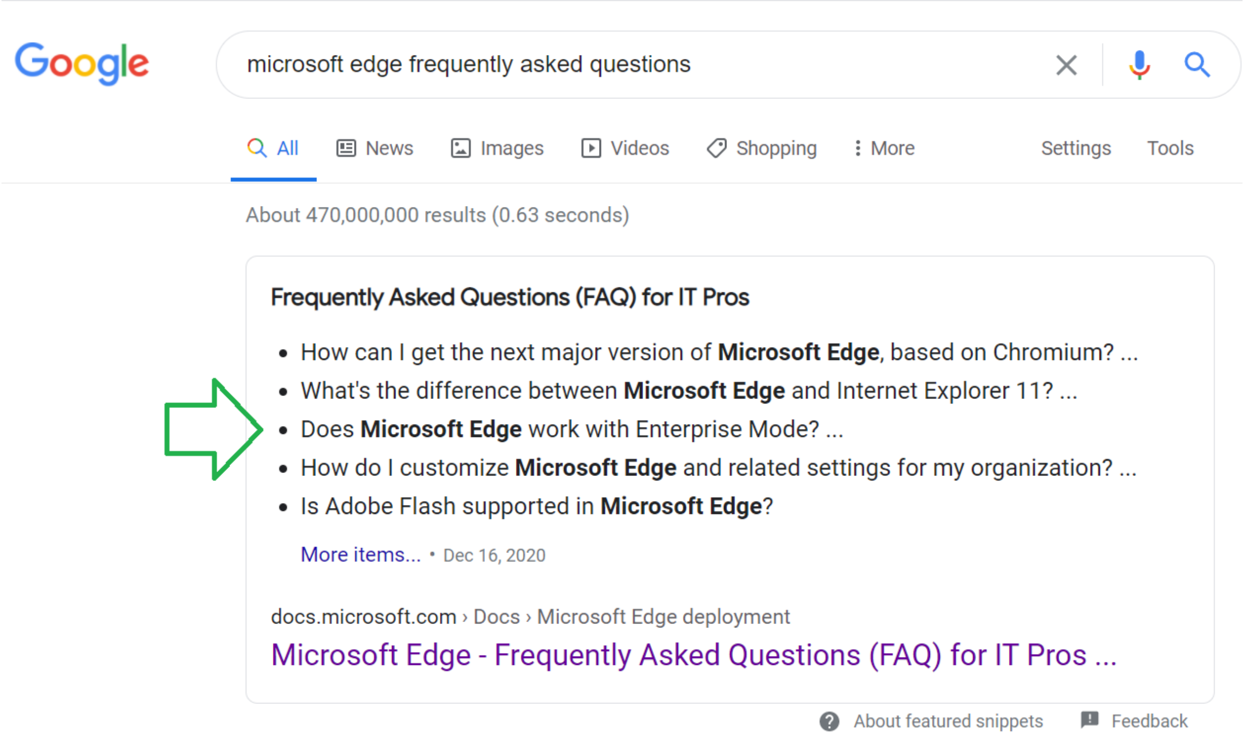
Benefits of using an FAQ template

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 state 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:



Here's an example of how FAQ content appears on a Google Featured Snippet on top position.



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:"GetanswerstocommonquestionsaboutAzureMarketplaceandMicrosoftAppSource."

title:CommonquestionsabouttheMicrosoftcommercialmarketplace
summary:Thisarticleanswerscommonlyaskedquestionsaboutthecommercialmarketplace.

sections:
-name:General
questions:
-question:WhatistheMicrosoftcommercialmarketplace?
answer:|
Thecommercialmarketplaceconnectsbusinessandgovernmentagencieswithsolutionsbuiltbyourpartners.Our
partnerscreateandmanageoffersinPartnerCenter,andcustomerscandiscoverandbuysolutions…
Tolearnmore,gotothe[Microsoftcommercialmarketplacehub]
(https://partner.microsoft.com/asset/collection/commercial-marketplace#/).

-question:WherecanIlearnmoreabouttheMicrosoftAdminCenter?
answer:ForinformationabouttheMicrosoftAdminCenter,see[Managethird-
partyappsubscriptionsforyourorganization](/microsoft-365/commerce/manage-saas-apps).

-name:Resources
questions:
-question:WherecanIfindmoreinformationaboutthecommercialmarketplace?
answer:|
Herearesomeresourcestogetyoustarted:
-[WhatistheMicrosoftcommercialmarketplace?](overview.md)
-[Microsoftcommercialmarketplacepartnerhub]
(https://partner.microsoft.com/asset/collection/commercial-marketplace#/)

additionalContent: |
##InConclusion
Here'ssomeoptionaltextthatcanbeplacedattheendofthedocument.
```

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)

7/14/2021 • 11 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's 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:

- Consolidate 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 select a zone at the top of the document. When a selection is made, any content within zone pivot code blocks will change to match that selection.	When using tabbed conceptual, the user can select a tab 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 selects a zone, the context is maintained as they navigate the developer website. For example, 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, zone pivots aren't the right choice, because zone pivots only appear once at the top of the document (consider tabbed conceptual). If the answer is yes, ask yourself these follow-up questions to inform your decision:

- Can others reuse the zone pivot configuration change you're proposing? For example, if it's a collection of programming languages, can another service use this same configuration? Or is your use case a one-off, perhaps to display features for a service? If the answer is no, your proposed zones aren't 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, this file may 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 principles are a work in progress. We'll consider 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:

- 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: Let's say you create a zone pivot group for programming languages that includes C#, Java, Python, Node.js, and Go. When you reference this group in your article, the reader will see an option to select C#, Java, Python, Node, or Go at the top of the page. 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, they should use a consistent naming convention and order of precedence. For example, the group `title` and `prompt` should match, and the pivot `id` and `title` for each entry should match. Consistent naming ensures that when readers move through the docs, the context of their previous choices follow them. *For example: Let's say a reader selects Python in an article. If you follow the rules listed here, when the reader moves to* *another article where Python is an available selection, the Python zone is automatically selected.*

- If your docset is nested inside another docset (for example, `/visualstudio/mac/` is a separate docset to `/visualstudio/`) *by default* your pages will use the zone pivots defined in the root docset (in other words, `/visualstudio/zone-pivot-groups.yml`). The same is true if your docset isn't part of a repo but inherits its URL pattern (for example, a repo that isn't part of azure-docs-pr but uses `/azure/` in its URL). If you rely on this behavior, testing on `review.docs.microsoft.com` will be tricky as the root docset won't 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 (for example, `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 to 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. 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 approach 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

NOTE

We recommend publishing the required changes to the `.openpublishing.publish.config.json` file and the zone pivot definition file, as described above, before continuing. Once your initial configuration changes are merged, you can open another pull request to enable zone pivots in your article.

After you've updated `zone-pivot-groups.yml`, the next step is to add the appropriate metadata tag to your article. Once you add the metadata tag to your article, the selector **will** render on the page, whether or not you add content to the zone pivots.

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 you want to render when that pivot is clicked should be 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. We used this approach because the sections and explanations aren't 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 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

6/24/2021 • 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:

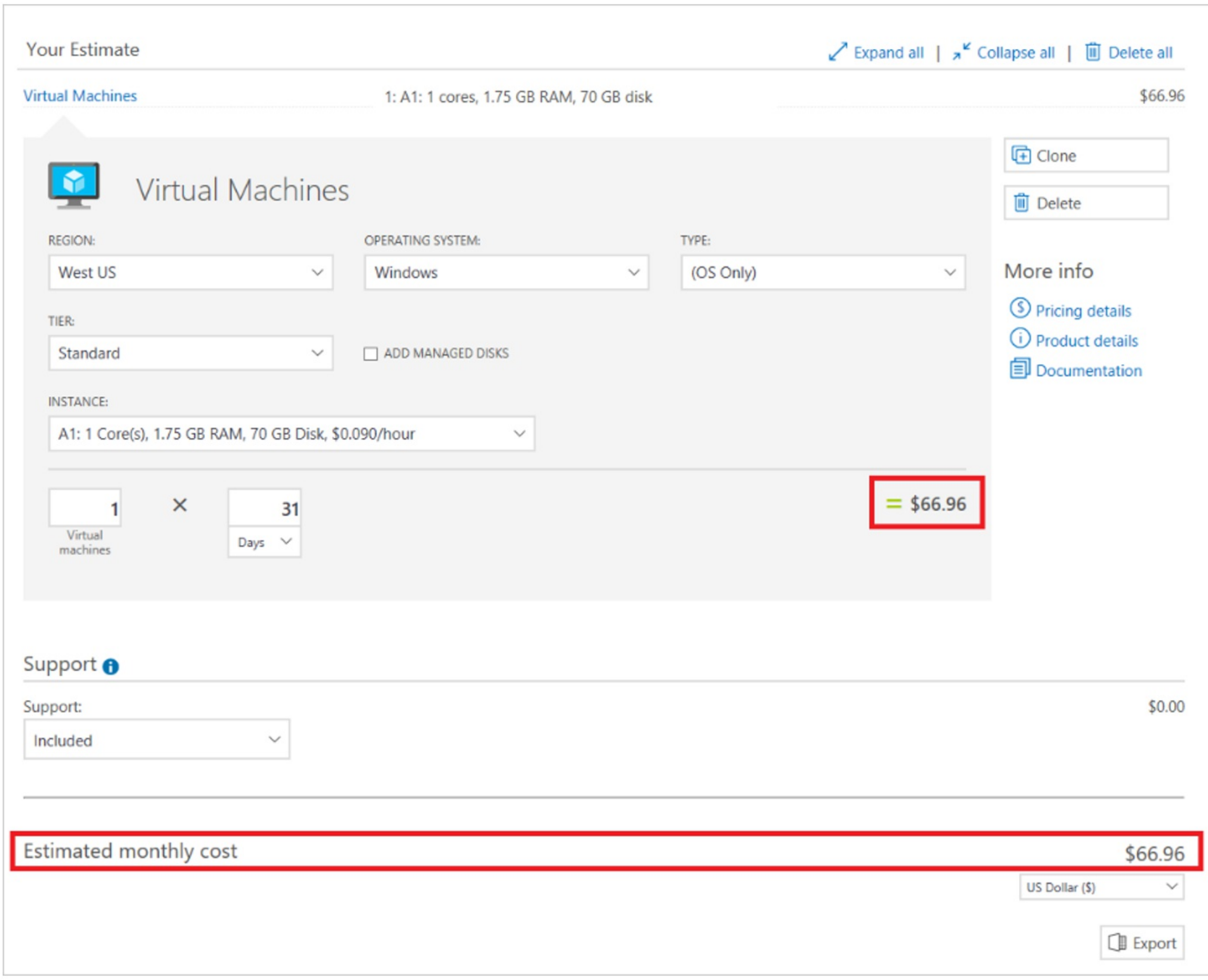
- https://docs.microsoft.com/azure/cosmos-db/plan-manage-costs
- https://docs.microsoft.com/azure/storage/common/storage-plan-manage-costs
- https://docs.microsoft.com/azure/machine-learning/concept-plan-manage-cost

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:



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:

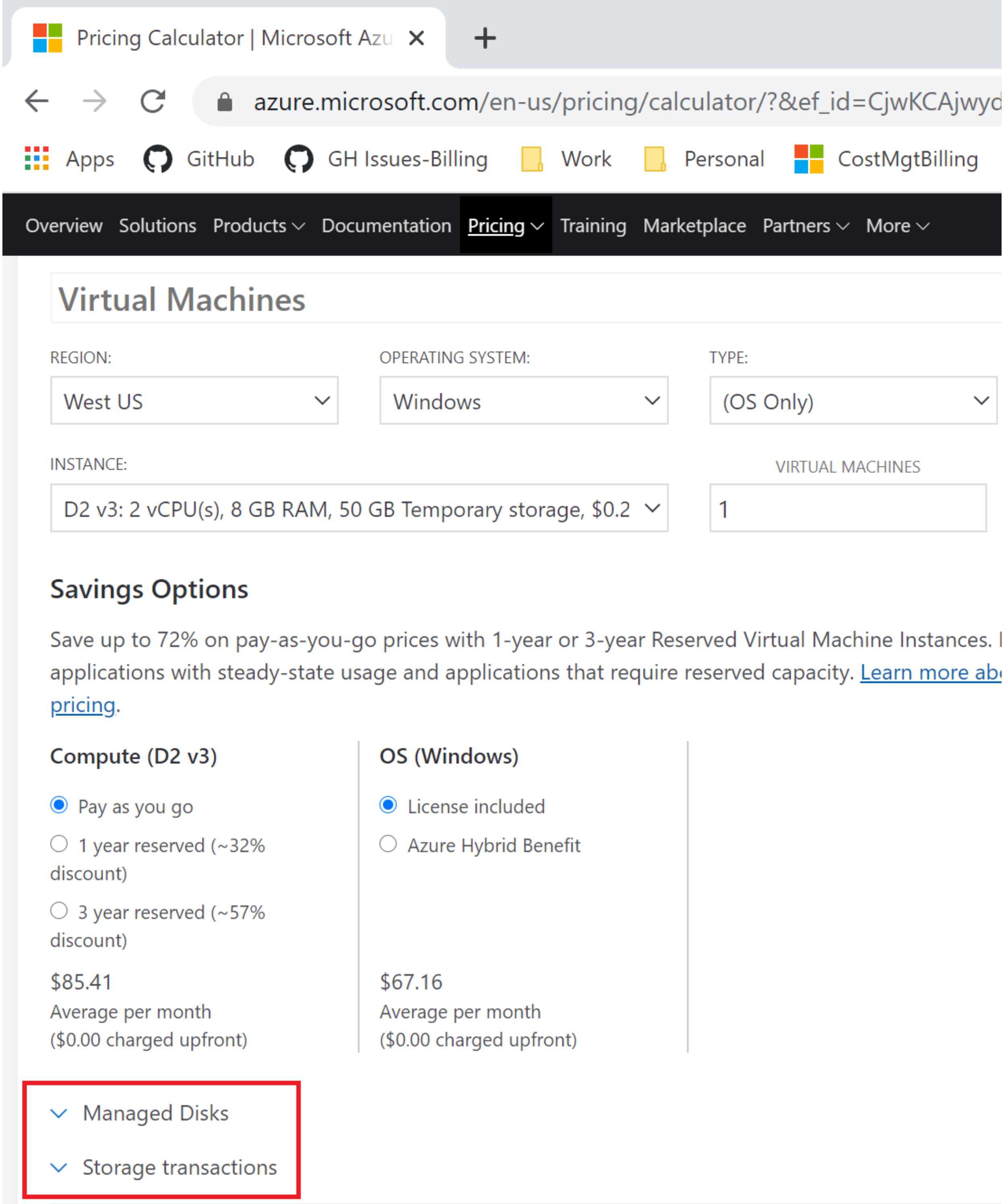
- When a user creates a resource for your service, does the new resource need *other* Azure service resources to function?
- Will the user need to manually create the other resources?
- Will the other resources get automatically created?

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.

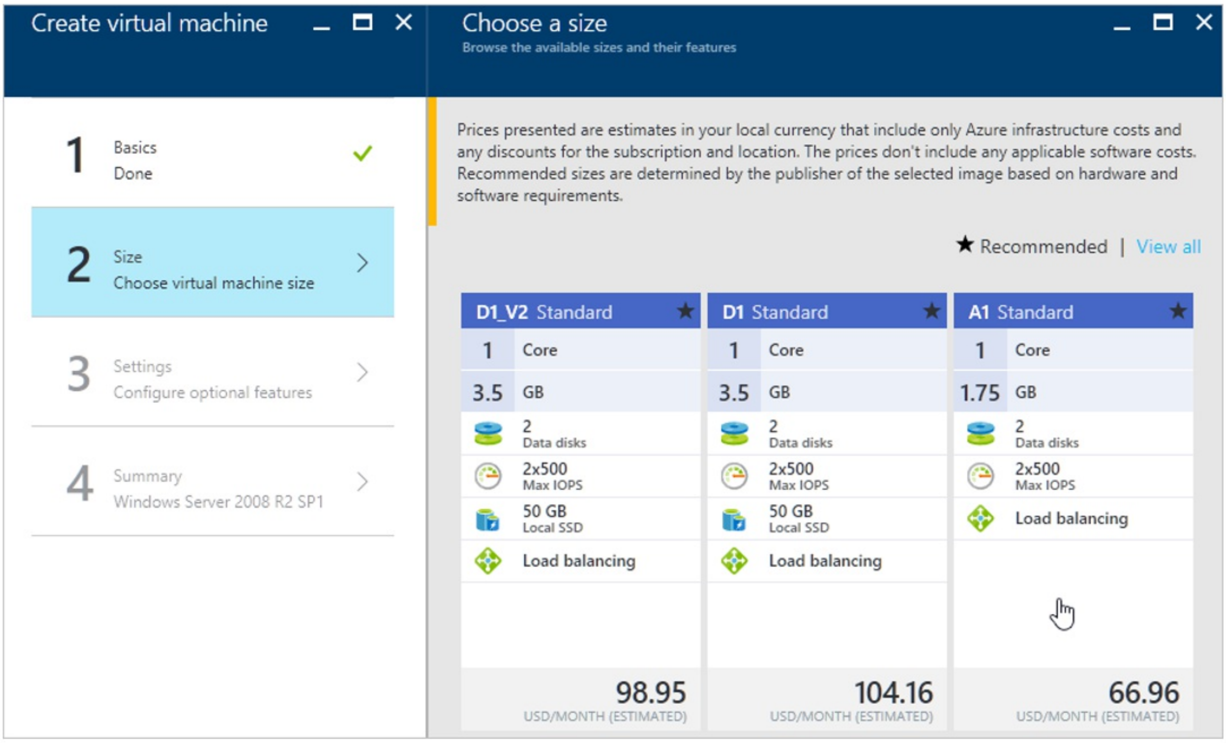


We need to let users know about the potential costs for those items. So, if applicable, add the following items to the section:

- 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.

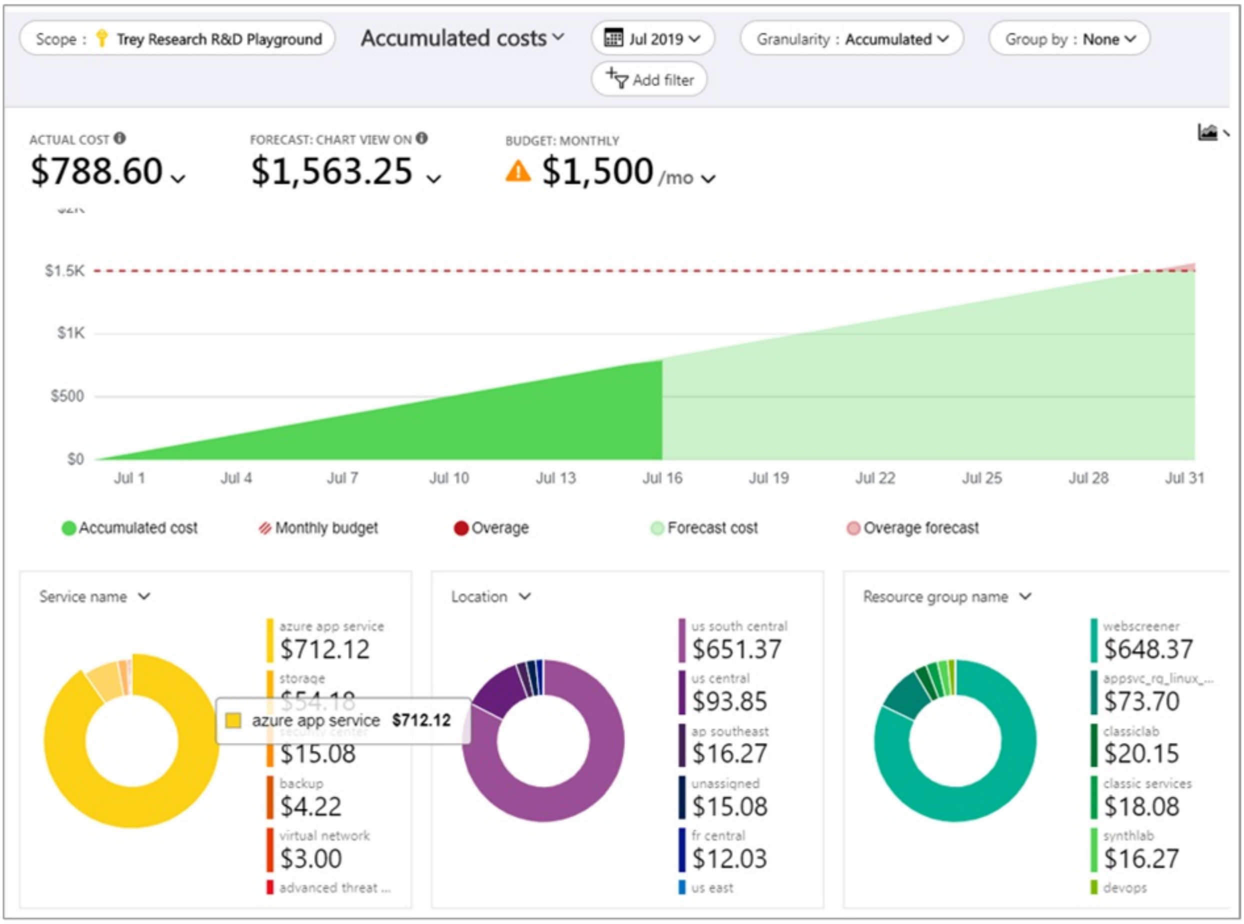
Review estimated costs in the Azure portal

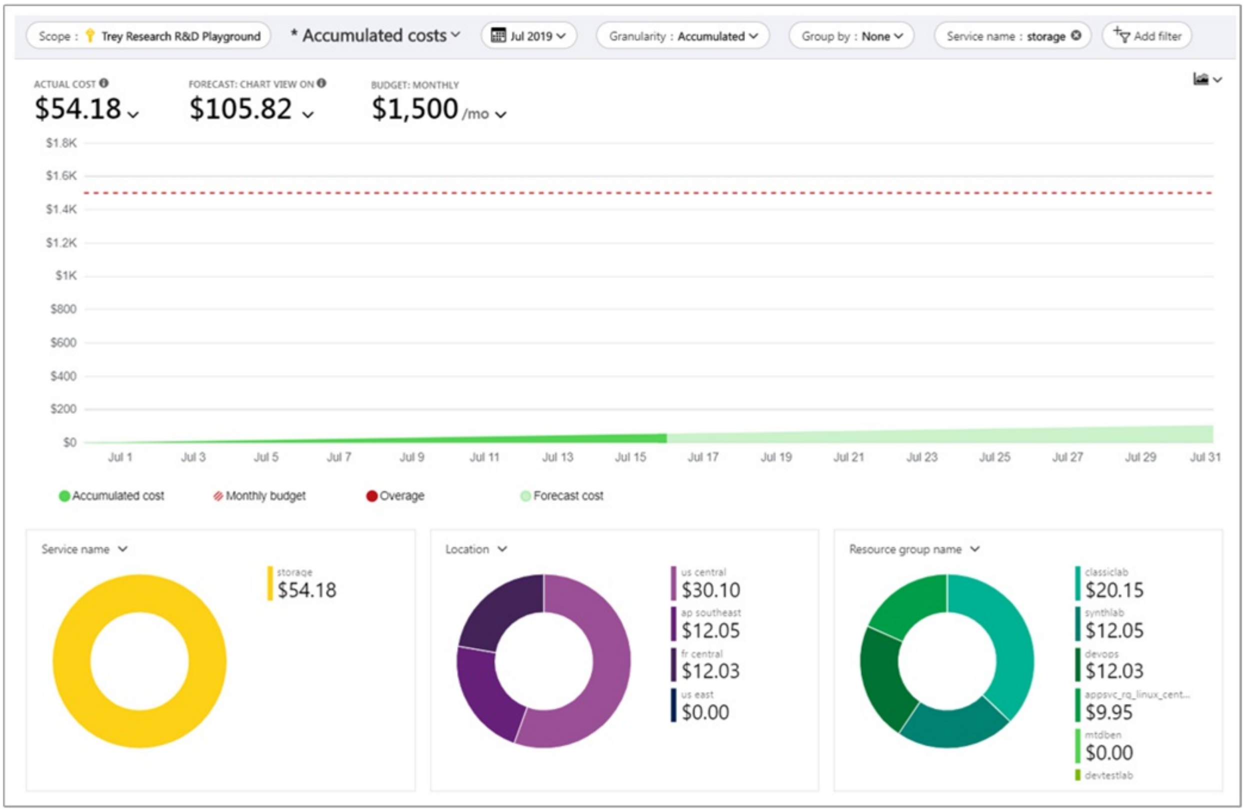
As users create resources for your service in the Azure portal, most services show estimated prices. This section, as an H2, discusses costs shown in the Azure portal when resources are created. Add an image that shows estimated costs when an Azure resource is getting created. Here's an example:



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/paths/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

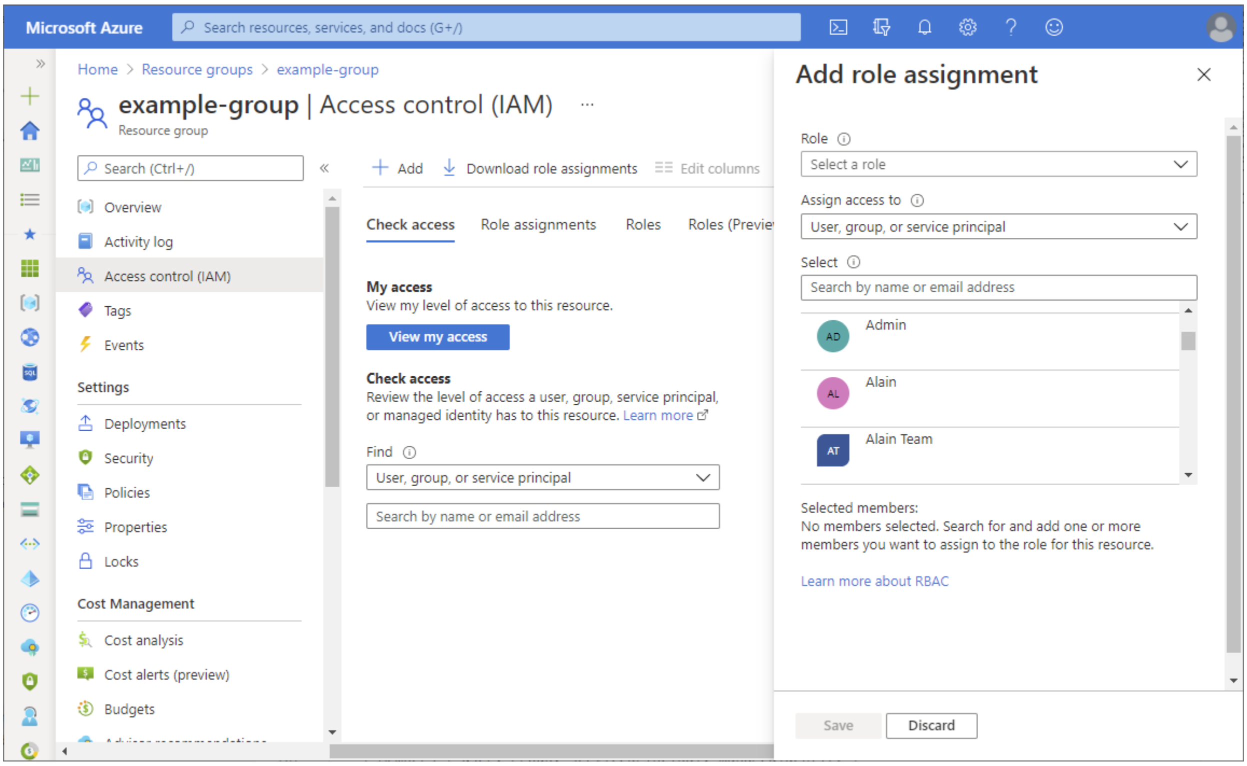
7/21/2021 • 5 minutes to read

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 your articles have steps to assign Azure roles, you'll need to update them.

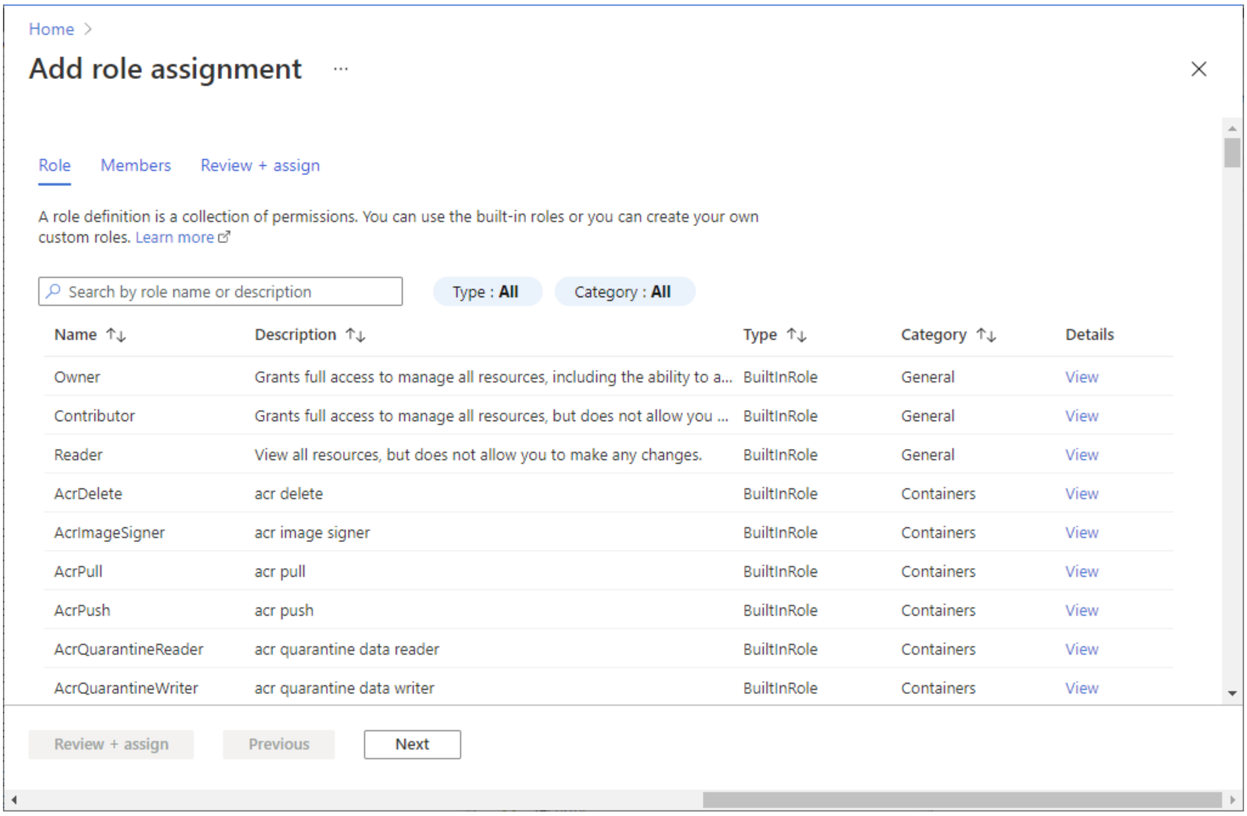
This article helps you write content to assign roles for Azure RBAC.

What is changing?

The current role assignment experience uses a pane:



The new role assignment experience uses a full-screen page with tabs:



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 following spreadsheet or query to find which of your docs might be affected.
   - Azure RBAC steps spreadsheet (See the **RBACSteps** tab)
   - Azure RBAC horizontal content query

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 general availability (GA) date is still **to be determined**.

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. For a published example, see Create, list, delete, or assign a role to a user-assigned managed identity using the Azure portal.

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)

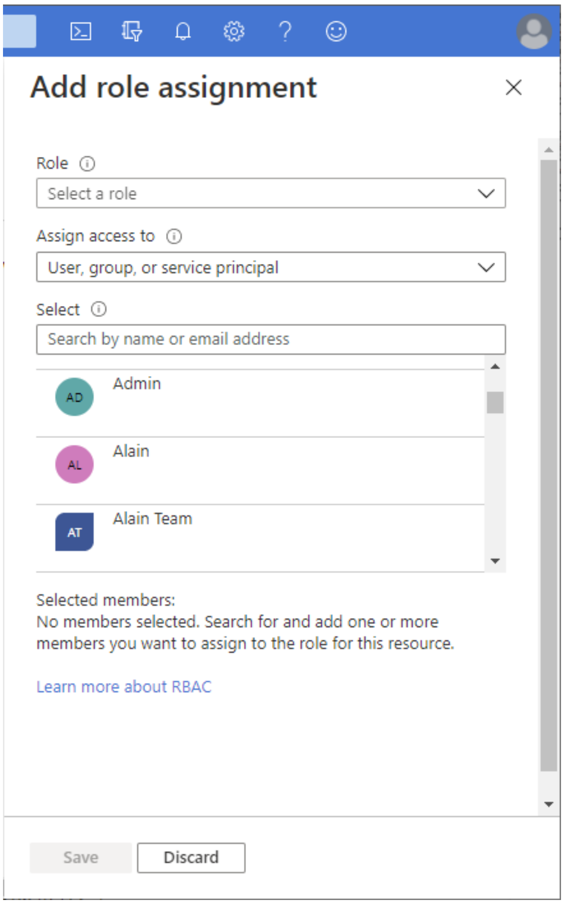
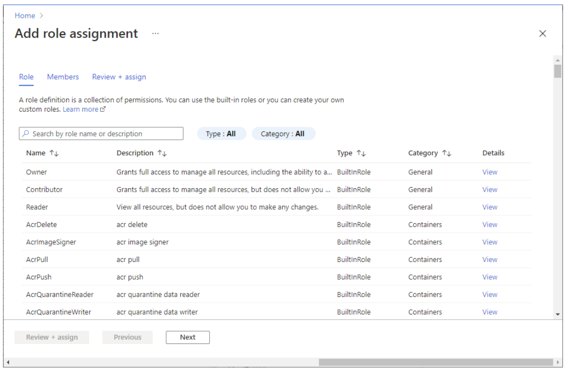
To make it easier to maintain your docs, we recommend removing the role assignment steps and linking to the Azure RBAC docs. For a published example, see Tutorial - Configure Azure Active Directory to issue verifiable credentials.

> **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

<table>
<tr><th>BEFORE GA</th><th>AFTER GA</th></tr>
<tr><td>
</td><td>
</td></tr>
</table>

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 | [ROLENAME] |
    | 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:

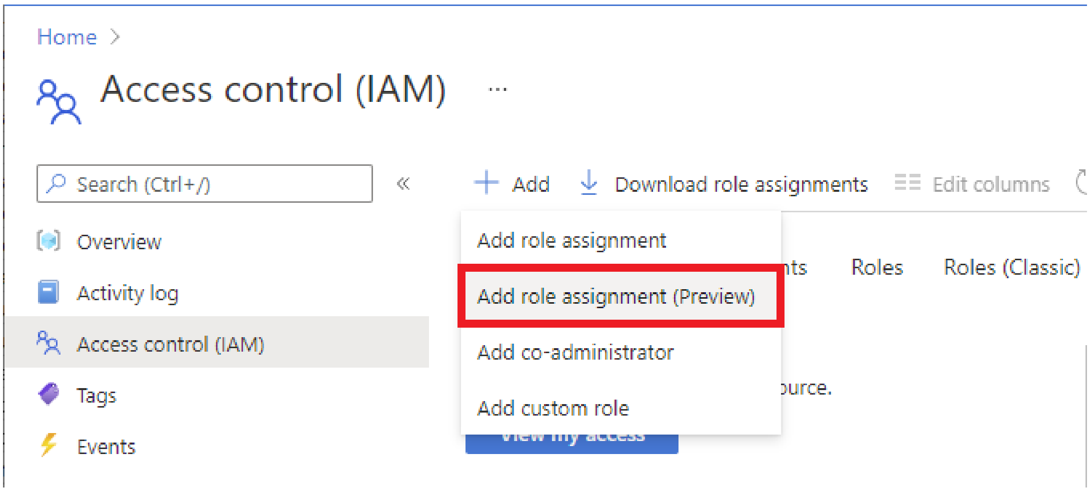
The following steps describe how to assign the Virtual Machine Contributor role to the Alain user at the example-group resource group scope. For detailed steps, see Assign Azure roles using the Azure portal.

1. Sign in to the Azure portal as User Access Administrator or Owner.

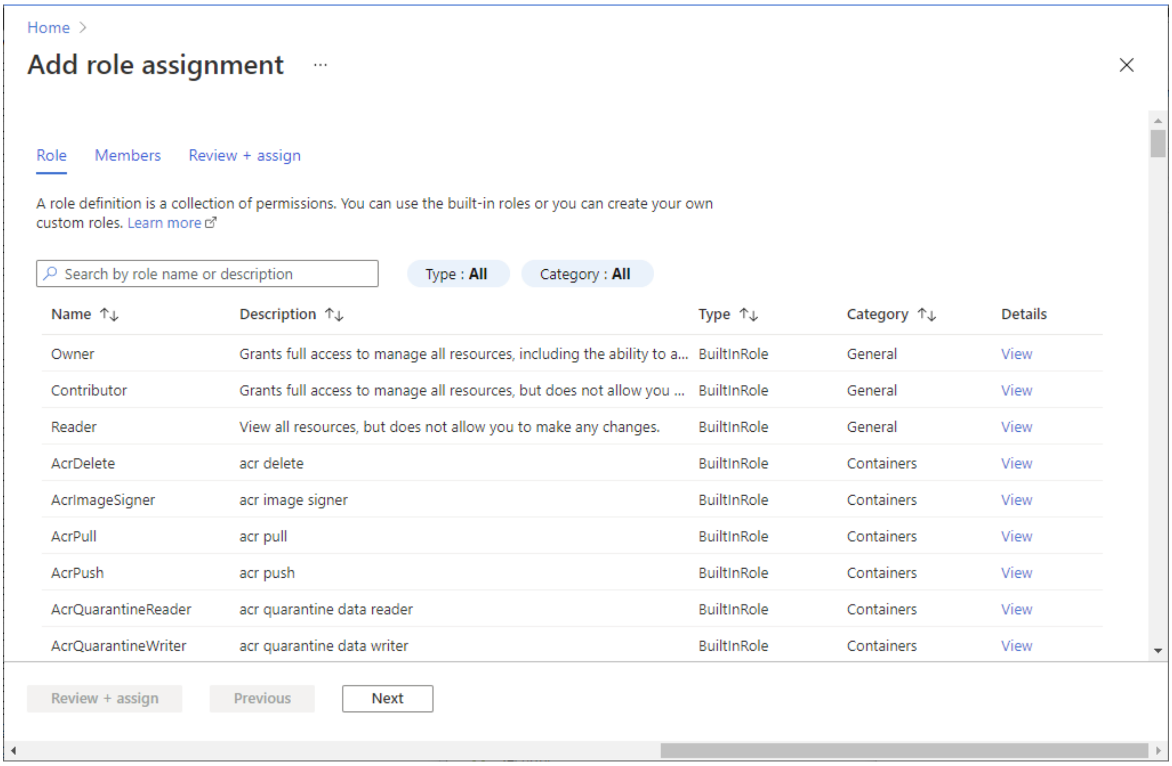
2. Open the example-group resource group.

3. In the navigation menu, select **Access control (IAM)**.

4. Select **Add** > **Add role assignment (Preview)**.



5. On the **Role** tab, select the **Virtual Machine Contributor** role.



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 [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).

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 **[ROLENAME]** 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
Azure built-in role -or- built-in role	built-in RBAC role
Azure custom role -or- custom role	custom RBAC role

For other RBAC systems, use the following terminology:

RECOMMENDED TERM	DESCRIPTION
Azure AD role	When referring to Azure Active Directory roles.
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 Managed HSM.
Kubernetes RBAC	When referring to Kubernetes RBAC.
Service Fabric RBAC	When referring to Service Fabric RBAC.
vSphere RBAC	When referring to vSphere RBAC.

Questions

If you have any questions or need help, contact Robert Lyon.

How to write about managed identities for Azure resources

7/8/2021 • 4 minutes to read

You may have received a request from your engineering team to help them create content for your service. This could be due to Enterprise promise documentation requirements. You can get more information here.

If you find a conflict with the guidance in How to write how-to guides - Docs Contributor Guide follow the contributor guide guidance and let us know.

General guidance

- Avoid content duplication and link to existing documentation when discussing managed identities features. If you find a content gap, email barclayn@microsoft.com.
- Call out implementation details unique to your service. For example, the way to get tokens using Managed Identities with Virtual Machines vs Azure Kubernetes Service vs Azure Functions is slightly different. So, each service should separately document how to get tokens in their articles.
- If you use scenarios, then we suggest you show how the managed identity is able of performing operations that would otherwise require the use of other less secure ways of authenticating access.

Some examples:

- How to use the Azure SDK and Azure. Identity library for accessing the target service.
- How to acquire AAD tokens using a managed identity
- How to then make the data-plane API call using the AAD token.

TOC

You need a TOC entry under How-to guides that includes the words **Managed identities**

Metadata

ATTRIBUTE	VALUE
ms.topci	how-to
ms.custom	ep-msia

H1

The H1 should include the words **Managed identities**

Introduction

- The first paragraph should include the words **managed identities**
- You should also have a pointer to

```
[!INCLUDE [managed identities](../../includes/managed-identities-definition.md)
```

The include file contains a brief definition of managed identities and a pointer to the managed identities overview.

Template

We're providing a template using tab conceptual to help you write your managed identities article. If you have template feedback write

```
---
title: Use managed identities in [Service name] | Microsoft Docs #Required; page title is displayed in search results.
description: Learn how to use managed identities with [Service name] using the Azure Portal, CLI, PowerShell, Azure resource manager template  #Required; article description that is displayed in search results.
services: active-directory
documentationcenter: ''
author: barclayn
manager: daveba
editor: ''

ms.service: active-directory
ms.subservice: msi
ms.workload: integration
ms.topic: how-to
ms.date: 07/07/2021
ms.author: barclayn
ms.custom: ep-msia

---

# How to use managed identities with [service name]

<!--
Remove all the comments in this template before you sign-off or merge to the main branch.
-->

This article shows you how to use managed identities with a [service name] instance.` [!INCLUDE [managed identities](../../includes/managed-identities-definition.md)] `.

## Prerequisites

- <!-- prerequisite 1 -->
- <!-- prerequisite 2 -->
- <!-- prerequisite n -->
<!-- remove this section if prerequisites are not needed -->

## Create an instance of [service name] with a system assigned managed identity

<!-- Show the steps required to create an instance of your service with a system assigned managed identity enabled using as many of the options show below as possible. You can see additional tab options https://review.docs.microsoft.com/en-us/help/contribute/validation-ref/tabbed-conceptual?branch=master -->

# [Portal](#tab/azure-portal)


# [PowerShell](#tab/azure-powershell)

# [Azure CLI](#tab/azure-cli)

# [Resource Manager Template](#tab/azure-resource-manager)

---
```

```
## Assign a user-assigned managed identity to a service instance

User-assigned managed identities can be used on multiple resources. To learn more about managed identities,
for information on how to create or delete user-assigned managed identities you can review [Manage user-
assigned managed identities](https://docs.microsoft.com/azure/active-directory/managed-identities-azure-
resources/how-manage-user-assigned-managed-identities?pivots)

<!-- Show the steps required to create an instance of your service using a user-assigned managed identity --
>

# [Portal](#tab/azure-portal)

# [PowerShell](#tab/azure-powershell)

# [Azure CLI](#tab/azure-cli)

# [Resource Manager Template](#tab/azure-resource-manager)


---

## Supported scenarios using managed identities

<!-- If there are differences between the support available for user assigned managed identities versus
system assigned managed identities call that out here-->

<!--

### User assigned managed identities


### System assigned managed identities


-->

### Grant access to a managed identity

<!--If customers need to grant access to managed identity so they may interact with data hosted by your
service include the steps here.

We recommend that you follow the RBAC guidance to grant the managed identities the necessary permissions to
access the resource because then you do not need to worry about maintaining these steps as part of your
documentation. Some examples are shown below:

Example 1:
“Using managed identities for Azure resources, your code can get access tokens to authenticate to resources
that support Azure AD authentication.  [Service] supports Azure AD authentication.  Grant the managed
identity access by assigning the [Role](../role-based-access-control/built-in-roles.md#role) role to the
managed-identity at the scope of the resource group that contains your [resource instance].

For detailed steps, see [Assign Azure roles using the Azure portal](../role-based-access-control/role-
assignments-portal.md).”

Example 2:

## Grant your VM's system-assigned managed identity access to use a storage SAS

Azure Storage does not natively support Azure AD authentication.  However, you can use your VM's system-
assigned managed identity to retrieve a storage SAS from Resource Manager, then use the SAS to access
storage.  In this step, you grant your VM's system-assigned managed identity access to your storage account
SAS. Grant access by assigning the [Storage Account Contributor](../role-based-access-control/built-in-
roles.md#storage-account-contributor) role to the managed-identity at the scope of the resource group that
contains your storage account.

For detailed steps, see [Assign Azure roles using the Azure portal](../role-based-access-control/role-
assignments-portal.md).”
```

```
>[!NOTE]
> For more information on the various roles that you can use to grant permissions to storage review
[Authorize access to blobs and queues using Azure Active Directory](../../storage/common/storage-auth-aad.md#assign-azure-roles-for-access-rights)

-->

## Clean up steps

<!--- Include the steps that customers should follow to clean up their environment --->

## Next steps

Learn more about managed identities for Azure resources:

* [What are managed identities for Azure resources?](https://docs.microsoft.com/azure/active-directory/managed-identities-azure-resources/overview)
* [Azure Resource Manager templates](https://github.com/Azure/azure-quickstart-templates)


<!--
Remove all the comments in this template before you sign-off or merge to the
main branch.
-->
```

Monitoring content for Azure services

7/12/2021 • 7 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) 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 vs. something sevice specific.

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
Monitoring <Service Name>	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 below.
Monitoring <Service Name> data reference	monitor-<service-name>-data-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 below.
Common and service-specific schema for Azure Resource Logs	resource-logs-schema.md	Azure Monitor	Azure monitor team. **Links updated by you.**	**UPDATE THIS ARTICLE.** 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 service specific schema section. If such an article(s) do not already exist, put the schema information into ***Monitor <servicename> data Reference*** 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	monitor-azure-resource.md	Azure Monitor	Azure Monitor content team	An article containing common information for services onboarded to Azure Monitor. Included in the *Monitoring <Service Name>* template text already. See more information below.	**No action**

TITLE	NAME	TOC LOCATION	COMPLETED BY	DESCRIPTION	ACTION
Resource providers / types relevant to your service	azure-services-resource-providers.md	Resource Manager	Resource Manager team.	A mapping of service to namespace (resource provider and type)	**You check**. 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	metrics-supported.md	Azure Monitor	Azure Monitor content team	Azure monitor team maintains this article. It's automated and pulled from backend systems.	**You check**. Verify that this article contains your metrics and refer to them in the reference article you write 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 (robb@microsoft.com) if you need more information.
Supported_categories for Azure Resource Logs	resource-logs-categories.md	Azure Monitor	Azure Monitor content team	Azure monitor team maintains this article. It's automated and pulled from backend systems.	**You check**. Verify that this article contains your log categories and refer to them in the reference article you write via bookmarks.
Resource provider operations	Resource Provider Operations	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	activity-log-schema.md	Azure Monitor	Azure Monitor content team	A standardized reference to entires added to the activity log.	No action.

TITLE	NAME	TOC LOCATION	COMPLETED BY	DESCRIPTION	ACTION
Azure Monitor Logs / Log Analytics Table Reference	Many articles	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 ***Monitor <servicename >*** or ***Monitor <servicename > data Reference*** 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.
- **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 specific to your 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. Examples include screenshots of the metrics explorer for metrics of the service or a command-line examples for configuring diagnostic settings.
- **Prescriptive guidance** - Every service has different requirements on 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

6/29/2021 • 10 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 - MORE INFO	DO THIS
First-party Microsoft image from Docker Hub	`microsoft/dotnet:2.1-runtime`	Replace with an equivalent image in Microsoft Container Registry - more info
Third-party image from Docker Hub	`node:10.15.0`	Substitute MCR image, or add mitigation guidance - more info
Docker Hub image metadata only (name, tags, etc.)	`https://hub.docker.com/_/mysql`	No change if the image isn't consumed
Image from Microsoft Container Registry	`mcr.microsoft.com/dotnet/core/sdk:2.1-stretch`	No change
Image from another public source other than Docker Hub	`public.ecr.aws/bitnami/tensorflow-resnet:latest`	Optionally substitute MCR image, or add mitigation guidance - more info
Private image pushed/pulled to a private Docker Hub repository	`mydockerid/myspecialimage:tag`	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` and can include:

- 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
`docker pull` or `docker run` command	`docker run mcr.microsoft.com/azure-sql-edge`
`FROM` statement in a Dockerfile	`FROM mcr.microsoft.com/azure-functions/python:2.0-python3.7-appservice`
`image:` statement in Docker Compose or Kubernetes deployment file	`image: mcr.microsoft.com/oss/nginx/nginx:1.15.5-alpine`

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. Microsoft images will be deprecated in Docker Hub starting July 2021.

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. See the draft repo mapping and 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.
- 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. Can your engineering team onboard the image to MCR?
4. If MCR isn't the solution, then see How to add mitigation guidance for public images

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	`https://mcr.microsoft.com/v2/_catalog`

List tags for a specific repository	`https://mcr.microsoft.com/v2/<repoName>/tags/list`
	
	Example:
	`https://mcr.microsoft.com/v2/dotnet/framework/aspnet/tags/list`

> **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**.

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, request your engineering team to onboard it. Refer your team to the detailed MCR onboarding steps.

The Microsoft Open Source team maintains certain common open source images in MCR for use by Microsoft engineering teams. If you use an image that you think meets their criteria, reach out to Jonathan Chauncey.

How to add mitigation guidance for public images

If an article or sample must use a public non-Microsoft container image from Docker Hub, add guidance to alert the customer of use of the image, possible reliability issues, and recommended mitigation. Choose one of the following options:

Option 1: Minimal guidance - alert when using a public image

When an article uses a non-Microsoft public image in a technical step, add a note immediately above the step to alert the customer and link to general guidance for consuming public images. The goal is to raise customer awareness with minimal interruption to article flow.

Add or adapt one of the following includes:

- Include for docker pull

  [!NOTE] The following example pulls a public container image from Docker Hub. We recommend that you authenticate with your Docker Hub account (docker login) first instead of making an anonymous pull request. To improve reliability when using public content, import and manage the image in a private Azure container registry. Learn more about working with public images.

- Include for Dockerfile

  [!NOTE] The Dockerfile used in the following example depends on a public base container image from Docker Hub. To improve reliability when using public content, import and manage the image in a private Azure container registry, and update your Dockerfile to use your privately managed base image. Learn more about working with public images.

- Include for Kubernetes deployment (not yet created)

  [!NOTE] The following example pulls a public container image from Docker Hub. We recommend that you set up a pull secret to authenticate using a Docker Hub account instead of making an anonymous pull request. To improve reliability when working with public content, import and manage the image in a private Azure container registry. Learn more about working with public images.

Option 2: Preferred guidance - add steps to use a private ACR

We recommend guiding customer to host a copy of any required public container image in a private Azure container registry. Adding this guidance helps customers maintain a specific image in an environment they control and maintain and avoid known risks when depending on public content.

Depending on the scenario, include some or all of the following guidance. Exact steps depend on article flow and scenario, so choose the steps and order that make sense for you.

- 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. Consider making these updates during a freshness update to existing content. 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
- Tutorial: Configure a Jenkins environment to enable CI/CD for a Java application on Service Fabric - alert for `docker pull` example
- Quickstart: Send events from private container registry to Event Grid - alert for Dockerfile with public base 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

7/7/2021 • 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

7/7/2021 • 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	**Optional:** 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 **How-to guides** -> **Troubleshoot**.
Problem resolution	**Optional:** 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 **How-to guides** -> **Troubleshoot** -> **Problem resolution**.
Known issues	**Optional:** 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 **How-to guides** -> **Troubleshoot**.
Error code lists	**Optional:** 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 **Reference**.

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 troublshooting 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 decribing 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

6/24/2021 • 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.

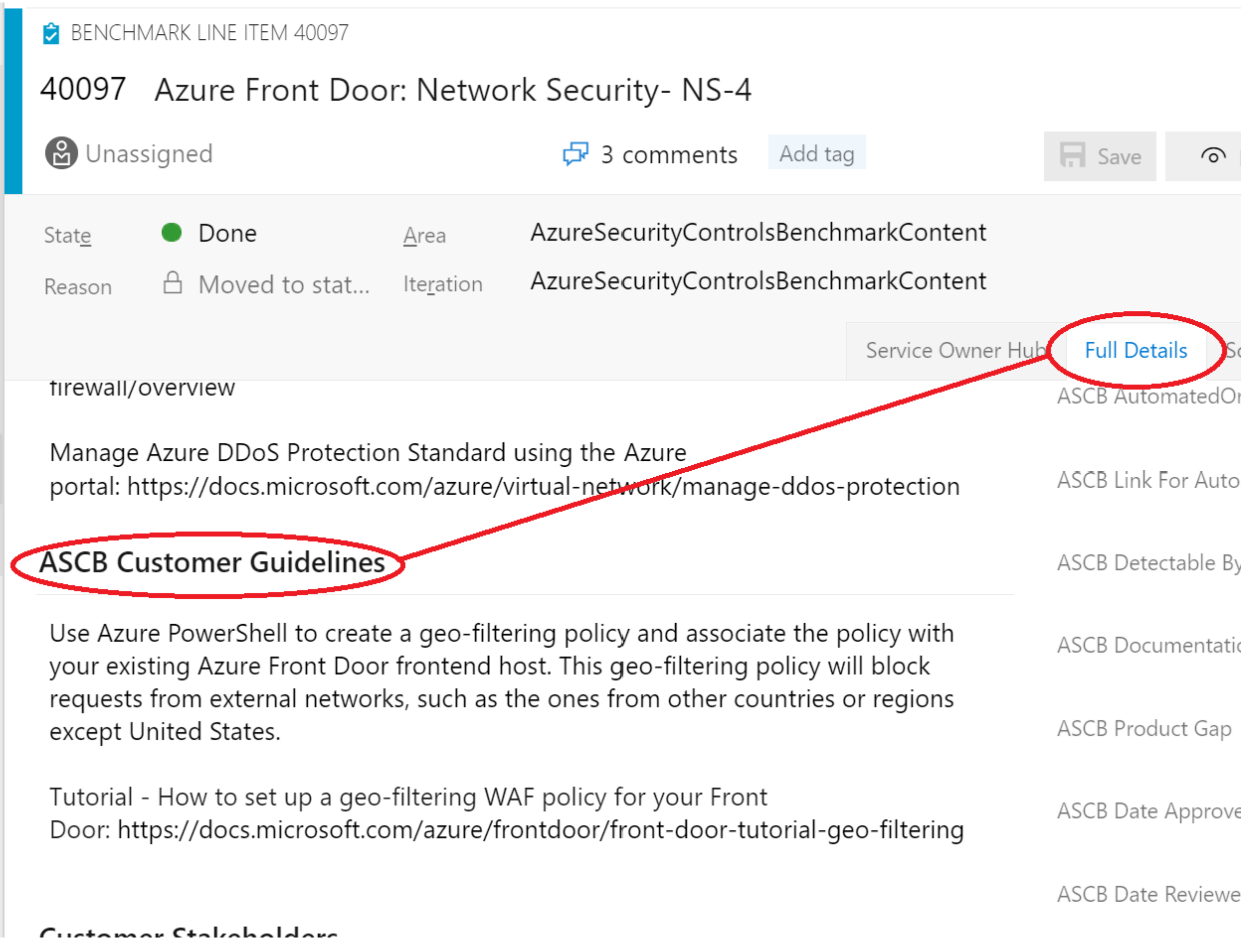
Published baselines only include the sections relevant to the service. Subcontrols which are not applicable are omitted.

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.



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: Yes

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 you 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

7/19/2021 • 5 minutes to read

"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.

STRATEGY	USE WHEN
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 *OR* 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 *AND* You have more than 1-2 best practices for each feature area *AND* 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 *AND* You have more than 1-2 best practices for each feature area *AND* 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

7/9/2021 • 2 minutes to read

Documentation in large repos like 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. These mechanisms 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 Corporate, External, and Legal Affairs (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. Funding those translations is also the product team's responsibility. 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 isn't product or service specific, use `ms.prod: legal`
  - All files should use the generic `ms.topic: article`

We recommend that all files use another metadata value: `layout: ContentPage`. It creates a flat-page experience with no TOC. The exception to the rule is groups who want to create a contextual TOC to make legal articles appear within the TOC of a product or service guide.

Don't create a regular TOC within the DocsLegal-pr repo. The root TOC and breadcrumbs are out of context and will confuse readers.

Publication process

Follow these steps to contribute source content to DocsLegal-pr, which publishes to the en-us docs.microsoft.com site.

1. Draft your Markdown article 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're ready for the content to go live.

   - If you require special (timed) or rushed publication, don't #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 isn't ready to be merged. It's your responsibility to 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's important that the English content is published **before** you move forward with updates to translated content.

Legal guidelines

7/14/2021 • 4 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.

Third-party logos, trademarks, images, and other intellectual property

Reference: CELA
It is never OK to use a third party's logo, trademark, image, or other intellectual property, unless:

1. Microsoft has explicit permission to use a third party's trademarks and IP assets as part of an agreement, or as a separate written permission received from the third party, or
2. Our proposed use is permitted by the third party's online trademark guidelines, or
3. Your CELA Contact has reviewed and approved your proposed use.

What are third-party trademarks, logos, and other intellectual property?

Reference: CELA
Business names, names associated with products and services, name acronyms, logos, launch icons, marketing taglines, slogans, certification marks, the "look and feel" of a product (for example, an image of an iPad tablet), and even smells and colors.

This list is not exhaustive, but covers the types of third-party trademarks and images that we typically use.

Review of your proposed use of third-party trademarks, logos, images, and other intellectual property

If your pull request (PR) contains such third-party assets, you must have explicit, documented approval for your proposed use of the assets.

- If you already have documented approval, such as an email from your CELA Contact, show it when the PR Review Team or leadership team asks for it.

> **IMPORTANT**
> Previous CELA approval to use third-party logos, trademarks, images, and other intellectual property in other Microsoft properties doesn't count as explicit approval for your proposed use of those third-party assets.

Get CELA approval for your proposed use of third-party trademarks, logos, images, and other intellectual property

If you need CELA approval in a PR for your proposed use of such third-party assets, create an email that includes screen captures of the third-party trademarks, logos, images, and intellectual property in context, and a link to your PR, send it to your CELA Contact, and ask for approval to use the assets. Include the PR Review Team (techdocprs@microsoft.com) on your email.

For more information, see CELA Copyright.

If you have a question, reach out to your CELA Contact.

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.

> **IMPORTANT**
>
> 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

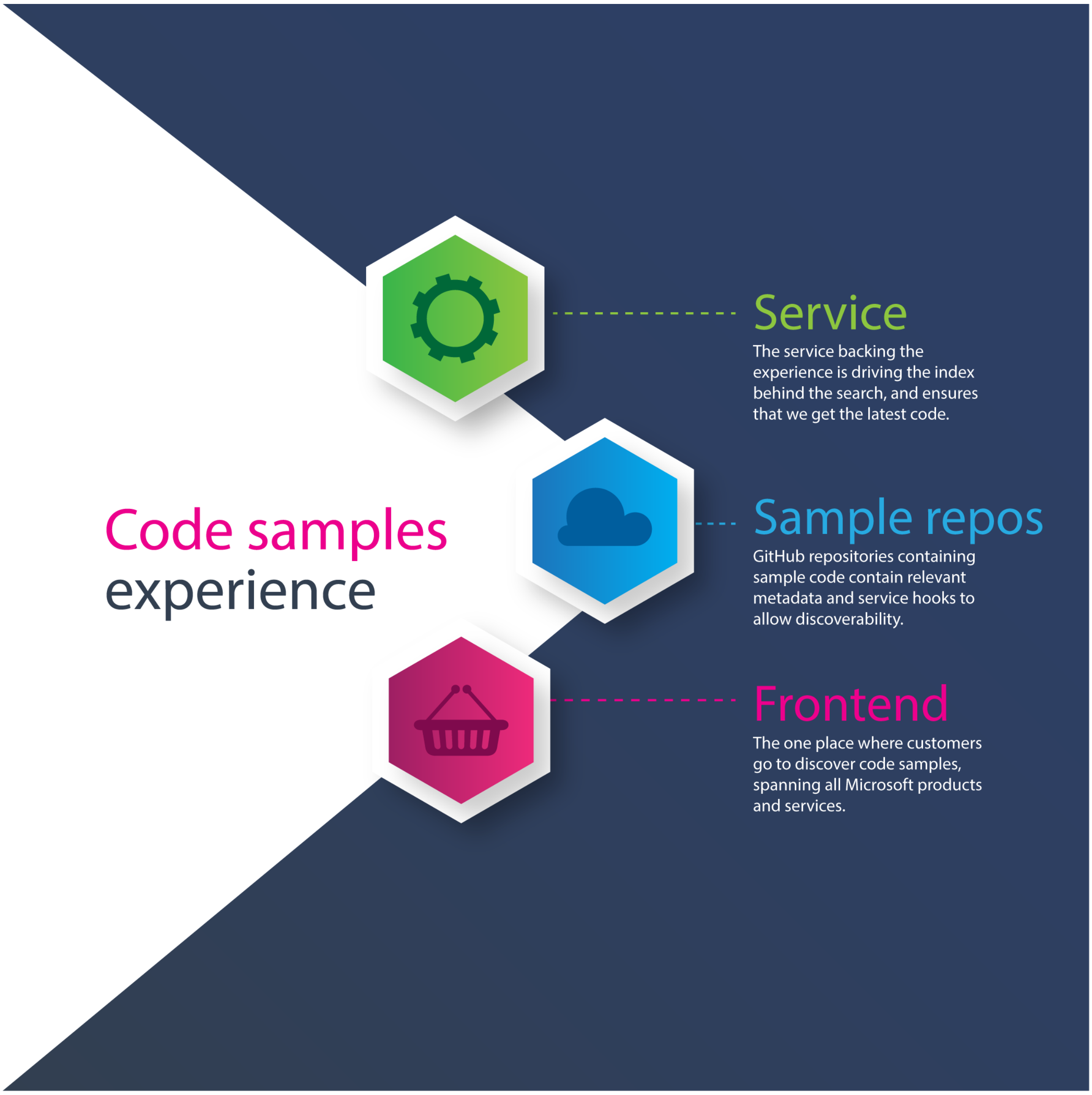
6/24/2021 • 3 minutes to read

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.

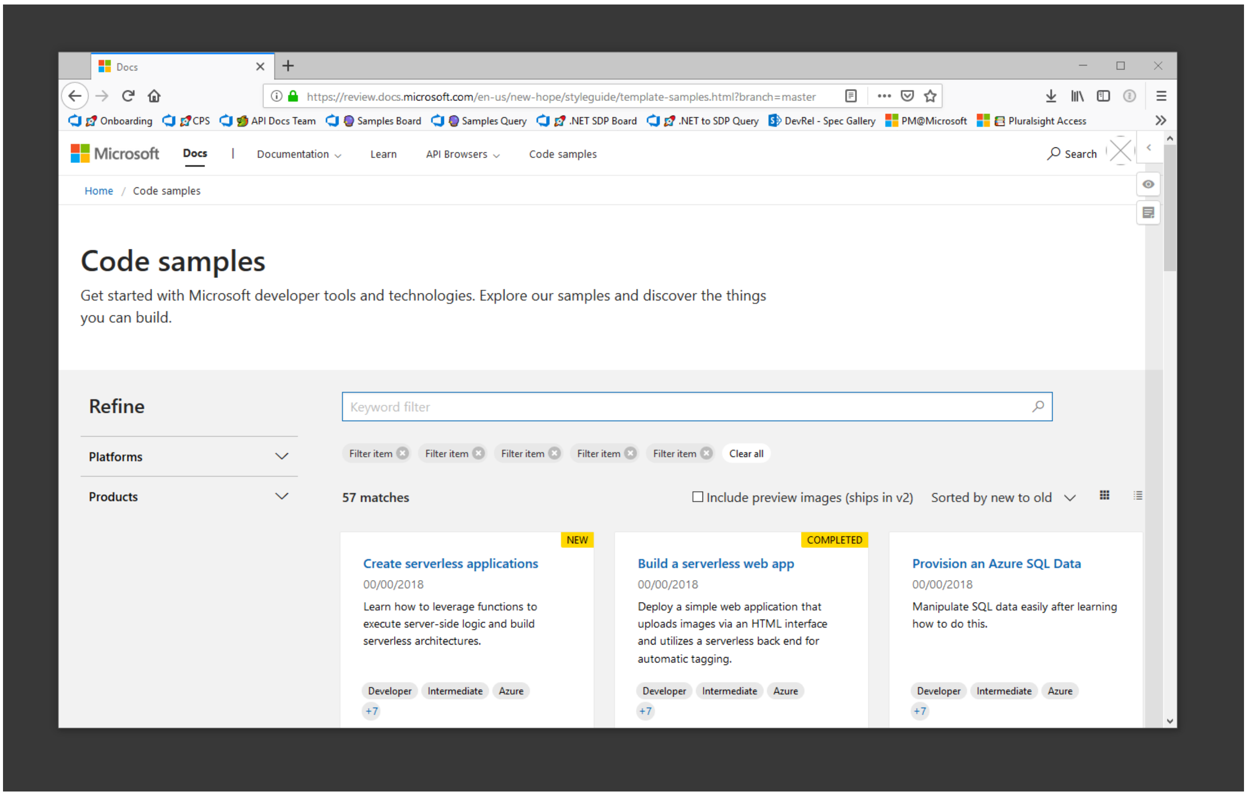
Each sample repository that is onboarded to the code samples experience has:

- 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.

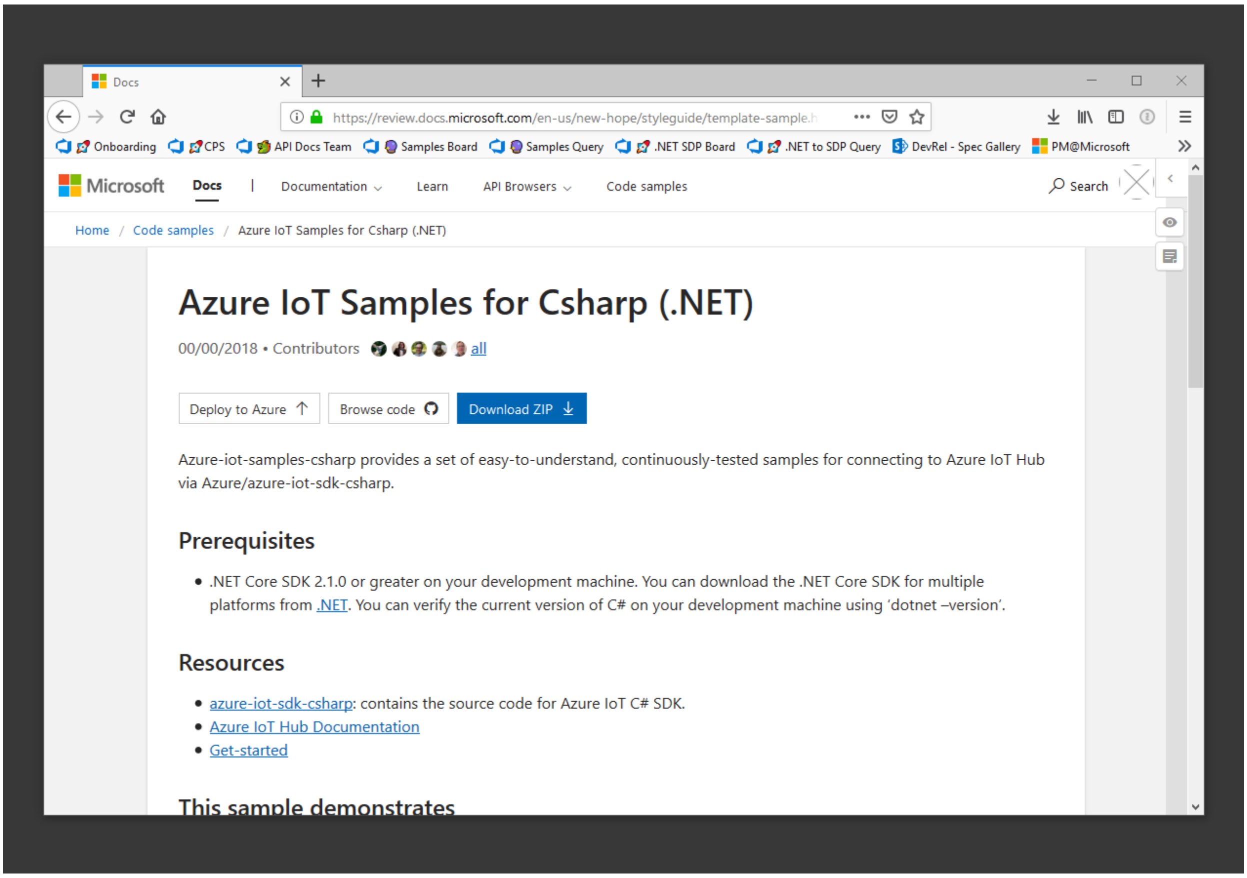
> **IMPORTANT**
>
> We do not support onboarding code samples from personal repositories or from organizations that are not approved by the Open Source Programs Office at this time.

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.



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:



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*).

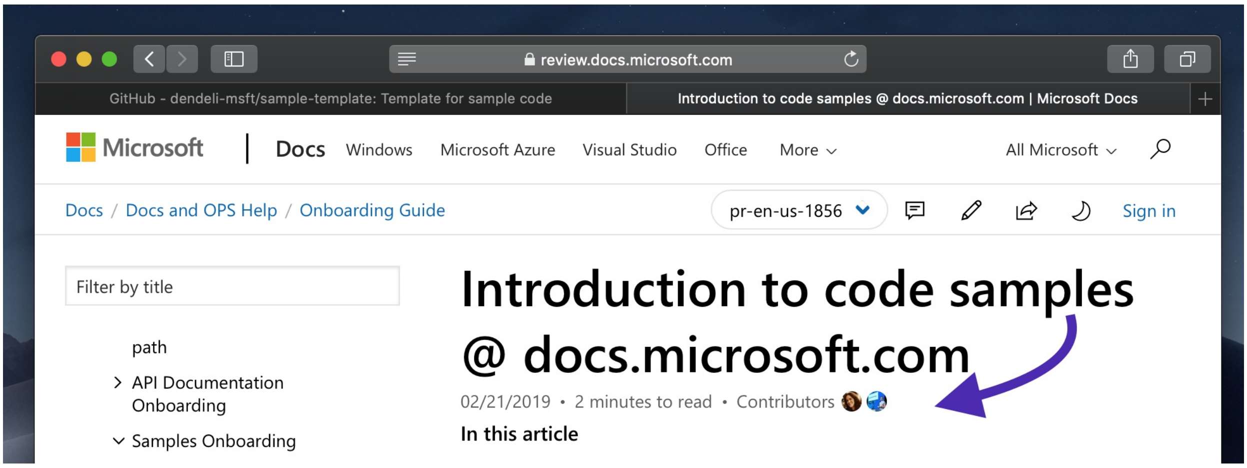
Onboarding

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

6/24/2021 • 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 *Issues* and *Pull Requests*.
`README.md`	See Sample README Template to learn about the standard sample template. 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 `README.md` files can be created for different samples within the same repository. Each `README.md` file in the repository, that covers an individual sample or set of samples should have the required metadata integrated to allow discovery in the samples browser. Ideally, the `README.md` file also contains rich media, such as static images or GIFs, that showcase the final output to the user, if applicable.
`LICENSE`	Code sample license. See information about licensing 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.

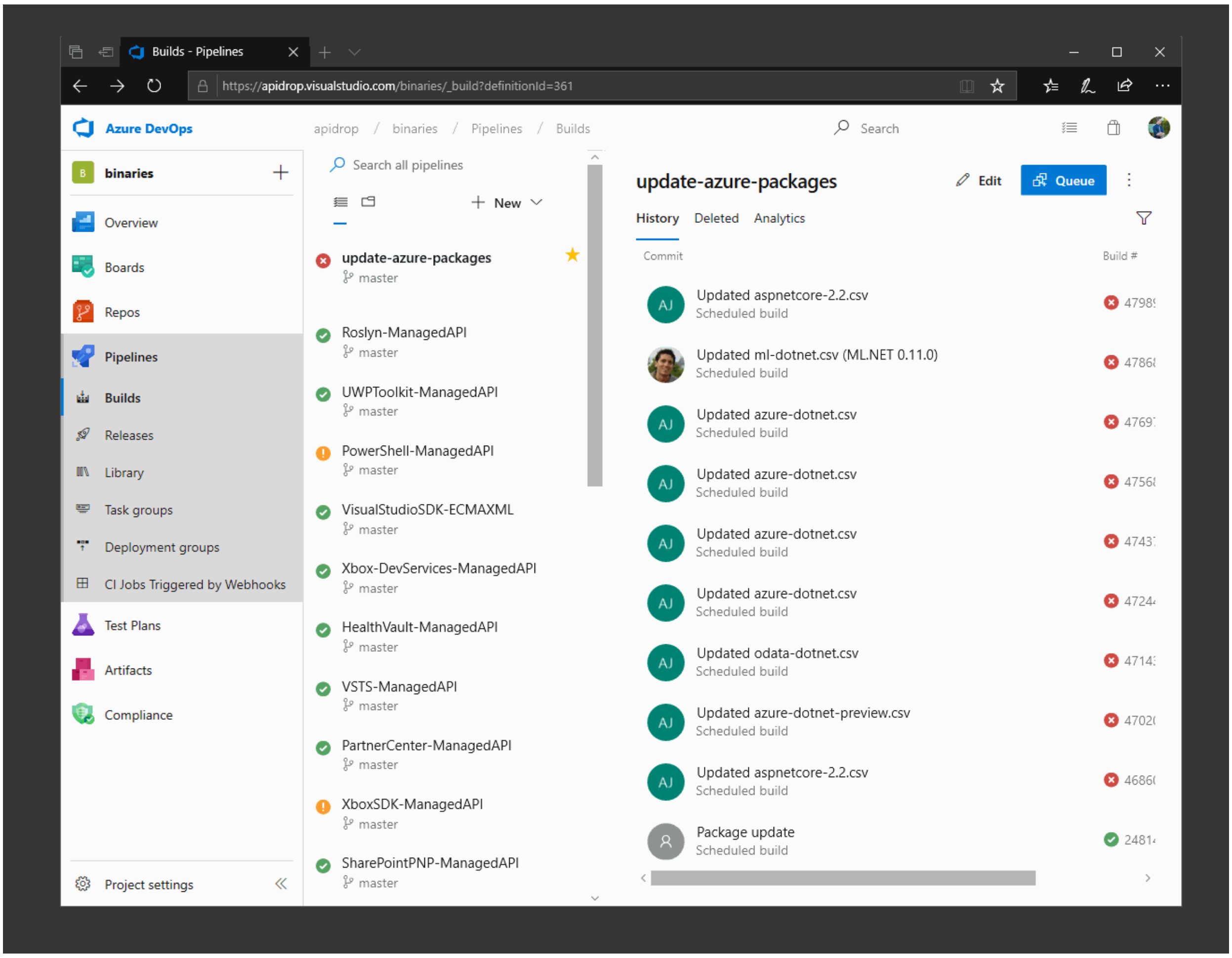
Make sure to keep your repositories structured in a way that allows customers to navigate them purely by following the folder names. Having descriptive file and folder names also makes maintenance easier.

> **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	https://github.com/azure/azure-docs-cli-python-samples (Store the scripts in a folder that's specific to your service.)

SCRIPT TYPE	REPO
Azure PowerShell	https://github.com/azure/azure-docs-powershell-samples (Store the scripts in a folder that's specific to your service.)
Resource Manager templates	We encourage teams to use https://github.com/Azure/azure-quickstart-templates. 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.

Next steps

- Create and publish sample code

Sample README Template

7/6/2021 • 12 minutes to read

This document outlines the requirements for authoring the `README.md` file for code samples developed by Microsoft and hosted on GitHub.

Why does the README.md matter?

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. For more information on writing READMEs, read The Art of the README before proceeding with the guidelines below.

The `README.md` is also the part of samples repo most likely to rank in search. Search discovery and the browse experience are important for any `README.md` onboarded to the Microsoft Docs Samples browser, where it's rendered in HTML as a doc article.

General guidelines

- A `README.md` should use simple language. Avoid acronyms, overly technical terms, and slang.
- A `README.md` title/Heading 1 and description must provide adequate context for organic search, site search, social media posts, and browse: What do the samples show and for which products?
- 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 using 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

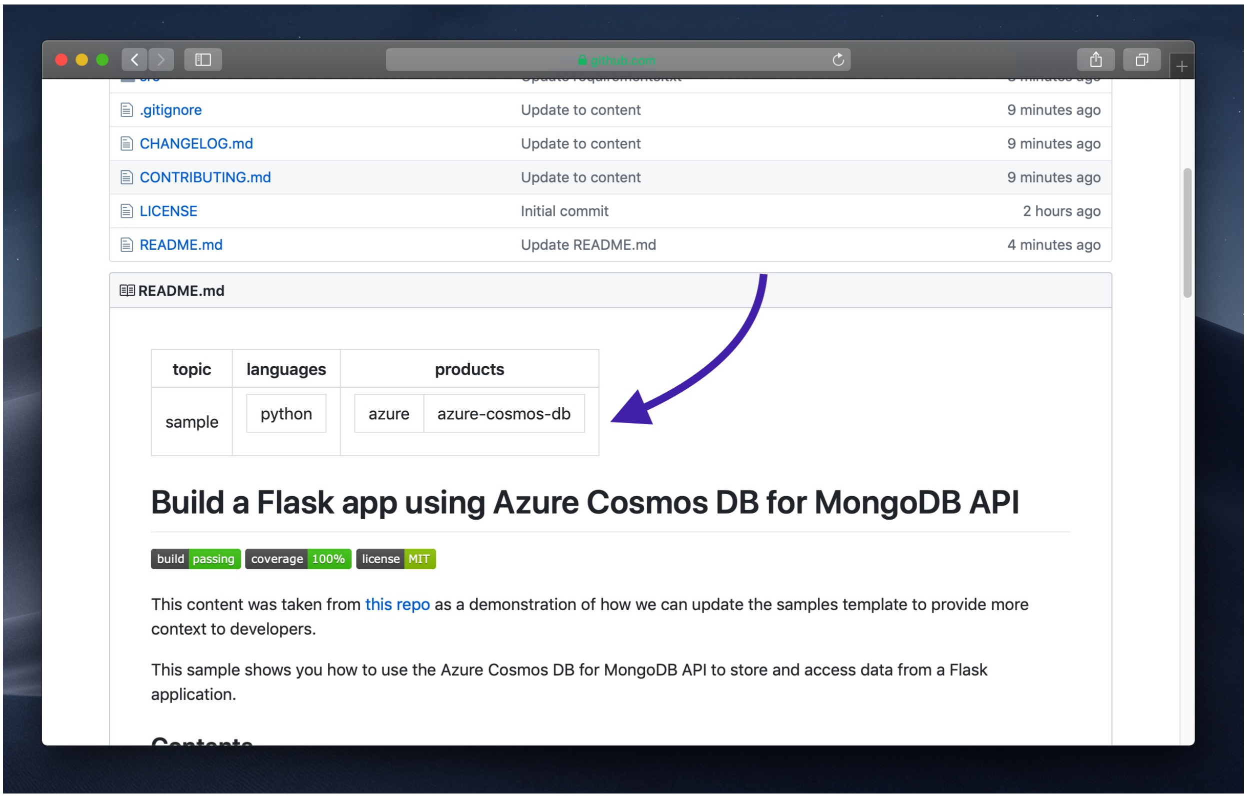
Every `README.md` should be a good standalone entity - both on GitHub as well as a rendered page on docs.microsoft.com/samples. If a customer finds the file on GitHub, they should be able to do everything to get started with just that file as the context. If they find the sample in the samples browser, 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:

- 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.
- Specify an explicit `description` value. It should be between 100- 150 characters including spaces, and should give a quick overview of what the sample is about. In absence of this metadata, the publishing system attempts 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/Heading 1 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/Heading 1

The Heading 1 immediately follows the metadata and is tagged with `#` in Markdown. The H1 renders as:

- A title for search, site search, social media, and the browse card on Microsoft Docs.
- The on-page heading at the top of the published page.

Guidance:

- Describe the intent of the sample and be sure to include relevant info, such as product brand or developer language.
- Preview how it will render on a search results page using the Moz title tool. It's OK if it truncates, as long as the relevant and intent are clear.
- 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/Heading 1

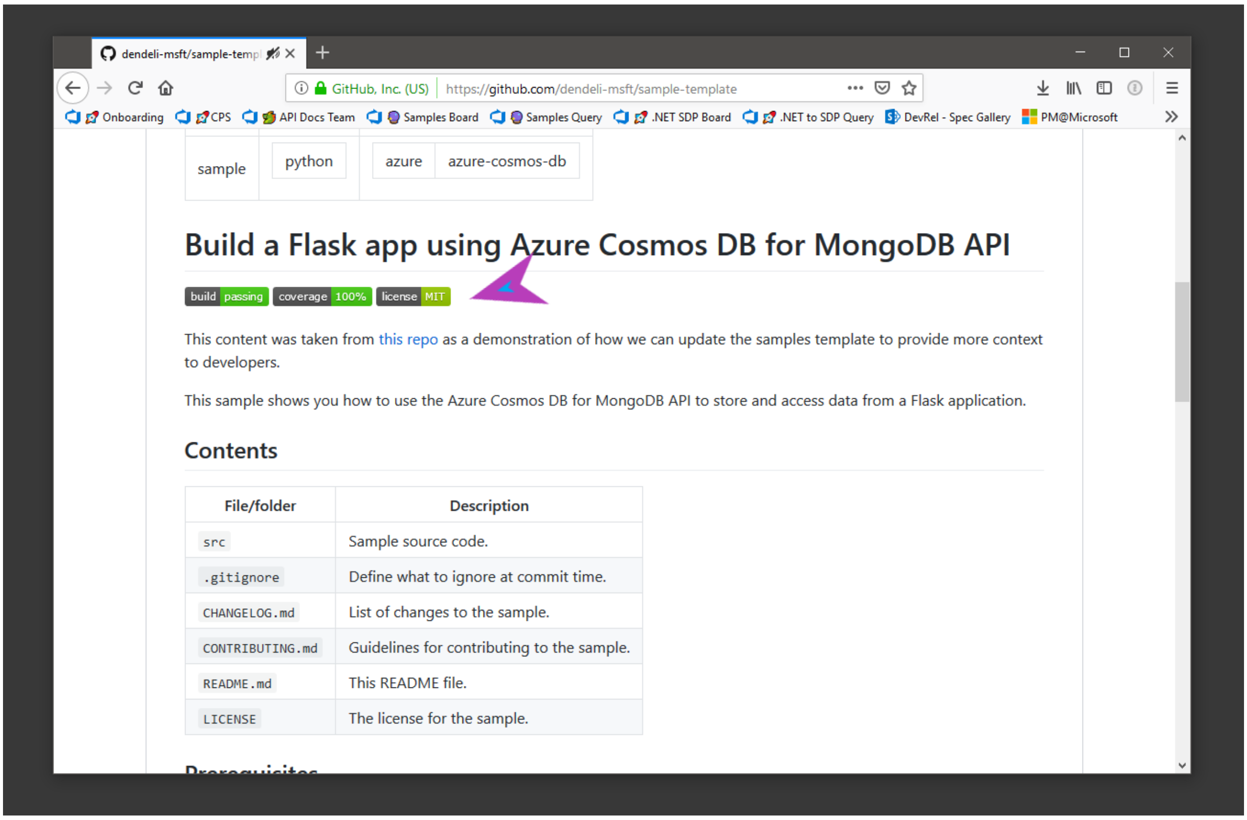
```
# Build a Flask app using Azure Cosmos DB for MongoDB API
```

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.



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.

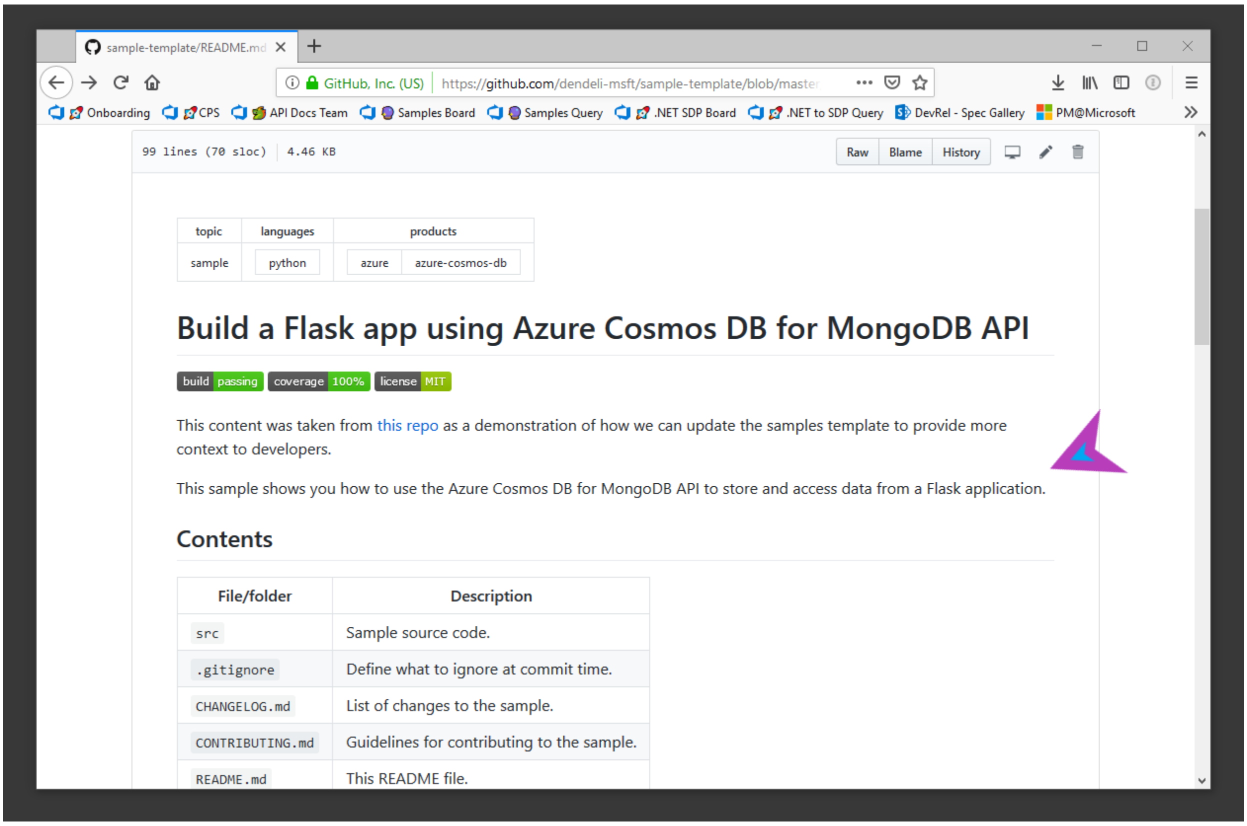
Example of badges

```
![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

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.



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

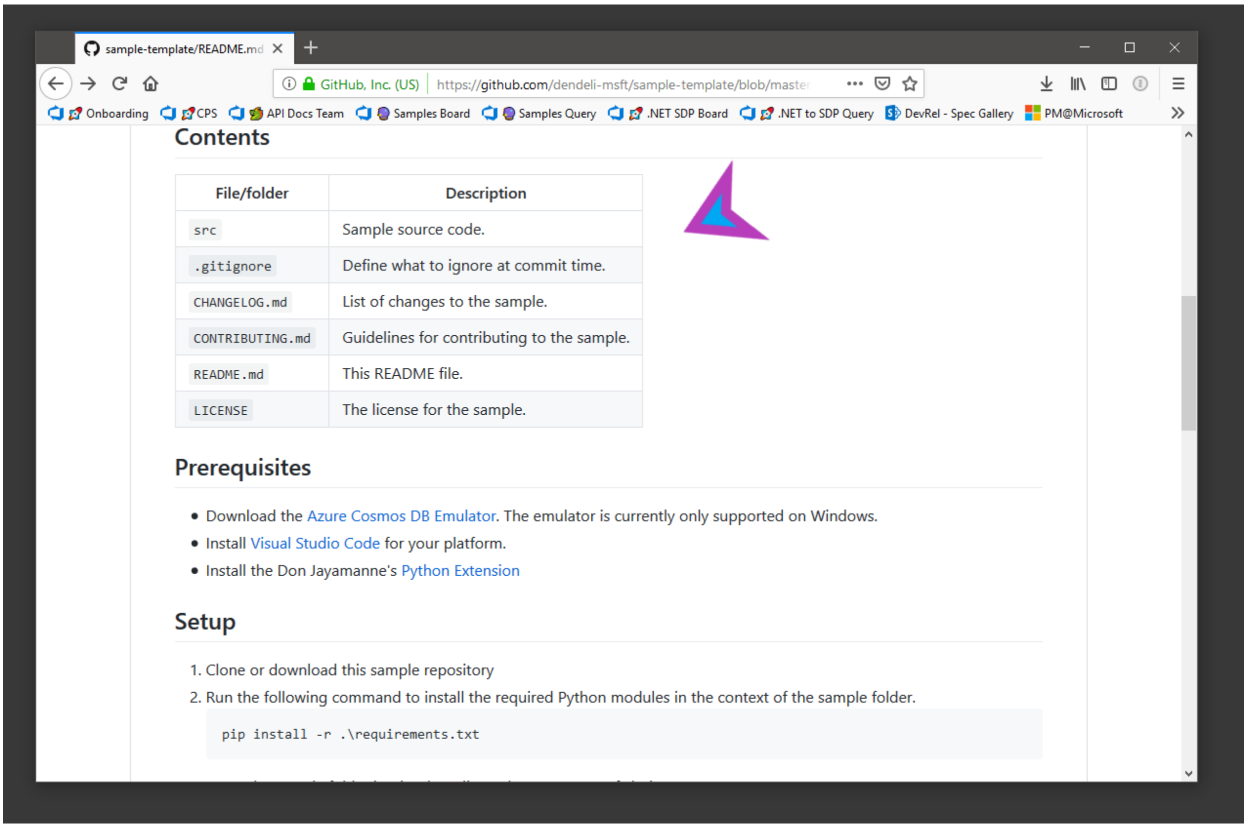
```
This sample shows you how to use the Azure Cosmos DB for MongoDB API to store and access data from a Flask application.
```

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.



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

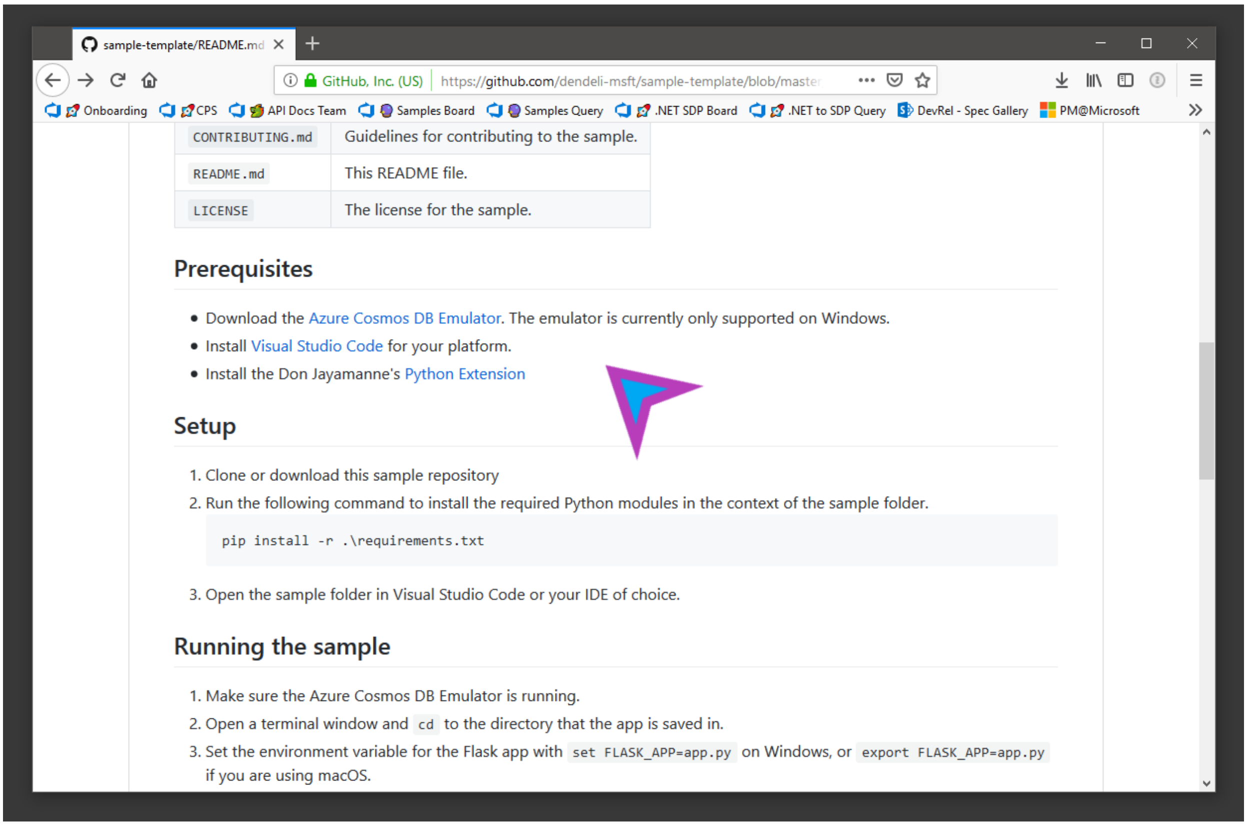
```
| File/folder | Description |
|-------------|-------------|
| `src`       | Sample source code. |
| `.gitignore` | Define what to ignore at commit time. |
| `CHANGELOG.md` | List of changes to the sample. |
| `CONTRIBUTING.md` | Guidelines for contributing to the sample. |
| `README.md` | This README 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`.



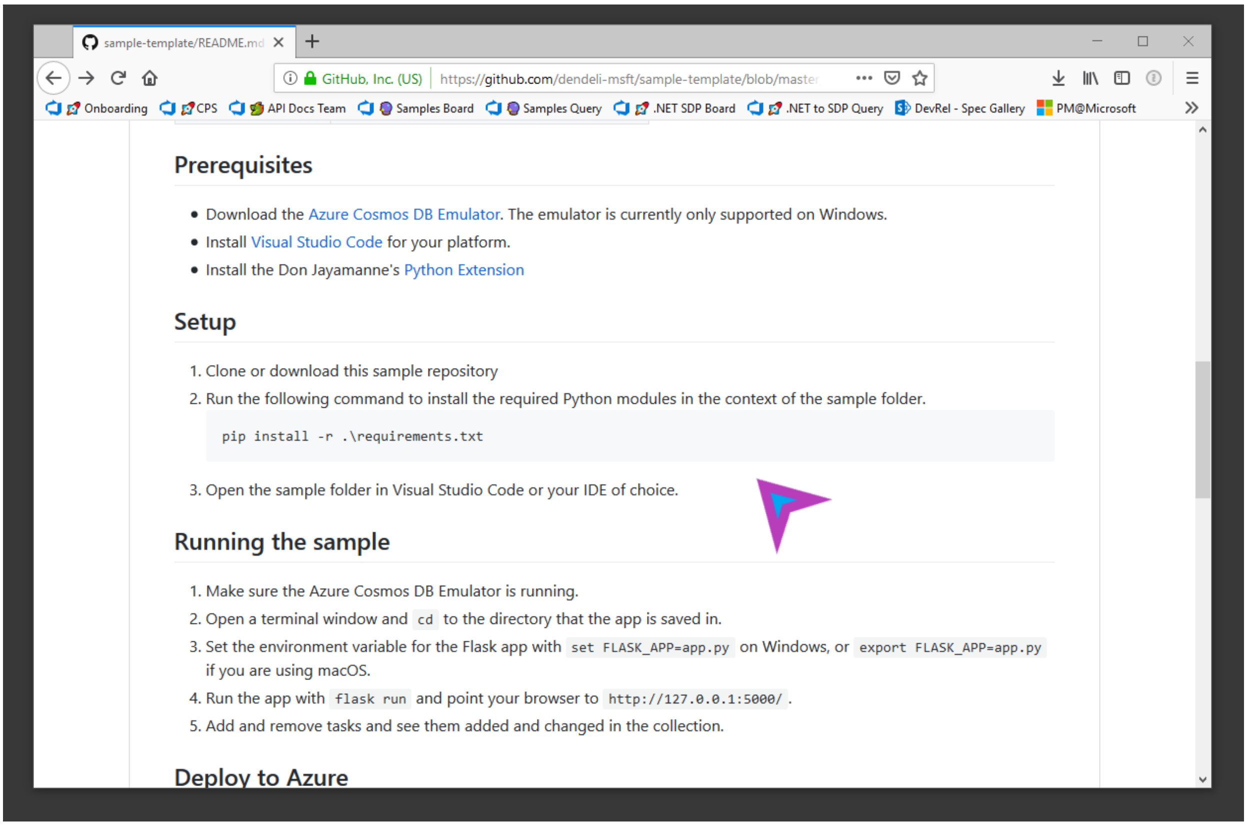
Example of prerequisites

```
- 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**.



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.

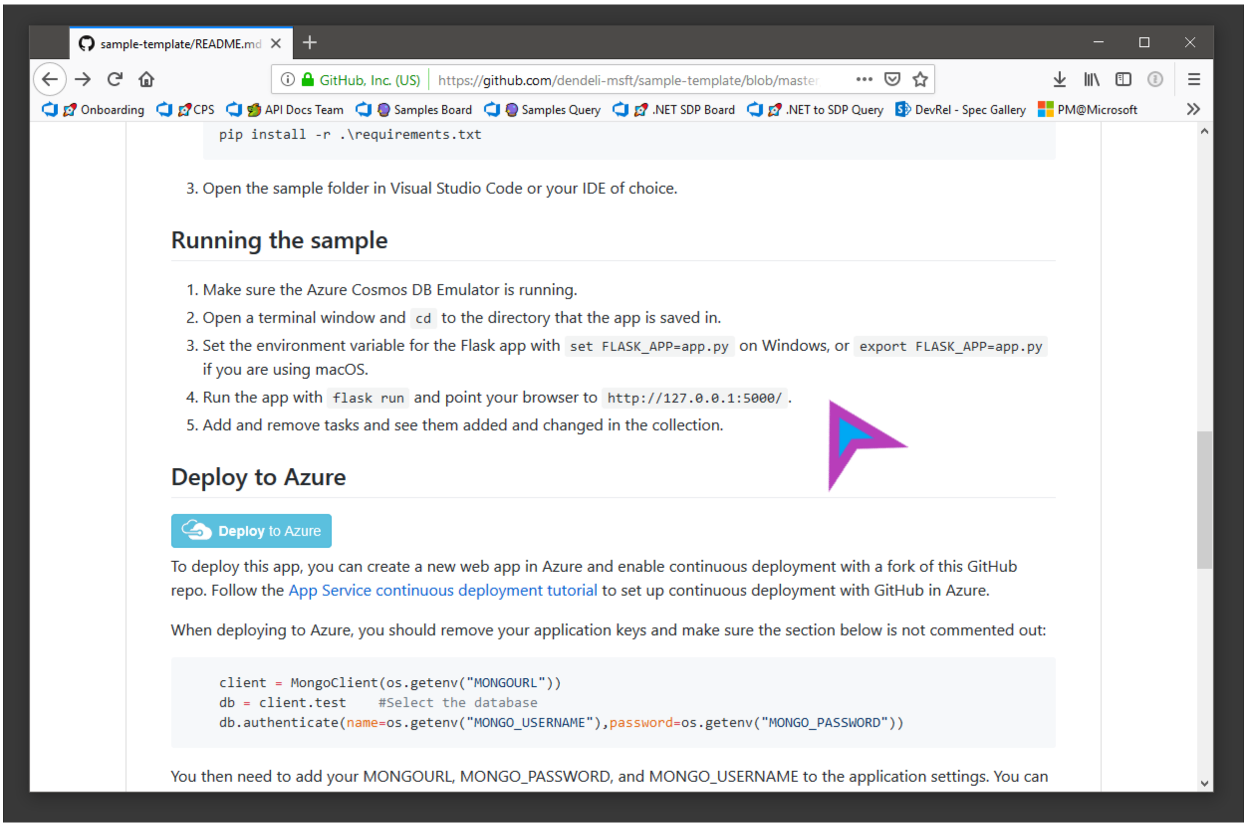
Example

```
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`.



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.
```

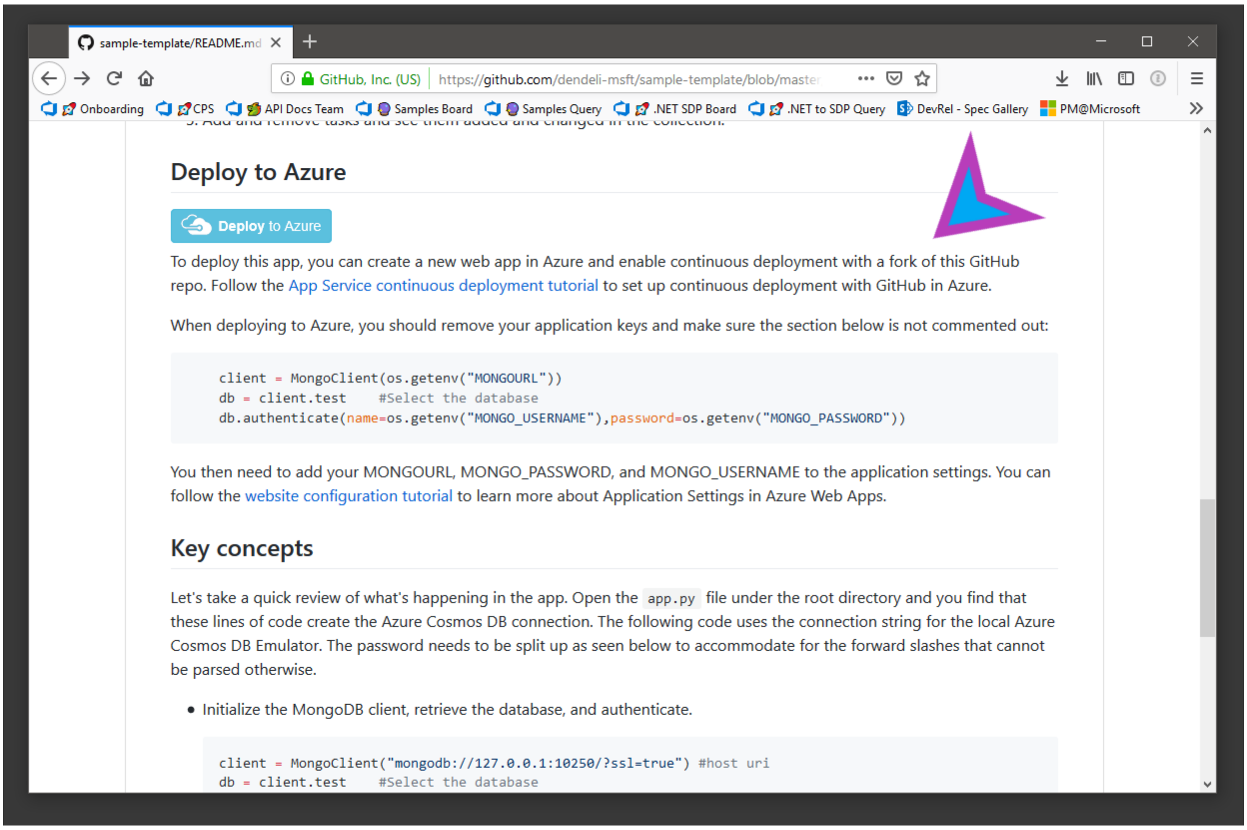
Deploying to Azure

> **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">
<img src="http://azuredeploy.net/deploybutton.png"/>
</a>
```

The section heading should be **Deploy to Azure**.



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.

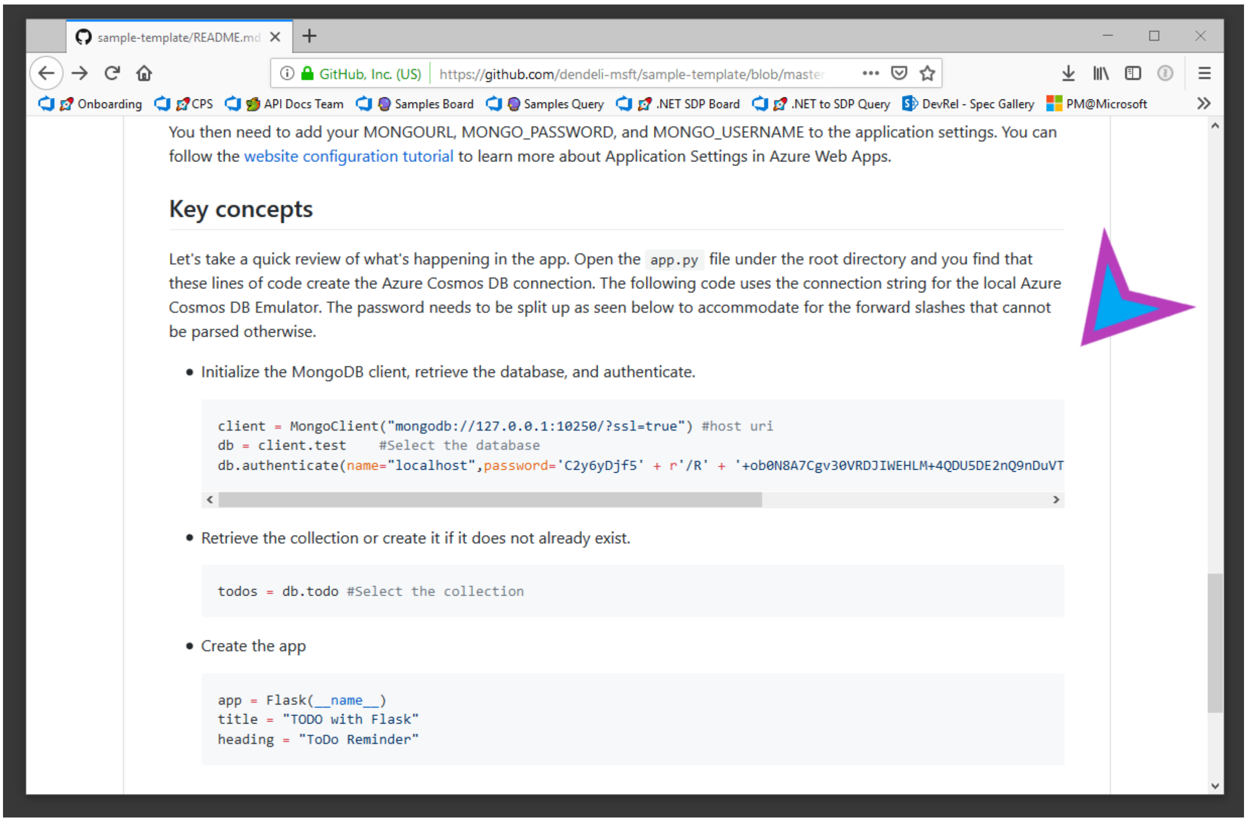
If you have an Azure Resource Manager template available, make sure to also set the `azureDeploy` metadata. This will allow us to enable the **Deploy to Azure** capability for the sample in the samples browser.

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**.



> **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.

Onboarding Samples to docs.microsoft.com/samples

6/24/2021 • 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 `README.md` equals one page on `docs.microsoft.com/samples`	You can have several `README.md` files in the repository (*for example, one for each sample that you document*) - each will become its own permalink page on the samples portal *if* indexing is successful for said sample.
Metadata-driven indexing	Only samples that have a `README.md` with a YAML front-matter containing `page_type: sample` will be indexed into the system. Any other `README.md` 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
`page_type`	Required	Always must be `sample`. This is used to identify `README.md` files that need to be ingested.
`languages`	Required	The array of programming languages covered by the sample. The identifiers need to map to the list of approved programming language values. See list of language slugs. If the metadata value you need is not on the list, refer to the section on requesting new values.
`products`	Required	The array of technologies covered by the sample. The identifiers need to map to the list of approved product values. See list of product slugs. If the metadata value you need is not on the list, refer to the section on requesting new values.

FIELD	REQUIRED?	DESCRIPTION
`name`	Optional	The name of the sample, as displayed in the list on the samples browser landing page. 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. **Default value:** The first H1 line (`#`). If first H1 line is absent, and the `name` attribute is missing, indexing of the code sample will fail.
`urlFragment`	Optional	The URL fragment of the published sample. The full URL will always be `https://docs.microsoft.com/{locale}/samples/{re organization}/{repo-id}/{urlFragment}` . While this metadata node is not required, our recommendation is to *always* have this value set - this will ensure that the page link is immutable, regardless of sample `README.md` changes. See below for fragment restrictions. **Default value:** First H1 line (`#`) transformed in a URL-compatible ID. If the first H1 line is absent and the `urlFragment` attribute is missing, ingestion will fail.
`description`	Optional	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. **Default value:** The first 150 characters following the first H1 (`#`) line. If no content is available, no description will be added.
`statusNotificationTargets`	Optional	[**CURRENTLY UNUSED**] Array of email aliases to which status updates on the sample build are sent.
`azureDeploy`	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
`extendedZipContent`	Deprecated	A list of additional files or folders to include in the ZIP. Each element in `extendedZipContent` has two properties: `path` (name in the repo), and `target` (name in the ZIP). Both are absolute paths. Use `/` as a directory separator. If your sample uses `extendedZipContent`, then the Samples service will generate a new public release of your sample on your GitHub project every time the sample is updated.
`extensions`	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.

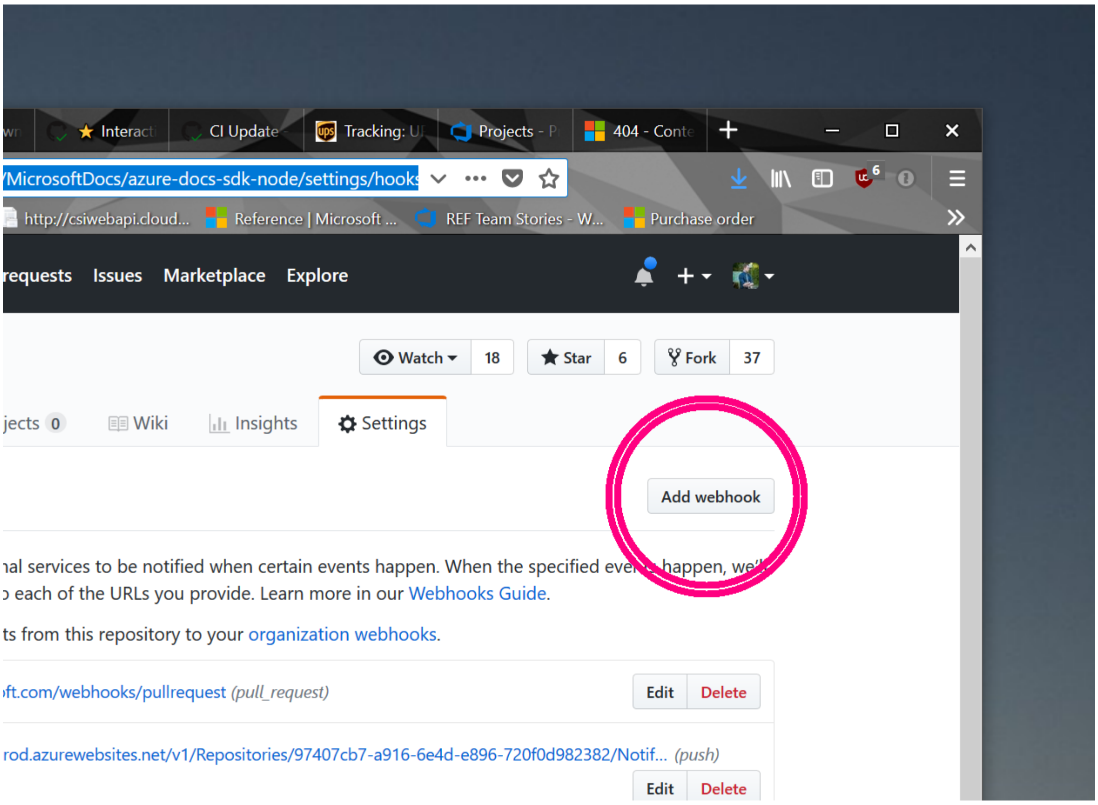
Create webhook in sample repo

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 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.

Diagnostics for code samples

6/24/2021 • 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.

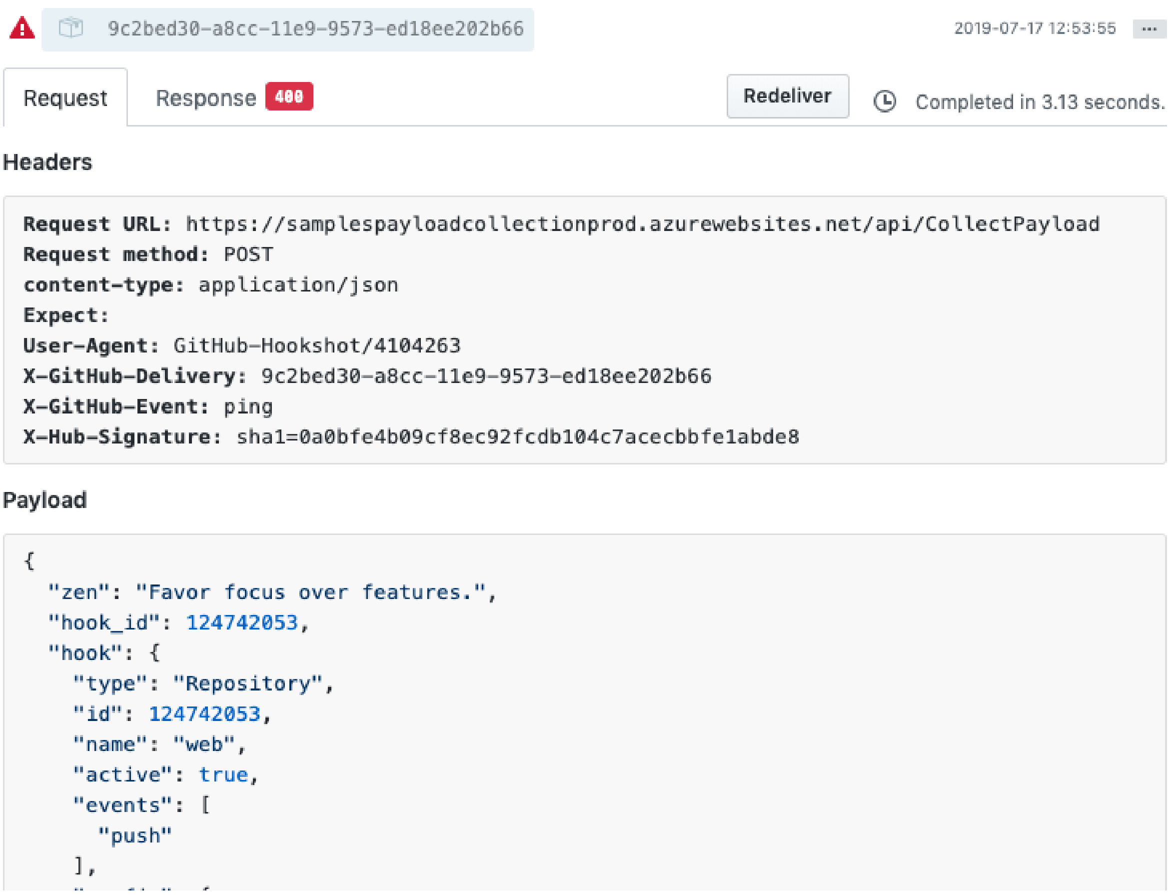
> **IMPORTANT**
>
> Only samples from the following organizations are supported: `Microsoft`, `MicrosoftDocs`, `dotnet`, `Xamarin`, `OfficeDev`, `MicrosoftGraph`, `Azure`, `Azure-Samples`, `aspnet`, `mspnp`, `iot-for-all`, `azure-rtos` and `MicrosoftEdge`. If your organization is not on the list, contact us.

https://msit.microsoftstream.com/embed/video/d525c670-55b0-4330-8108-a9fdaebbe9a8?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**:



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

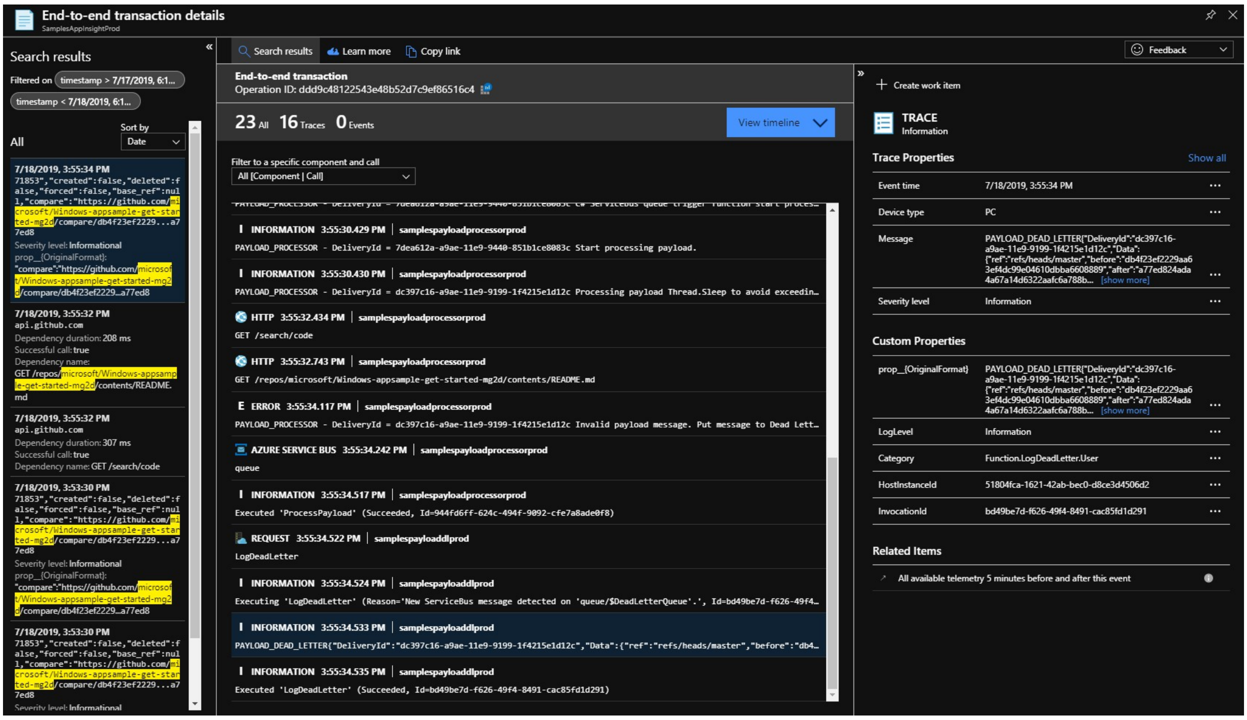
If you are part of the Microsoft Samples core team and do not have access to the resource above, contact Banani Rath.

Make sure that you are in the **Search** view to see all events.

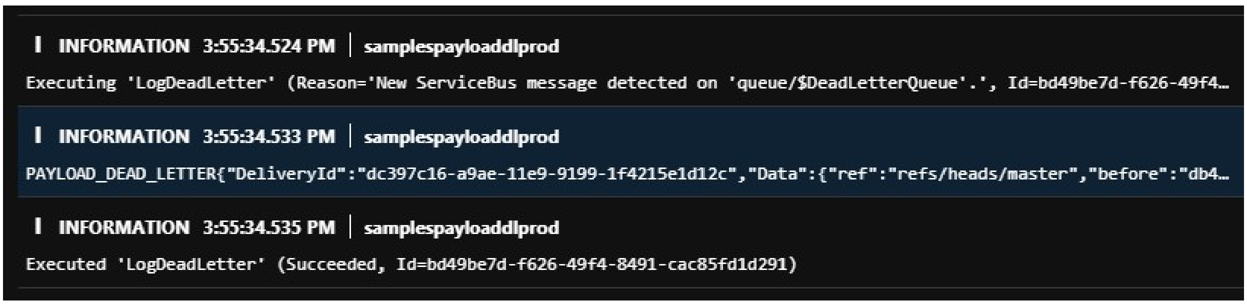
When it comes to figuring out what went wrong, this should be the place to start. The easiest way to get the chain of events is to search for the repository. For example, if you want to understand why the `https://github.com/Microsoft/windows-appsample-get-started-mg2d` repository fails to onboard, they 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.

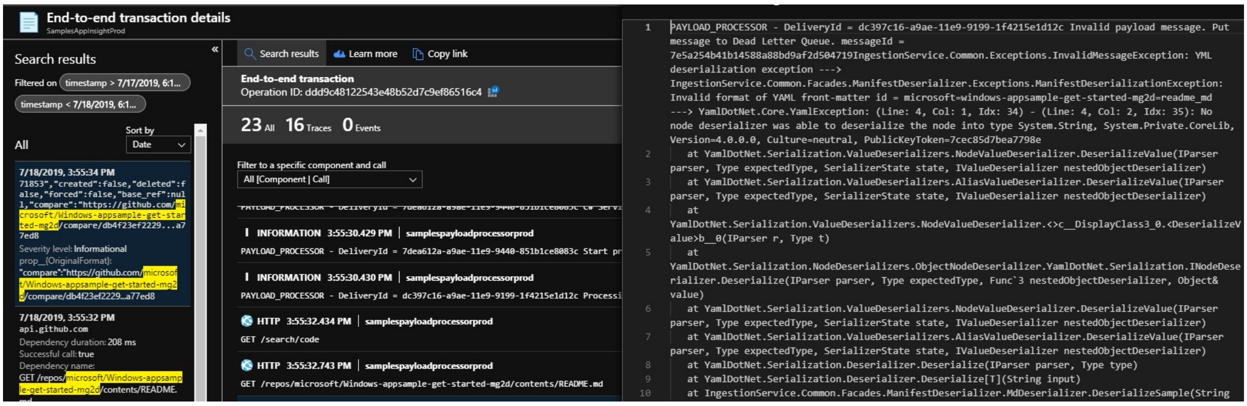


If you see an entry with `PAYLOAD_DEAD_LETTER`, that's your first clue that the onboarding failed, because we "dead-letter"-ed the change.

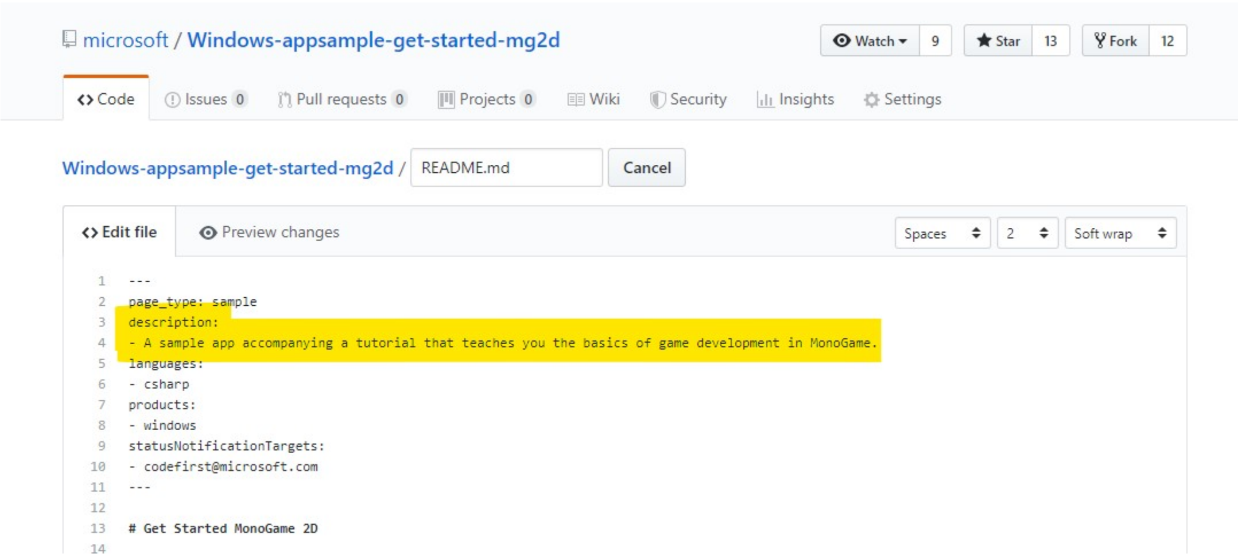


Once you see this, you need to scroll through the list of events, until you find one of type **Error** - that contains the stack trace of what went bad.

Expanding the item and looking at the exception, we can infer the problem - the Application Insights log entry is explicit about the issue:



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:



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

7/7/2021 • 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.	Overview
To give new customers an example of some functionality.	Quickstart
To help the customer build a skill by showing them the best way to do something.	Tutorial
To help the customer complete a specific task.	How-to guide
To help the customer understand, rather than do, something.	Concept
To help the customer resolve a specific error or problem that customers commonly run into and would search on.	Troubleshoot

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]
```

```
[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). -->
- <!-- 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
```

````
## 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:

    ```azurepowershell-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%2Fa
zure-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 (`azurecli-interactive`) or Azure PowerShell (`azurepowershell-
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 (`azurecli-interactive`) or Azure PowerShell (`azurepowershell-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:

> [!div class="nextstepaction"]
> [Tutorial: Create and deploy your first ARM template](/azure/azure-resource-manager/templates/template-
tutorial-create-first-template)
````

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.
-->
```

```markdown
[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

6/24/2021 • 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

7/9/2021 • 5 minutes to read

This article provides a Markdown template to use when writing an Azure tutorial article.

All articles need the metadata header along with the required values for reporting. For detailed instructions for all tutorial articles, see Template for tutorial articles.

You can copy and paste the following Markdown for your Azure tutorial article:

```
---
title: Page title has the greatest effect 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:
Include the outline of the tutorial 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 we intend tutorials to help new customers use the product or
service to finish 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, the article requires this H2 and
link.--->

## 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
finish 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 finish
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 the reader needs to do to finish 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 the reader needs to do to finish 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 the reader needs to do to finish the
```

````
include a sentence or two to explain only what the reader needs to do to finish 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:

    ```azurepowershell-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, enter...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).

## 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/costs/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/paths/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 you 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-resource).
```

```
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. -->

> [!IMPORTANT]
> 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 of 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/virtualMachinesscaleset](/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/azuremetrics).
```

```
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/azuremetrics) | <!-- 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

6/24/2021 • 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.config.json 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
```

```
4.  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:
    * `#`:  `#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 `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 hte 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 an advanced tutorial

6/24/2021 • 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
```

```
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.
Mention that this tutorial is relatively long and complex.
--->

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
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:
    * `#`:  `#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 `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 hte 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

6/24/2021 • 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
````
````

`````
    ```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'e 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.
> [!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

6/28/2021 • 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 decribing 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?

7/7/2021 • 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.
- **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. However, some product groups do publish more granular policies or terms to 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. 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 follow 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

7/19/2021 • 6 minutes to read

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 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 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), 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 Policheck 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.

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

7/7/2021 • 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) 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

7/7/2021 • 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 Toftemark.
- **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.

See also

- Quick tips for creating a blog post

Where to host downloadable files

7/7/2021 • 2 minutes to read

Sometimes, teams need to post content in the form of downloadable file formats like:

- A Word document
- A compressed file
- A PDF
- and so on

Repos on GitHub aren't intended as hosting locations for these types of files. Limit content in the MicrosoftDocs repos to technical documents authored as markdown or YML files and image files that you've used in those documents.

The Microsoft Download Center (DLC) is the correct channel for posting downloadable content.

For more information, see the DLC documentation: Download Management Studio.

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

7/21/2021 • 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?

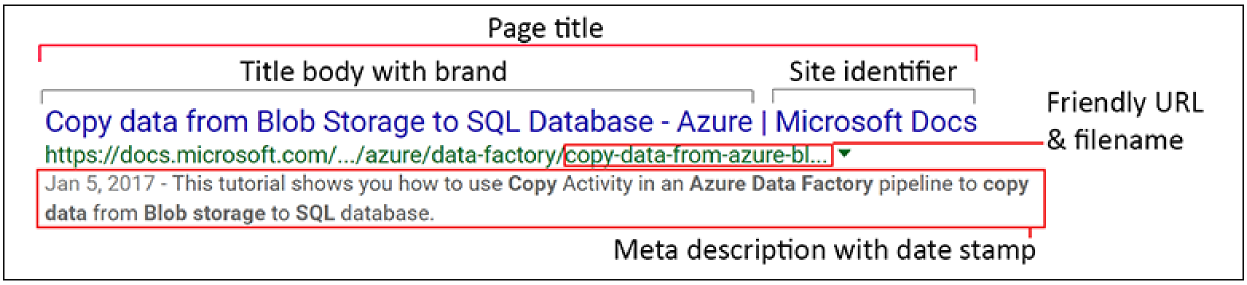
Search is how most customers find information on the web. Some data points to consider:

- 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.
```

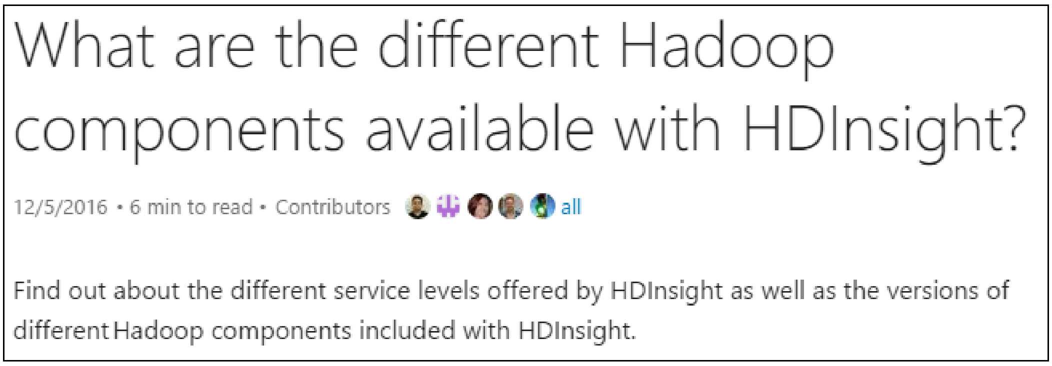
TAG	GUIDANCE	NOTES
title (length)	Character counts are estimates: Check a preview. Title body is about 60-65 chars max including spaces and brand.	Unique text string with important terms first. The article intent must be clear in search results.

TAG	GUIDANCE	NOTES
title (branding)	Include the product or technology, such as Azure or ASP.NET.	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	**Don't use this tag**	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.



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.

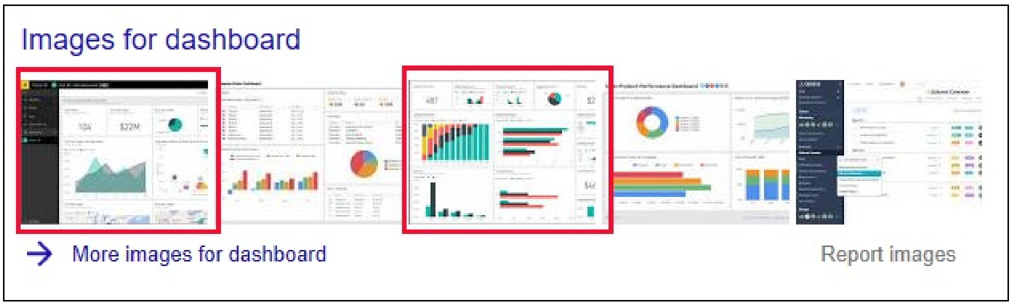


Image results on page 1 for "dashboard" from Microsoft Docs content.

> **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?

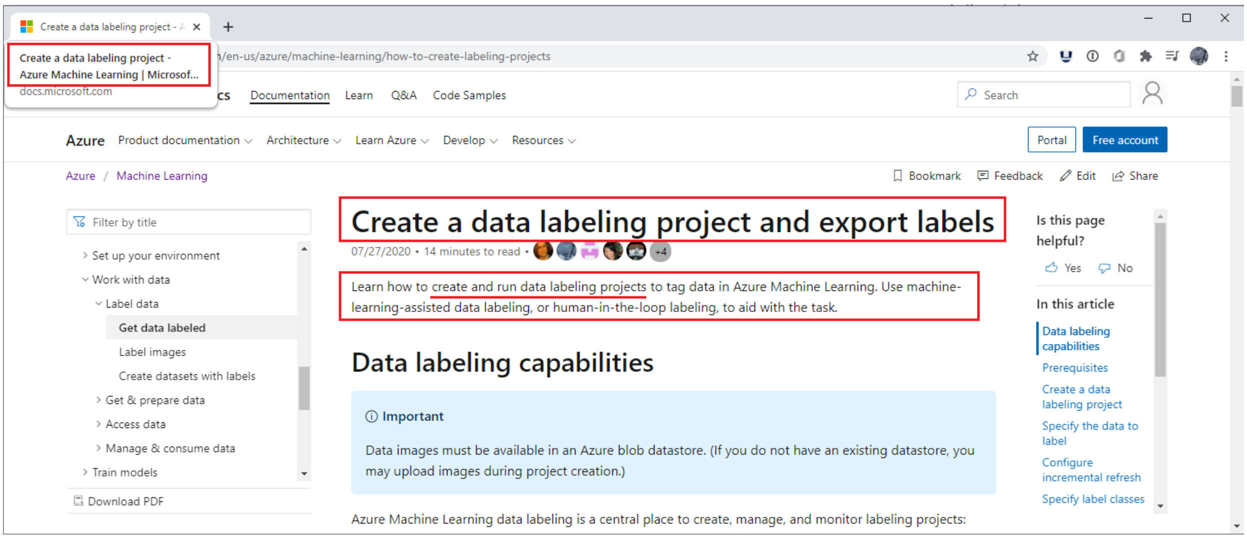
Find a succinct, customer-friendly way to state the intent, and then repeat it in the:

- 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:



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

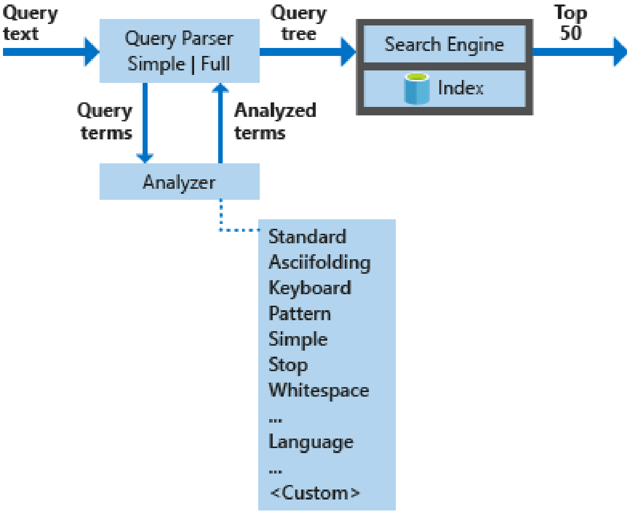
AWS service	Azure service	Description
Kinesis Analytics	Stream Analytics Azure Data Explorer Data Lake Analytics Data Lake Store	Storage and analysis platforms that create insights from large quantities of data, or data that originates from many sources.
QuickSight	Power BI	Business intelligence tools that build visualizations, perform ad hoc analysis, and develop business insights from data.

Do diagrams use text sparingly?

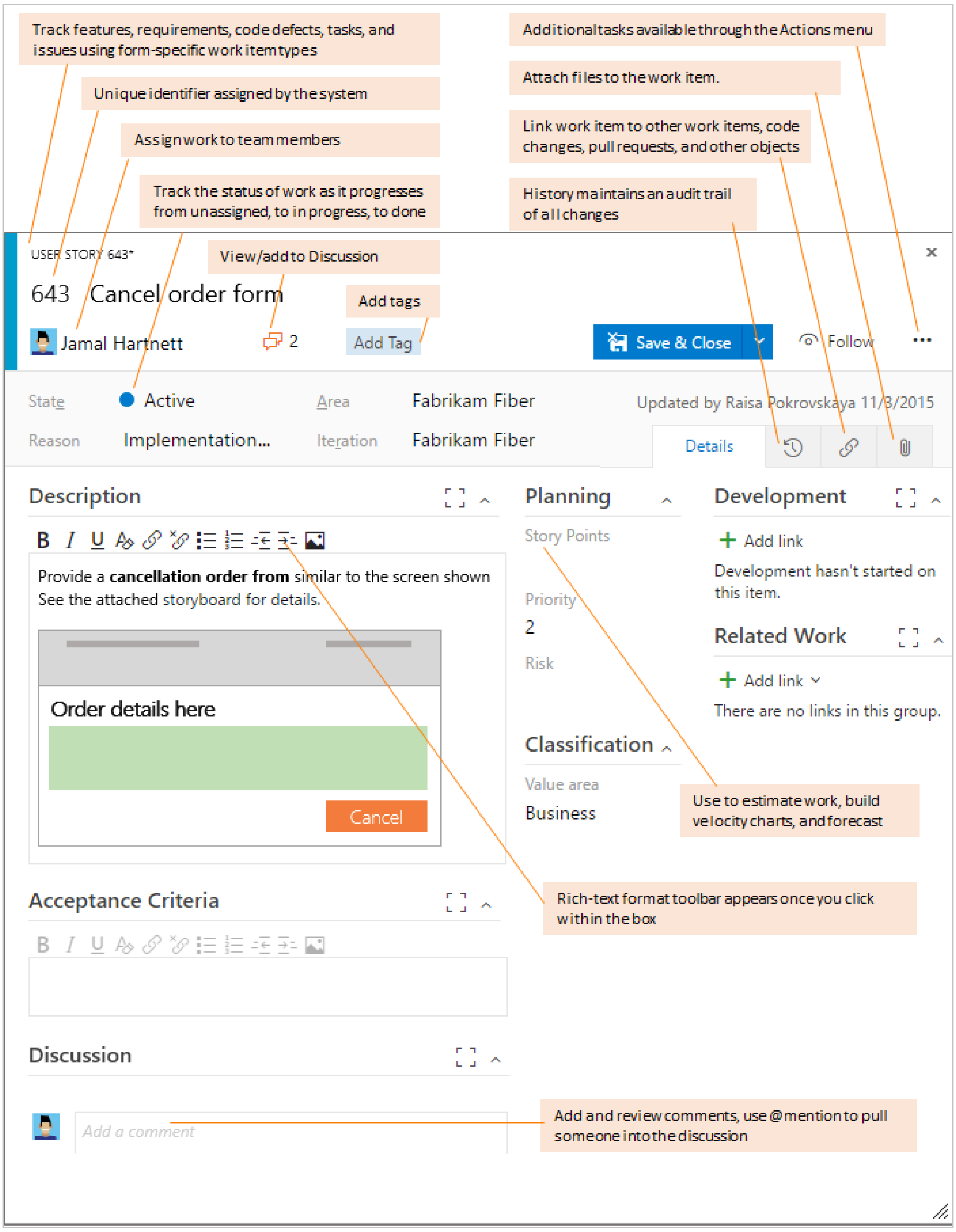
Avoid text-heavy diagrams. Text in diagrams isn't accessible, and Google can't crawl it.

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

7/9/2021 • 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 search engine result page (SERP) 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 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 preview 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. It also shows 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.

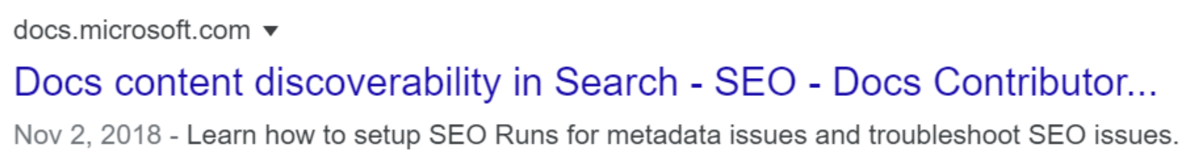
To use Search Results Preview:

1. In VS Code with the Docs Authoring Pack, press **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:



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/en-us/sql/relational-databases/logs/troubleshoot-a-full-transaction-log-sql-server-error-9002?view=sql-server-ver15&preserve-view=true

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, article-level metadata overwrites folder level and global metadata. Here are the examples for each type.

Article 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 comput
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 in 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's 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's an example from Xamarin repo:

```
      "globalMetadata": {
        "uhfHeaderId":"MSDocsHeader-Xamarin",
        "breadcrumb_path": "/xamarin/breadcrumb/toc.json",
        "titleSuffix": "Xamarin",
        "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

7/9/2021 • 3 minutes to read

The meta description is an article summary that displays under page titles on the search engine results page (SERP). The description also displays 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 article 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/paths/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 provides more information to users and reduces 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 select the article.

SEO: Tips for writing a heading 1 (H1)

7/9/2021 • 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 affect on search ranking and the greatest affect 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. 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's 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 use bold or italicized text as 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

6/28/2021 • 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'.

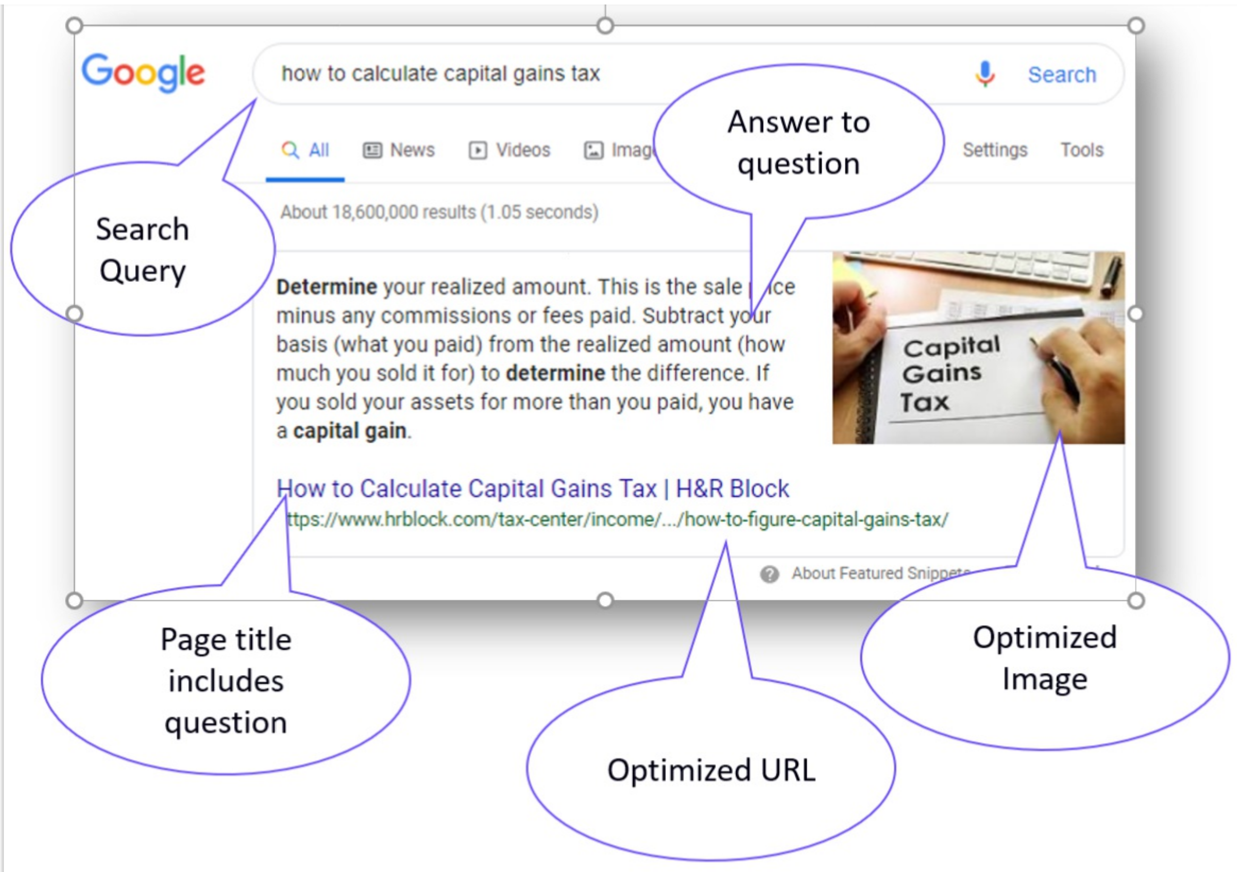
The quick answer box result allows users to get the information they need without clicking on the blue link. Long tail keywords will be an effective tool in helping product or service articles to appear in Google's rich answer box. Because rich answers focus on searched questions rather than keywords, 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.

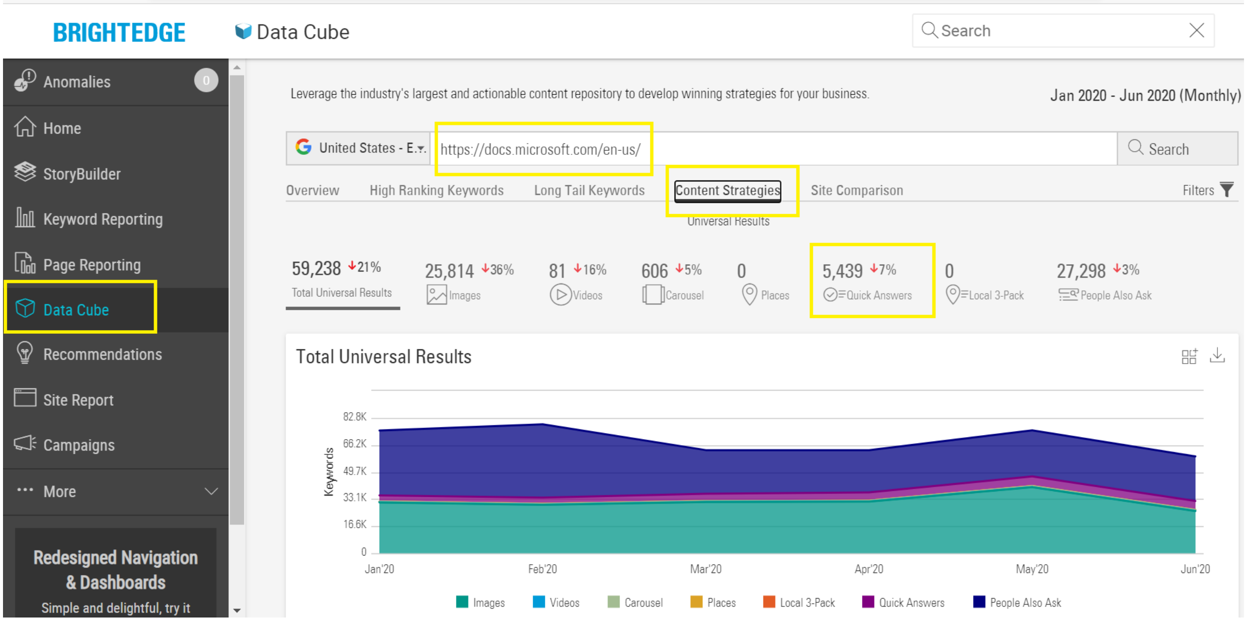
Below image shows how different elements make a quick answer box results in Google search result.



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.

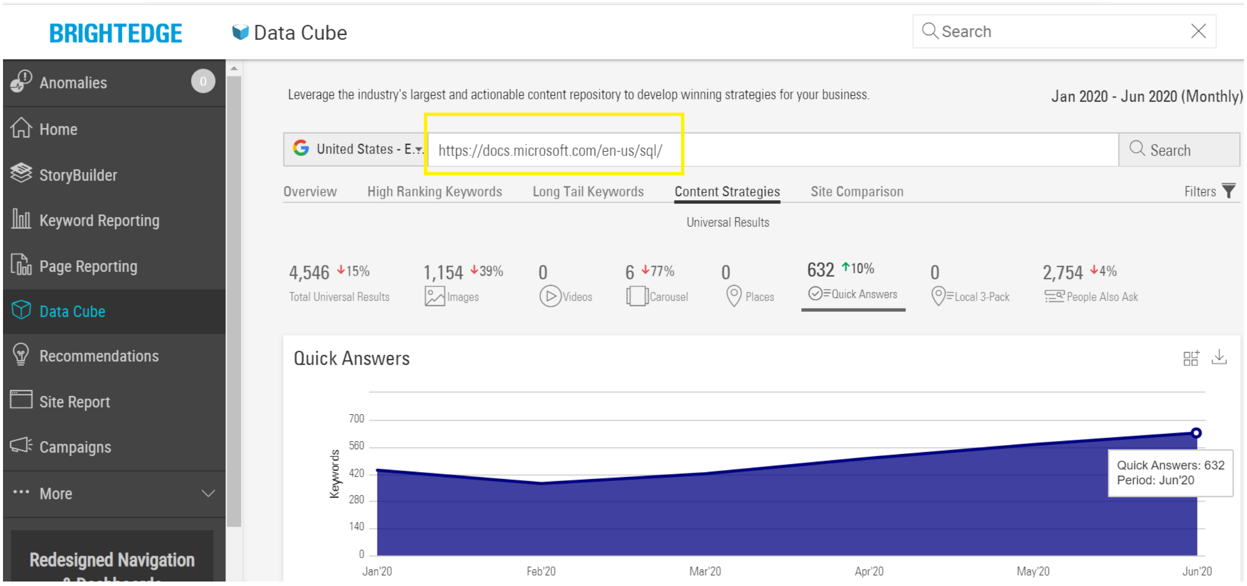


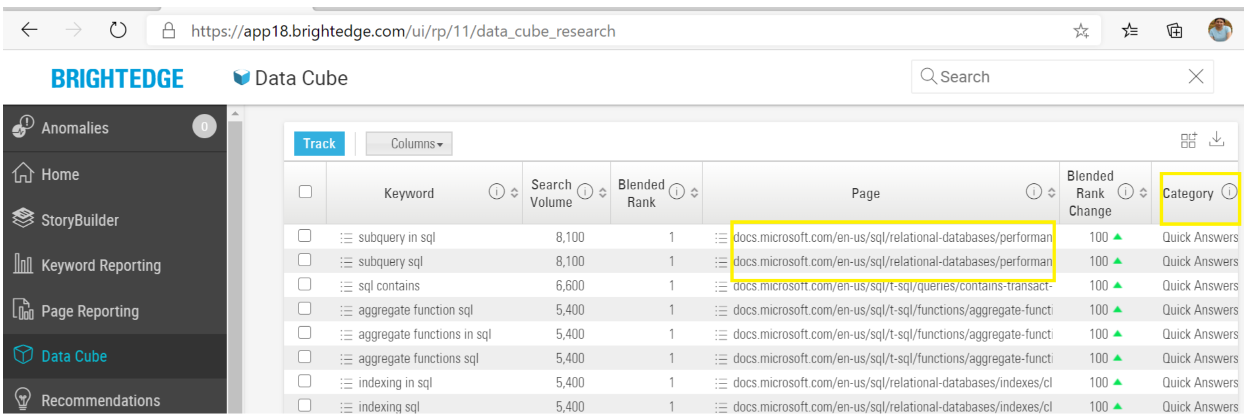
Content Strategies - high-level results

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.





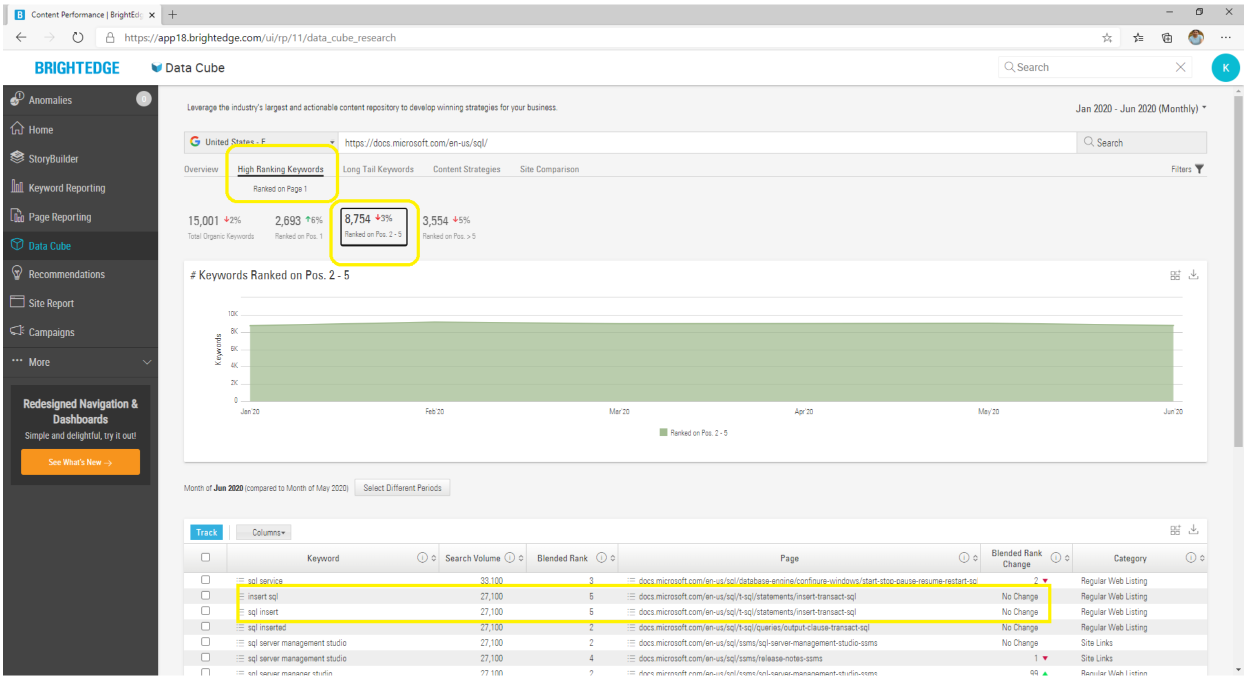
Click on Export icon and then the CSV export option to get a raw file of all of the keywords and URLs ranking for Quick Answer Boxes.

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.



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 about key words and phrases
- Refresh your SEO Basics.

What is a Keyword or Key Phrase?

6/28/2021 • 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.

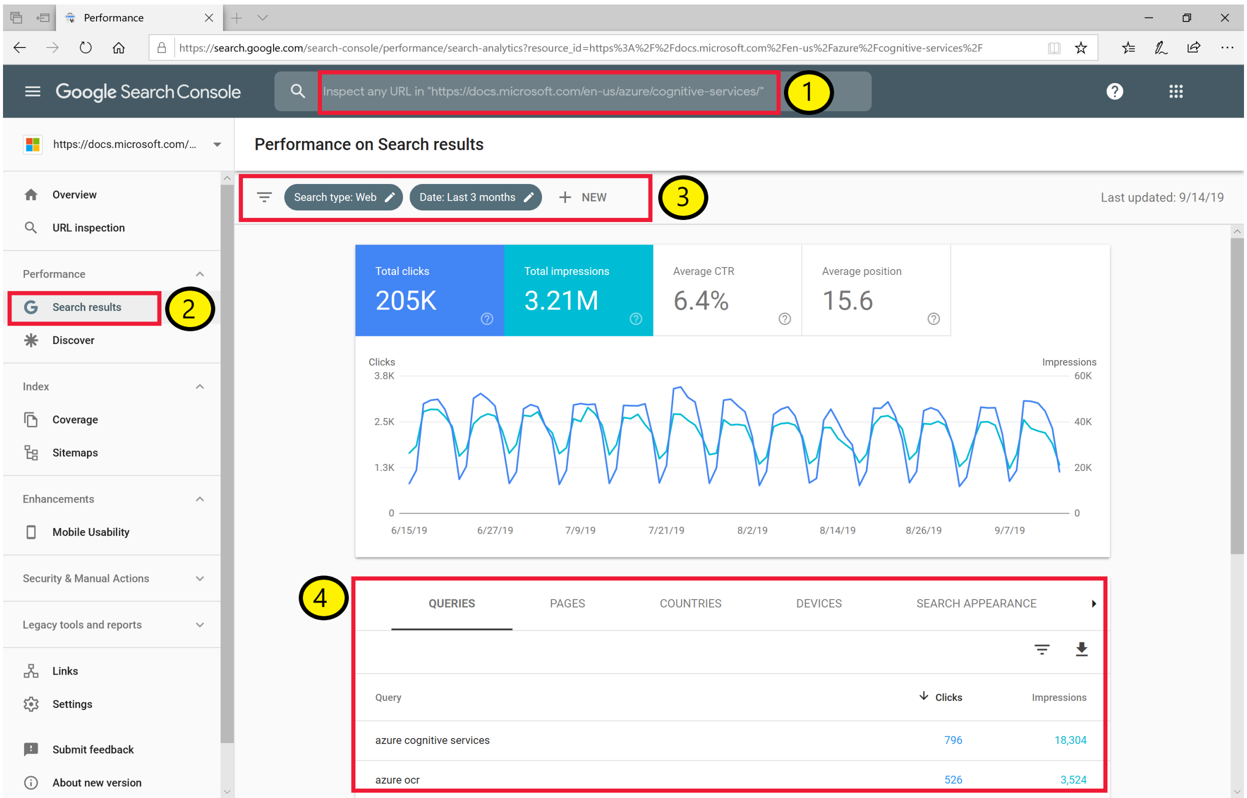
GOOGLE SEARCH CONSOLE	BRIGHTEDGE	SEMRUSH	STAT ANALYTICS
Provides the list of keywords the article ranked in Google	Provides search volume data for keywords	Provides search volume data for keywords	STAT is used to track the keywords ranking and search volume data
Google impressions and clicks for site section and articles	Provides data for **Long tail** and **high ranking** keywords	Provides **keywords variations** and its search volume	STAT provides **related searches** reports from Google
Average position on Google for the site	Provides data for **backlinks and Quick answer box**	Provides **related keywords** 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.



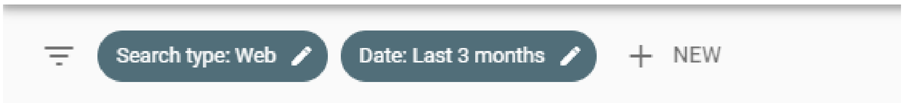
1. The property that is being measured.
2. You want to have Search Results selected.
3. Where you set the metrics and filters for measurement.
4. Where your query results are detailed.

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.

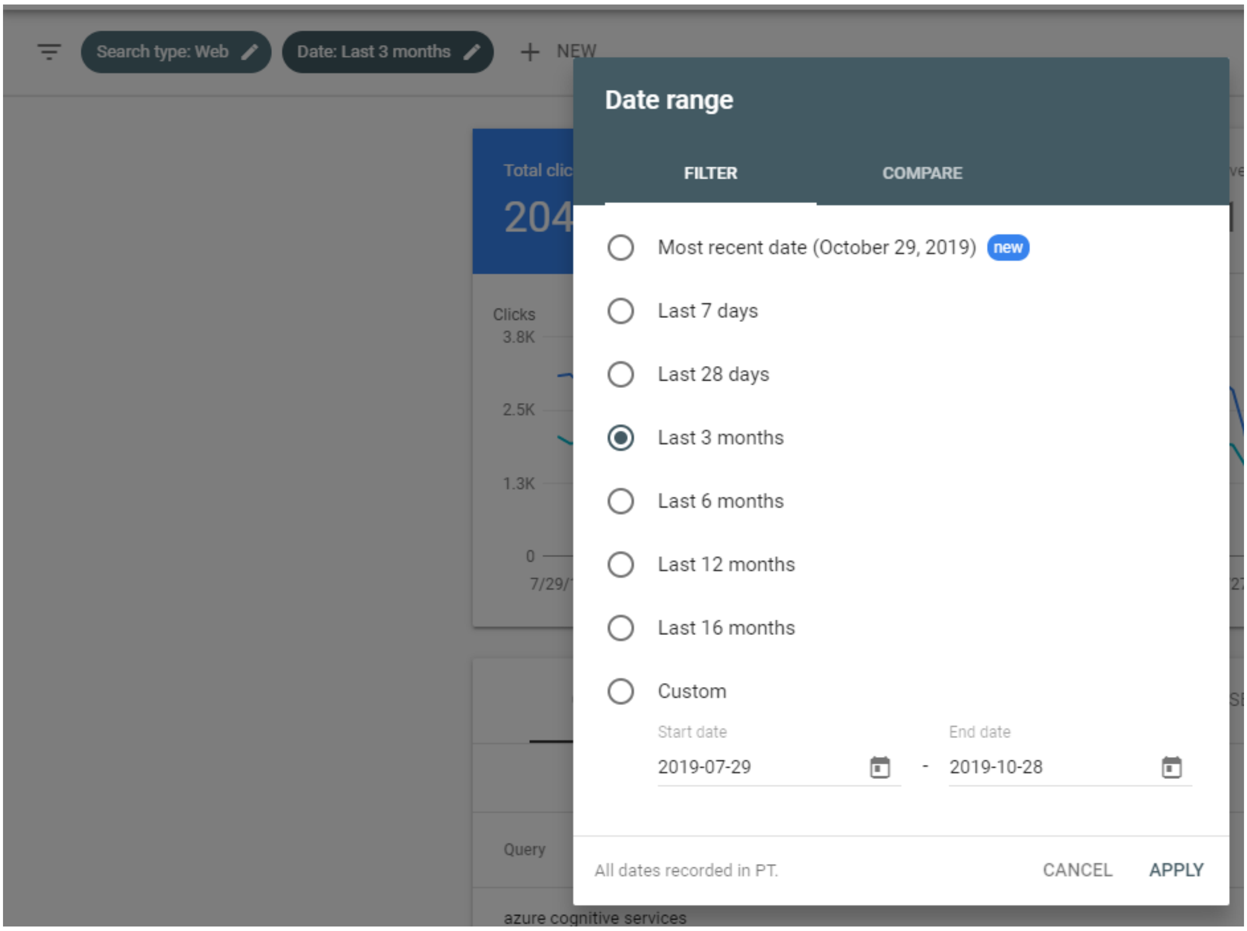
Configuring the Google Search Console report

The default metrics that will be set upon entry will include:

- Search type: Web
- Date: Last three months

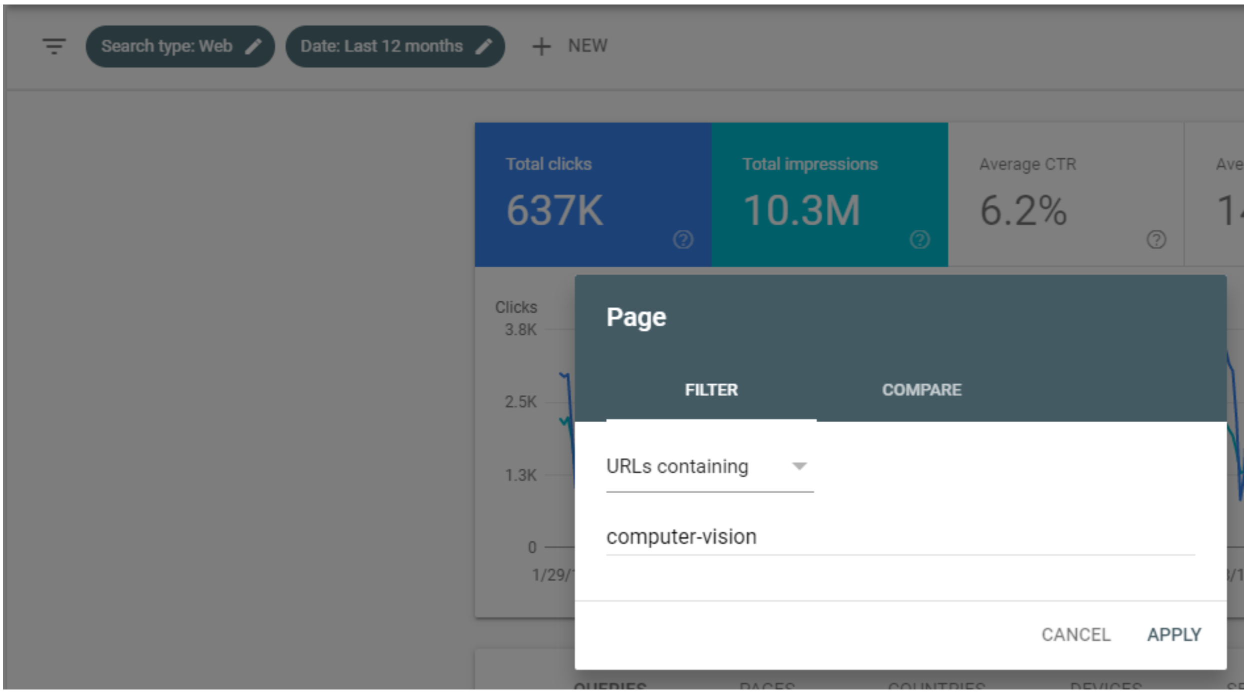


Each metric can be edited by selecting the pencil icon on the metric label.

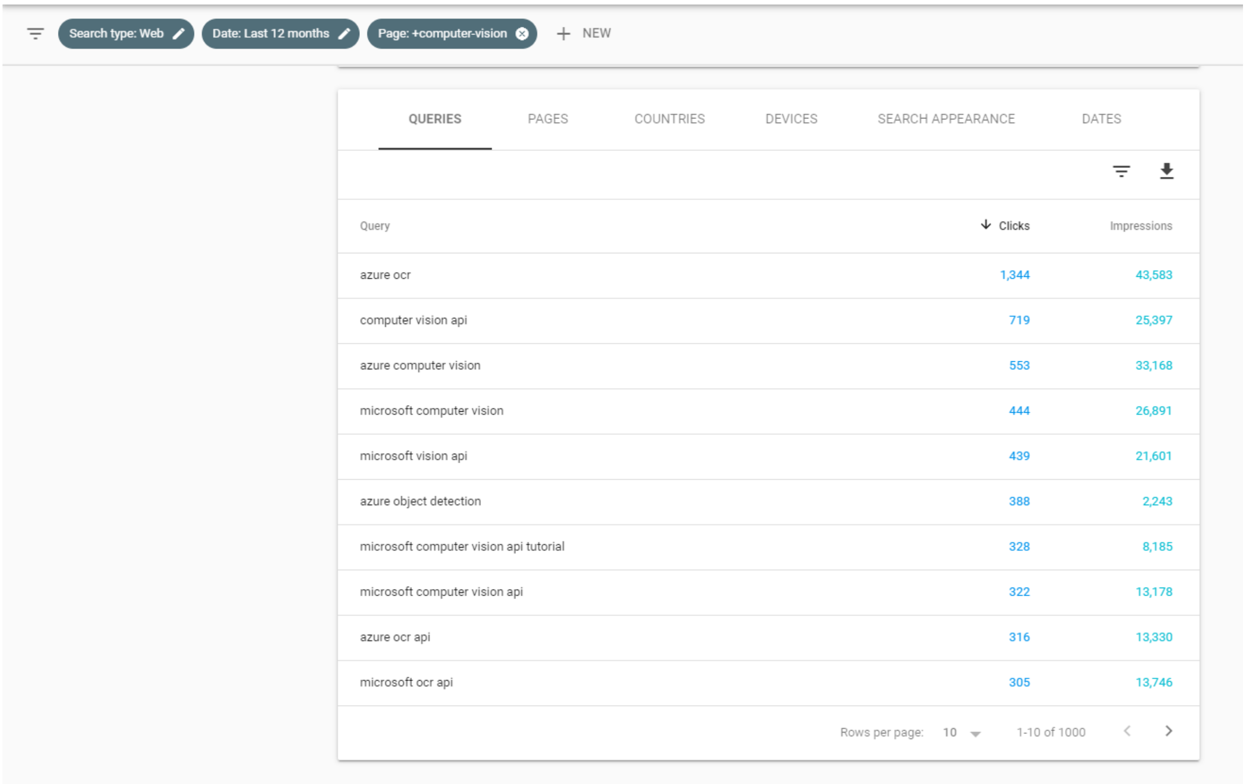


You can select new metrics including query, page, country, device, and search appearance. For the purpose of this article, we will focus on the page and query metrics.

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."

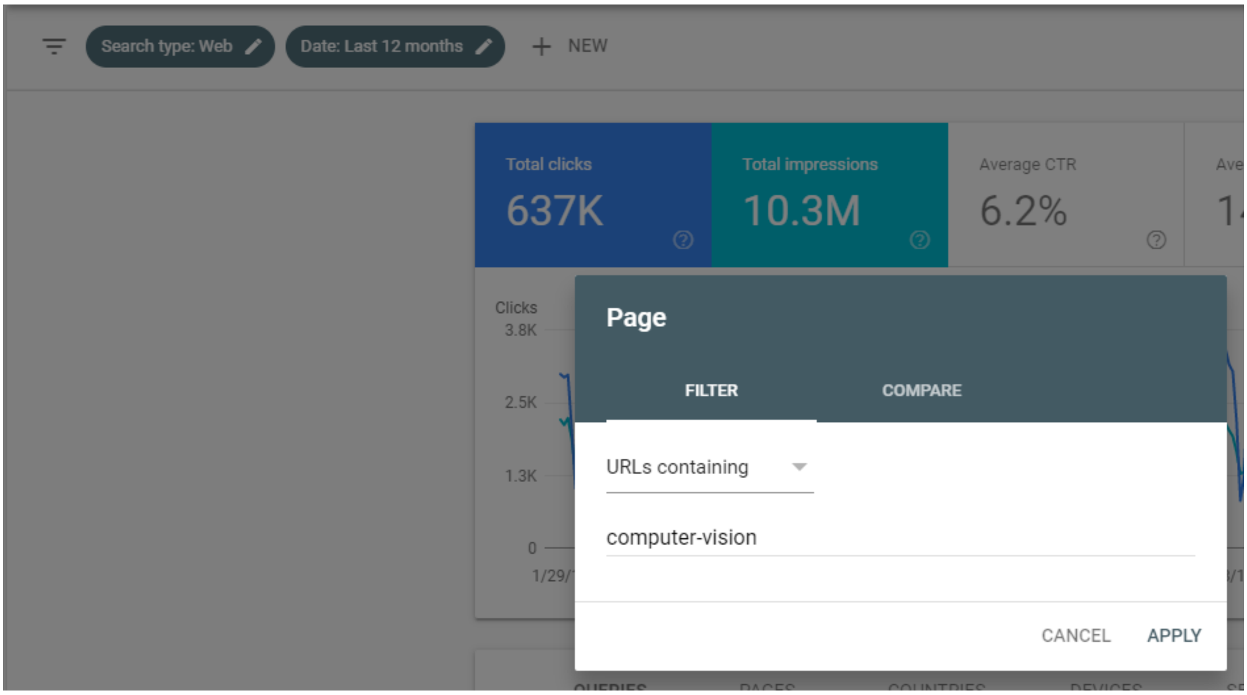


Your results will be displayed. Your top results will most likely include branded terms. You'll want to dig deeper into the results to find non-branded terms with significant traffic but not necessarily high click count.



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."

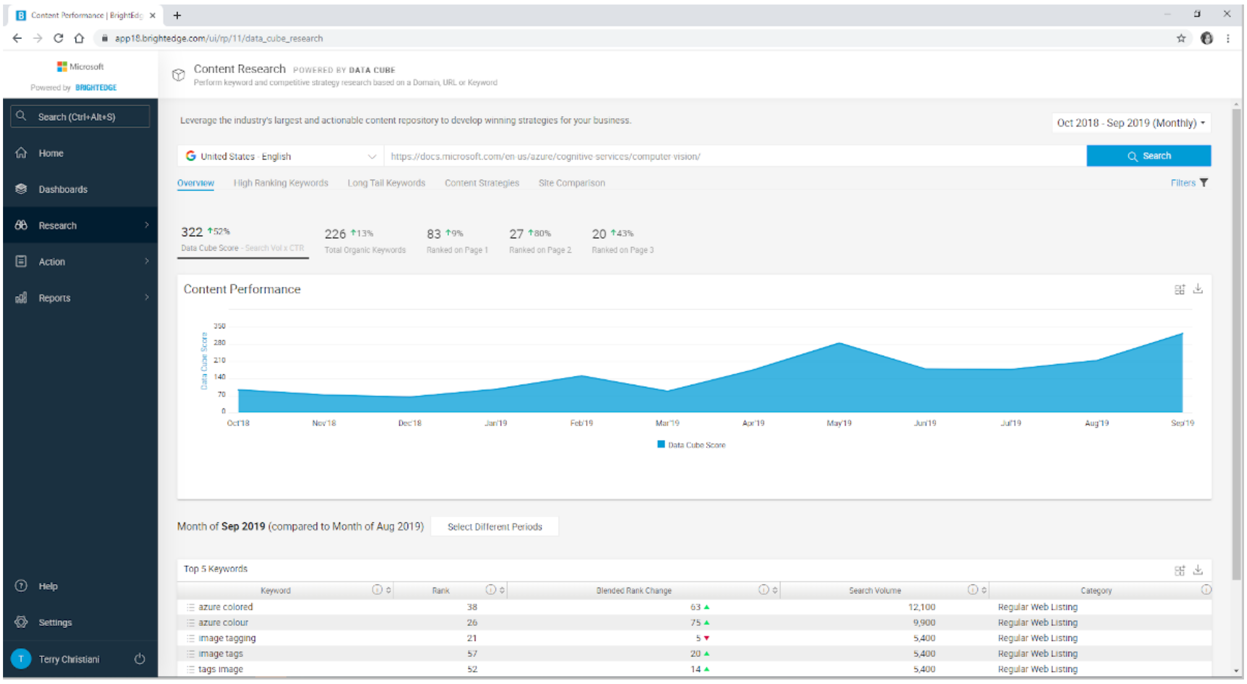


Sort the results by the number of impressions. The phrases with the highest number of impressions but low clicks may be a good starting point for you to explore new key word and phrase ideas.

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.



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

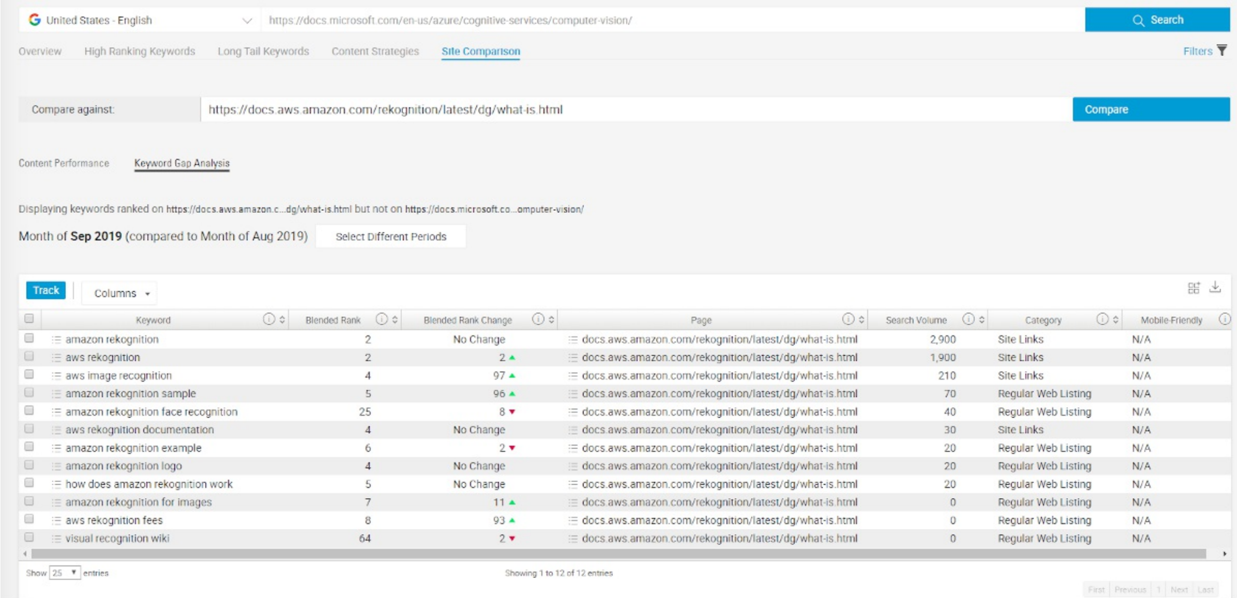
The report will default to an overview of your content's performance for the last 12 months. It will also provide you with the top five keywords that your URL is ranking for in this period.

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.



Next Steps

- Track keywords in STAT analytics tool Docs search keywords research and tracking and measure the ranking performance.
- Refresh your SEO Basics.
- Gather your colleagues together for a SEO Hackadoc.

Docs search keywords research and tracking

6/28/2021 • 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.

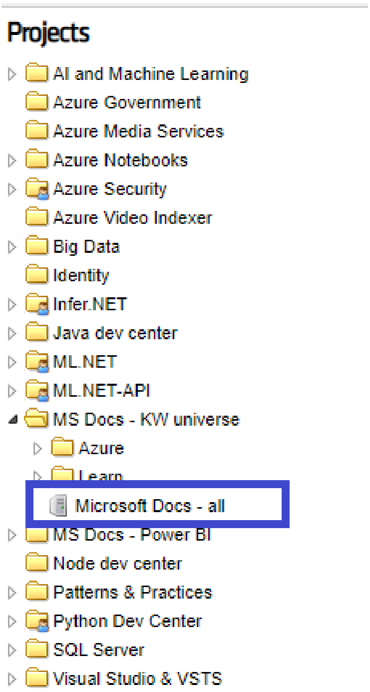
Since the keyword tracking limit in STAT analytics tool is about ~10,000 keywords for Microsoft Docs, it's recommended to track about 200-400 keywords for main site section (Azure, Windows, SQ, Visual Studio). Depending on business priority, the size of each site section, or upcoming content initiatives, 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

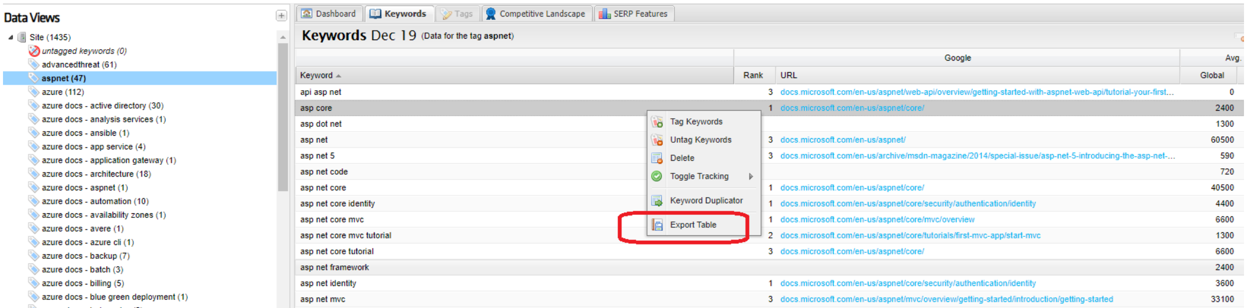
To gain access to STAT analytics send your email to khairunj@microsoft.com and she will grant the access to all the tools to start the keyword research.

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.



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.



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.

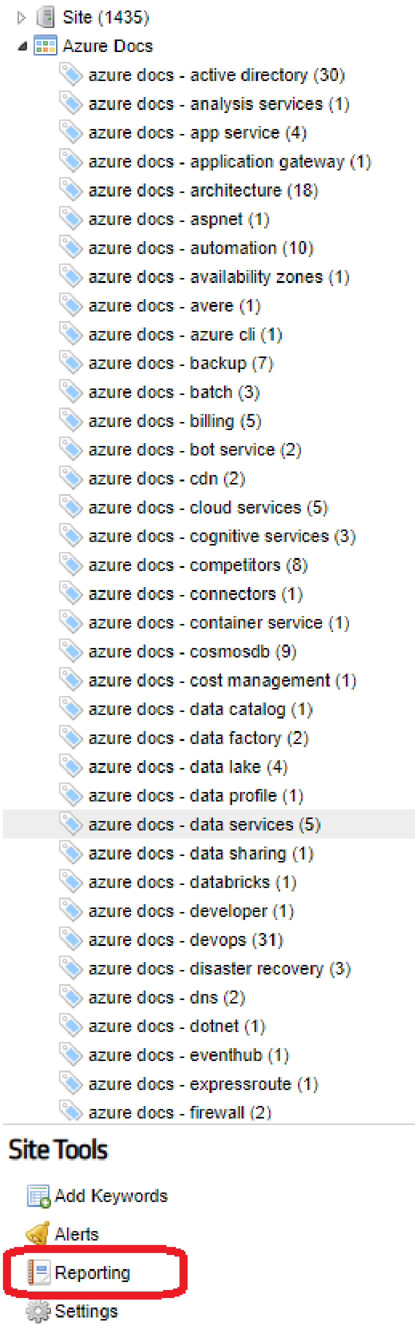
- Is the keyword currently ranking well for a page?
- Is the keyword relevant to the site section? If users search for the term(s), should they expect to see Docs listed on the first page of the search engine results?
- Does the keyword have search volume?

> **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.

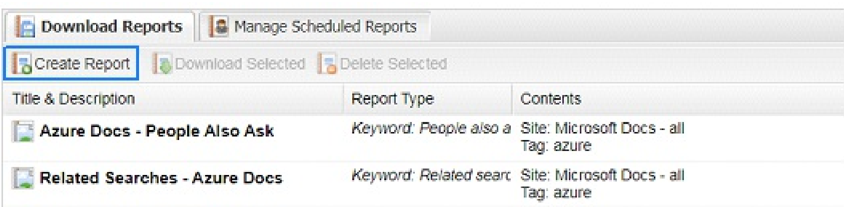
Step 3: Select keywords from the People Also Ask and Related Searches Reports in STAT

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.

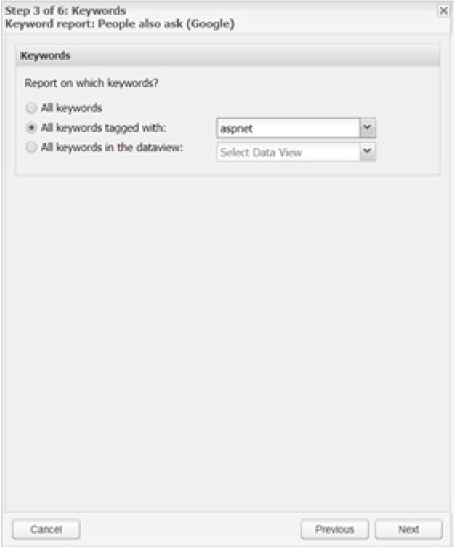


Once you click on the Reporting tab, go to **"Create Report"** to pull the People Also Ask and Related Searches Reports.



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:

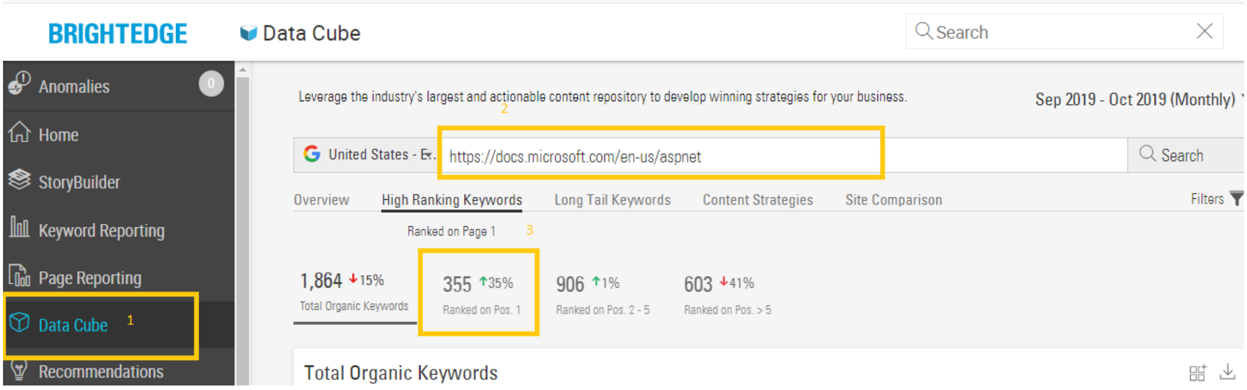
- Keywords containing "documentation"
- Keywords containing "vs" or "comparison"
- Keywords containing "tutorial"
- Keywords containing "what is"
- Keywords containing "definition"
- Keywords containing "how to" or "how do"

This is just a starting point, but it can help to narrow down the list a bit.

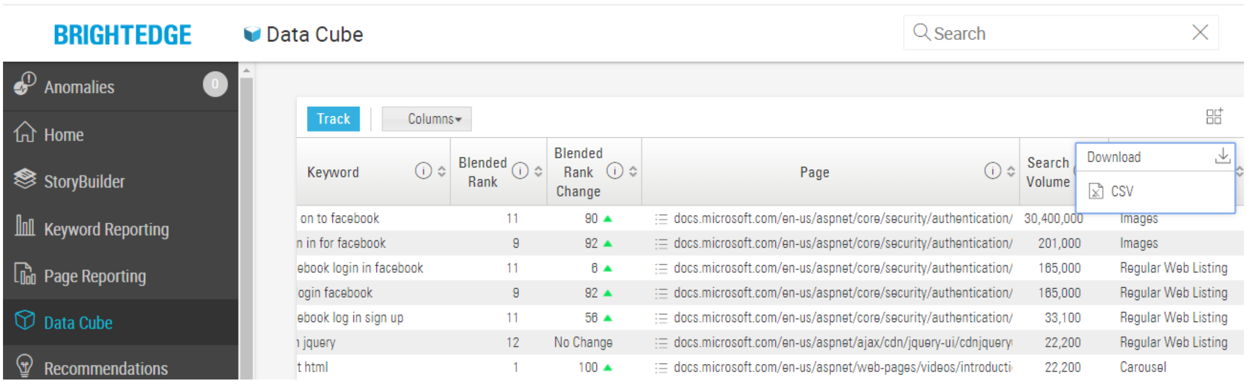
Step 4 - Select keywords from the BrightEdge DataCube tool

> **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.



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 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

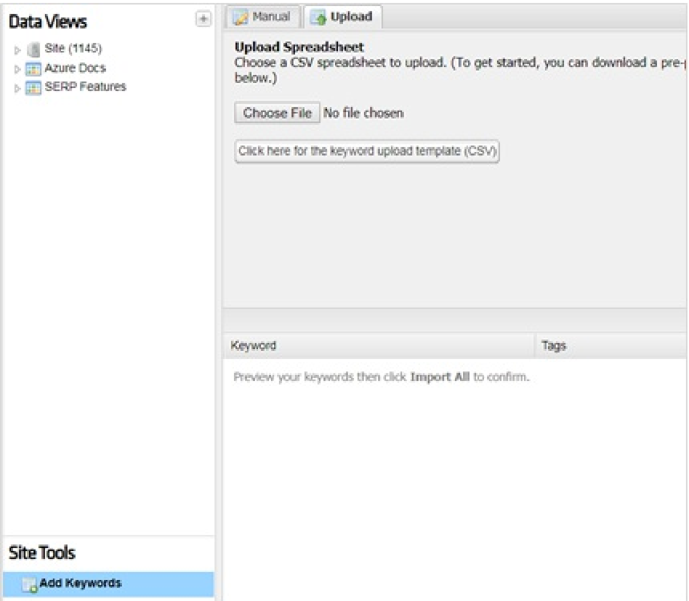
The goal number of tracked keywords is 200-400 total per site section, so it's important to whittle down the lists to the best keyword selections possible.

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

7/7/2021 • 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`.

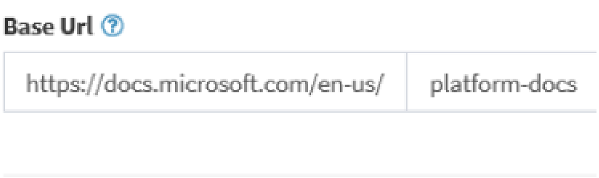
Define a base URL

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:



> **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.yml* file in each folder. Here is what you need to do:

1. Create an *index.yml* 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.yml*, 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

6/24/2021 • 2 minutes to read

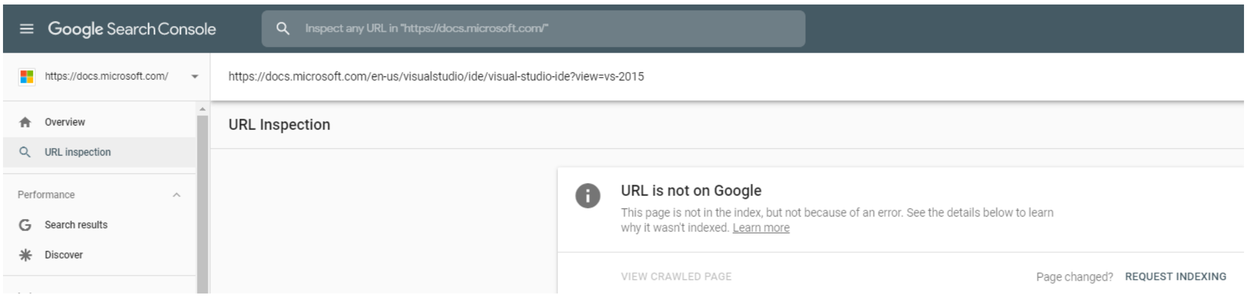
Having difficulties finding your content in Google, or wondering about indexing of the content in Google search? You can use some of the tips below to troubleshoot the issue.

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.



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.

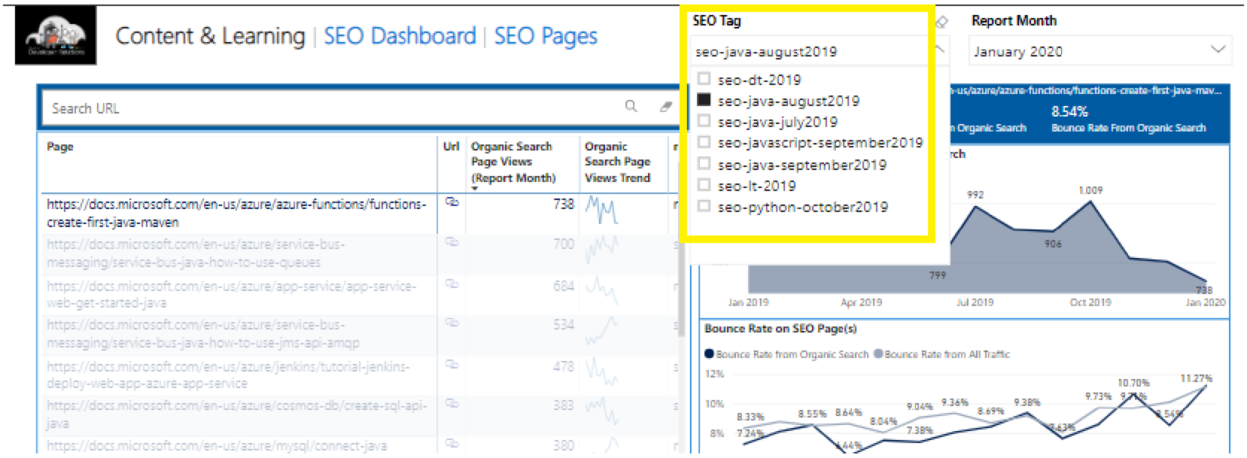
Release docs and measure results

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

6/29/2021 • 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 and is one of the scores calculated in the Content Health score and report. You can use the Content Health report to see how your content set is doing overall and the Content Performance report for individual articles.

As an article ages from its last freshness update, based on the ms.date, it goes from freshest to stale. For FY22, the freshness groups have been updated as follows.

DEFINITION	CONTENT PERFORMANCE REPORT FORMATTING
Freshest (0-90 days)	No conditional formatting
Fresh (91-365 days)	Light yellow
Stale (> 365 days)	Dark yellow

Why does freshness matter?

The **ms.date** metadata value is published at the top of most 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 do I prioritize for the most impact?

You can use the content performance report to find your highest traffic articles.

1. Plan a rotation so that your highest traffic articles and getting started/acquisition materials are reviewed and updated most frequently to ensure customers have a good experience.
2. Consider archiving low traffic and obsolete articles to reduce maintenance costs.
3. You can, in some cases, ask for vendor support for freshness reviews.

How to mark an article as "fresh"

Content freshness reporting is calculated from the manually defined **ms.date** metadata value in your markdown and yml files. 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**.

DO: When you do a full review of an article or make major updates to a page, manually change the ms.date value to the date you made your edits.

DON'T: When you make typo fixes or minor, quick updates without a full review of the page for accuracy, don't update the date.

> **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.	Use either the verbatim comment report or the Drill down report.
Check for GitHub issues for the article.	On the **Issues** 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. You can also use the GitHub issue request report
Check for open pull requests in the public repo.	See the pull request report.

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 ms.date value older than a year. 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

1. **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.
2. **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.
3. **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."
4. **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.
5. **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."
6. **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.
7. **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 un-publish 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

7/7/2021 • 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

7/8/2021 • 3 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

Keep when:
• The page covers a product/service that is still supported **and** can be kept healthy and performant.
 	
• Or, the page is legally required on our live site regardless of lifecycle, health, or performance.
	Important!
	Confirm with product engineering, marketing, customer support, and CELA that there is no legal requirements to keep this content.
	
	Archive if:
	• The page covers a product/service that is OUT of the support lifecycle -OR- obsolete
	• The page is for an on-prem product is on an extended support cycle and seldom maintained
	• The page no longer maintained. (>24 months since reviewed)
	• The page has very low page views https://aka.ms/docs-low-no-pv
	
	Result:
	• Page(s) moved to the previous versions site.
	• GitHub issues and public PRs are turned off.
	• Content marked as `NOINDEX,NOFOLLOW` for search engines. Only if archive content falls under high value product (HVP) category, can they be marked `INDEX,FOLLOW` : (* Product is under regulatory supervision)
	  • Windows XP + greater (including .NET Framework) (*)
	  • Windows Server 2008 and forward (*)
	  • All versions of .NET Framework (not including .NET Core) (*)
	  • Office 2007 + greater
	  • SharePoint 2007 + greater
	  • Exchange 2007 + greater
	  • SQL Server 2008 + greater
		Important!
		Confirm with product engineering, marketing, customer support, and CELA that there is no legal requirements to keep this content.
		
		Cut if:
		• the page was published accidentally
		
		• Or, if this content is legally required on docs.microsoft.com -and- the page is either performing poorly or offers no value to customers

- 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
- When on-prem product is on an extended support cycle the content owner can archive the content to previous version of the site, which contains a disclaimer. This is especially true when content is not maintained.
- When an Azure or any other product-specific service is obsolete the associated content should be archive.
- When the content is moved to previous version site, it should be marked NOINDEX to remove it from organic search indexes. This will allow latest content to appear in search engine result page.
- If archive content falls under HVP (high value product) category, it can be marked INDEX,FOLLOW on the

previous-version site.

- HVP list: The first three bullets are under regulatory supervision.
  - Windows XP and forward (including .NET Framework)
  - Windows Server 2008 and forward
  - All versions of .NET Framework (not including .NET Core)
  - Office 2007 and forward
  - SharePoint 2007 and forward
  - Exchange 2007 and forward
  - SQL Server 2008 and forward
- If content is stale in current site and haven't been refreshed in 2 years its an archival candidate for review and making decision.
- If content is not maintained on current site, the GitHub issues and public contributions are turned off, and content does not need to be discoverable in organic search it can be archive.
- If content is not generating any traffic and page views data shows zero traffic, it should be consider to archive especially if it's not a legal requirement to keep such content on the current site.

Content removal guidelines

- If a content set is accidentally published the content owners have the authority to delete such content from a repo or make it offline from Docs Portal.
- If the content author decides that content is not valuable to end user, either because it is not a valid because technology is deprecated they have the authority to remove the content or redirect to another target on Docs or any other Microsoft site.
- To request that a set of docs be taken offline, complete the docset-change form: https://aka.ms/publish-on-docs/docset-change

Archive and retire articles and folders by using the Docs Portal

6/22/2021 • 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:

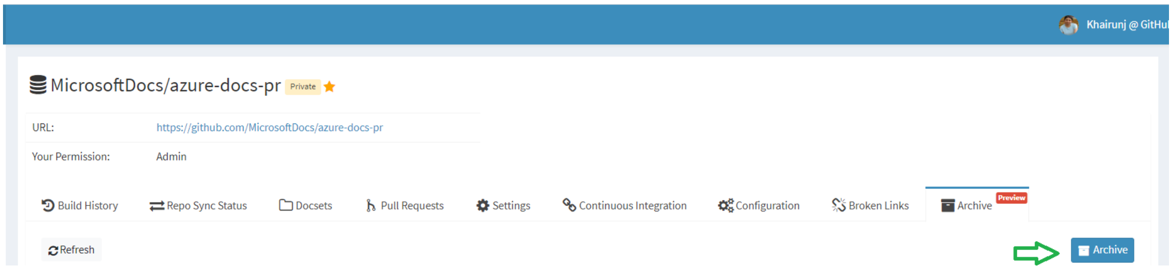
1. Confirm with your product partners that they're ready to archive content.
2. Open OPS and the archive tool.
3. Identify the source and archival destination target.
4. Run the tool to generate PRs in the originating repo and the destination repo.
5. In each PR, review output, fix warnings, and sign off.

Let's get started.

Open the archive tool in OPS

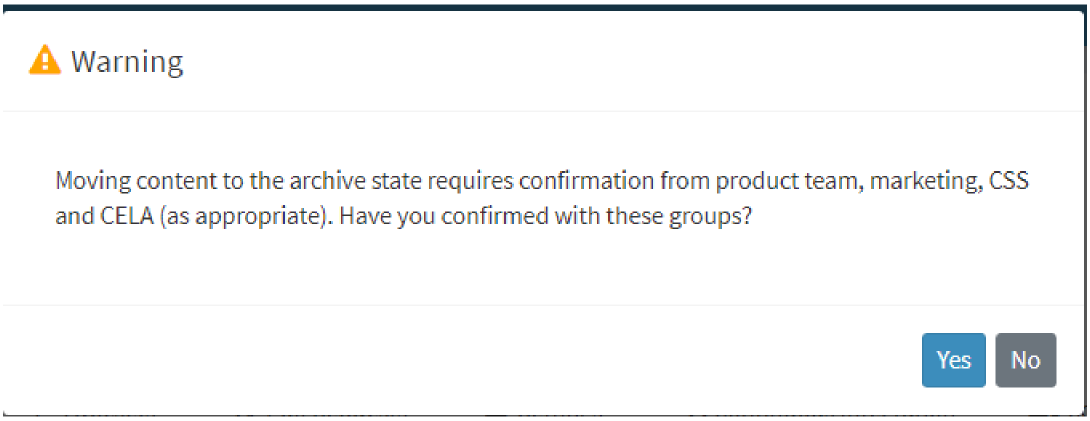
After you've gotten confirmation from your product partners, such as engineering, PMs, legal, and marketing, you can start the archival process.

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.



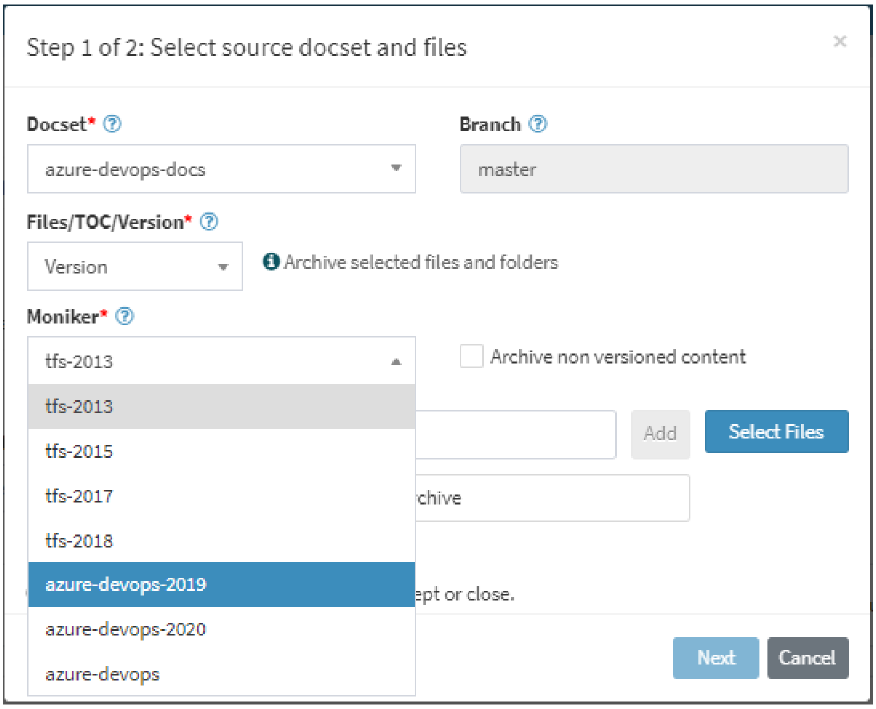
Specify files to archive

1. When the tool starts, select **Yes** to confirm you've checked that the product team is ready to archive.



2. In the **Step 1 of 2** dialog box, specify the articles or version of the product to be archived.

3. 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**.

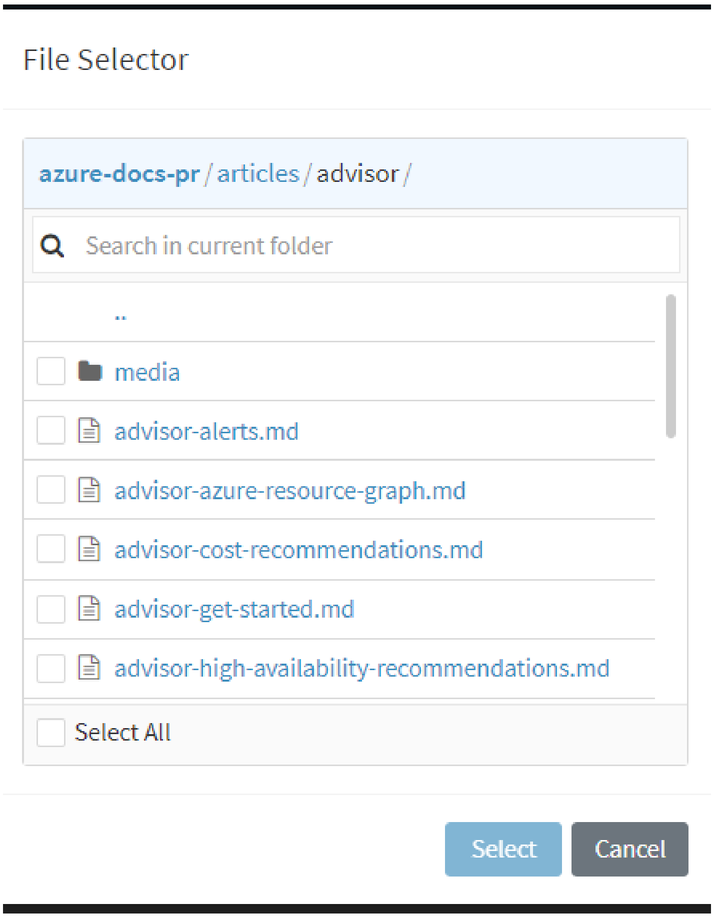


FIELDS	INSTRUCTIONS
Docset	Choose the docset name that contains the content to be archived. The azure-docs-pr repo has only one docset named `azure documents`. Docset isn't a folder name in the repo. It's a grouping of content in an OPS platform. Usually, repos should have one docset only. But some repos have more than one docset.
Branch	This value is always `master`. 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.

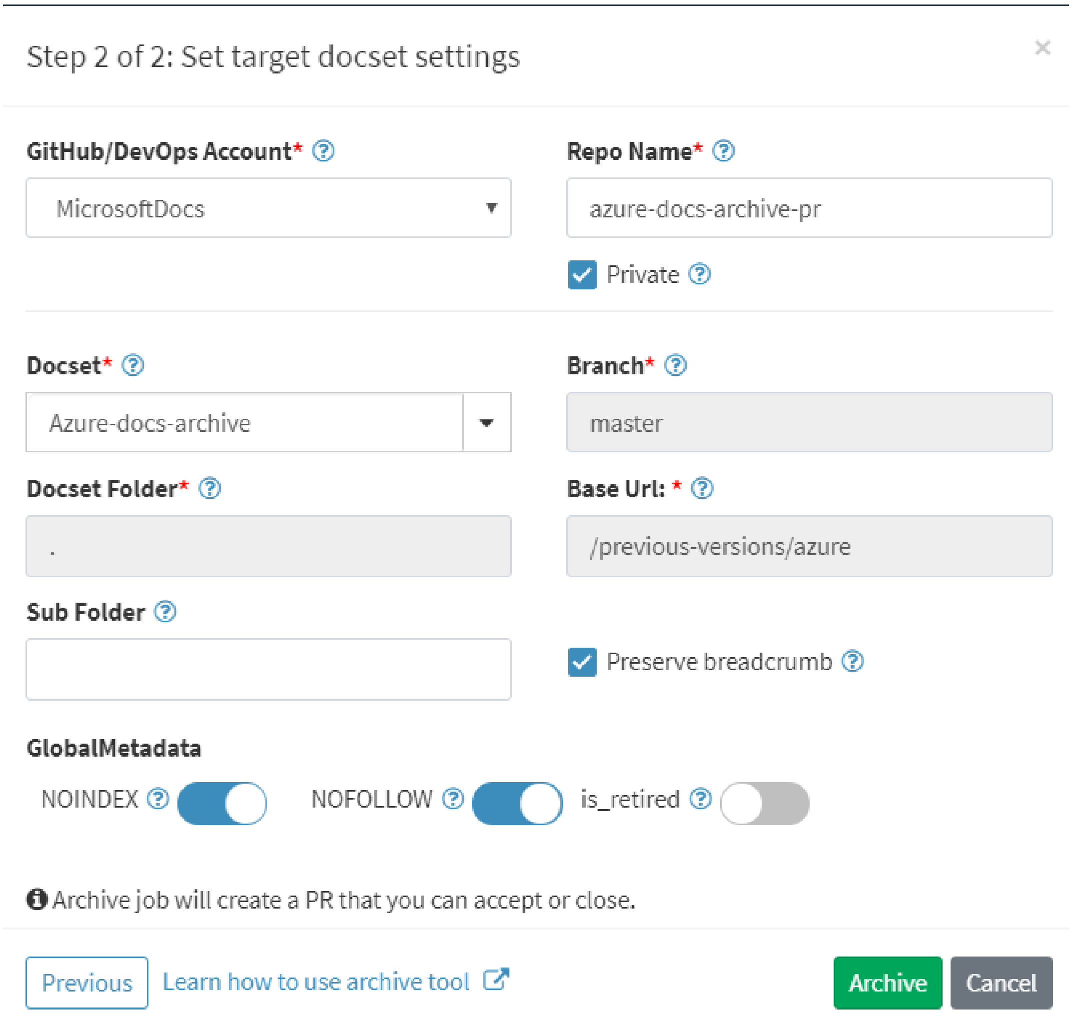


4. After you select files or folders, select **Next** in the **Step 1 of 2** dialog box to start specifying the target destination for the archive.

Configure the target archive destination

1. 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.



2. 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.

3. 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.

4. 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:

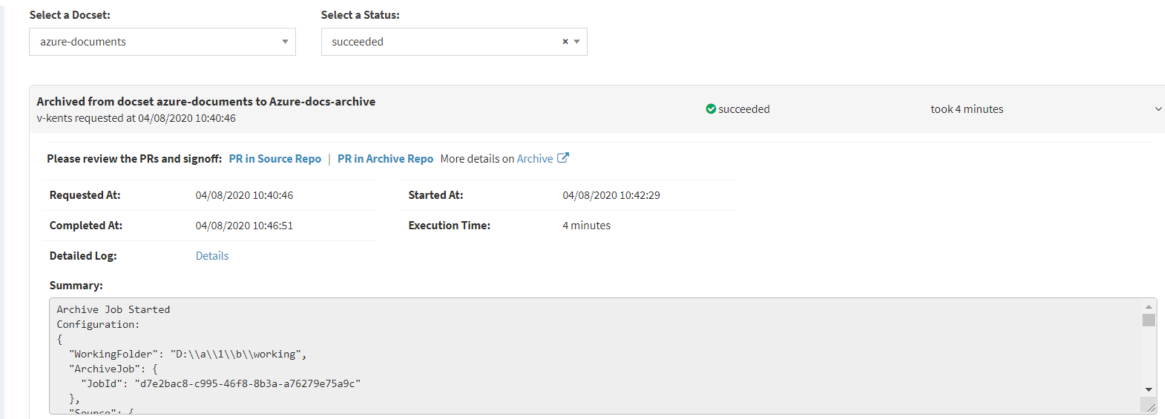
OPTIONS	DESCRIPTIONS
NOINDEX	Provides crawlers with instructions for how to crawl or index web page content. Tells a search engine not to index a page. Because this content is archived, a life in search engines is usually undesirable. If the content needs to be discoverable in Google and Bing, turn off 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	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 **is_retired** toggle. **Archive banner text:** "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 text:** "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."

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.



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.

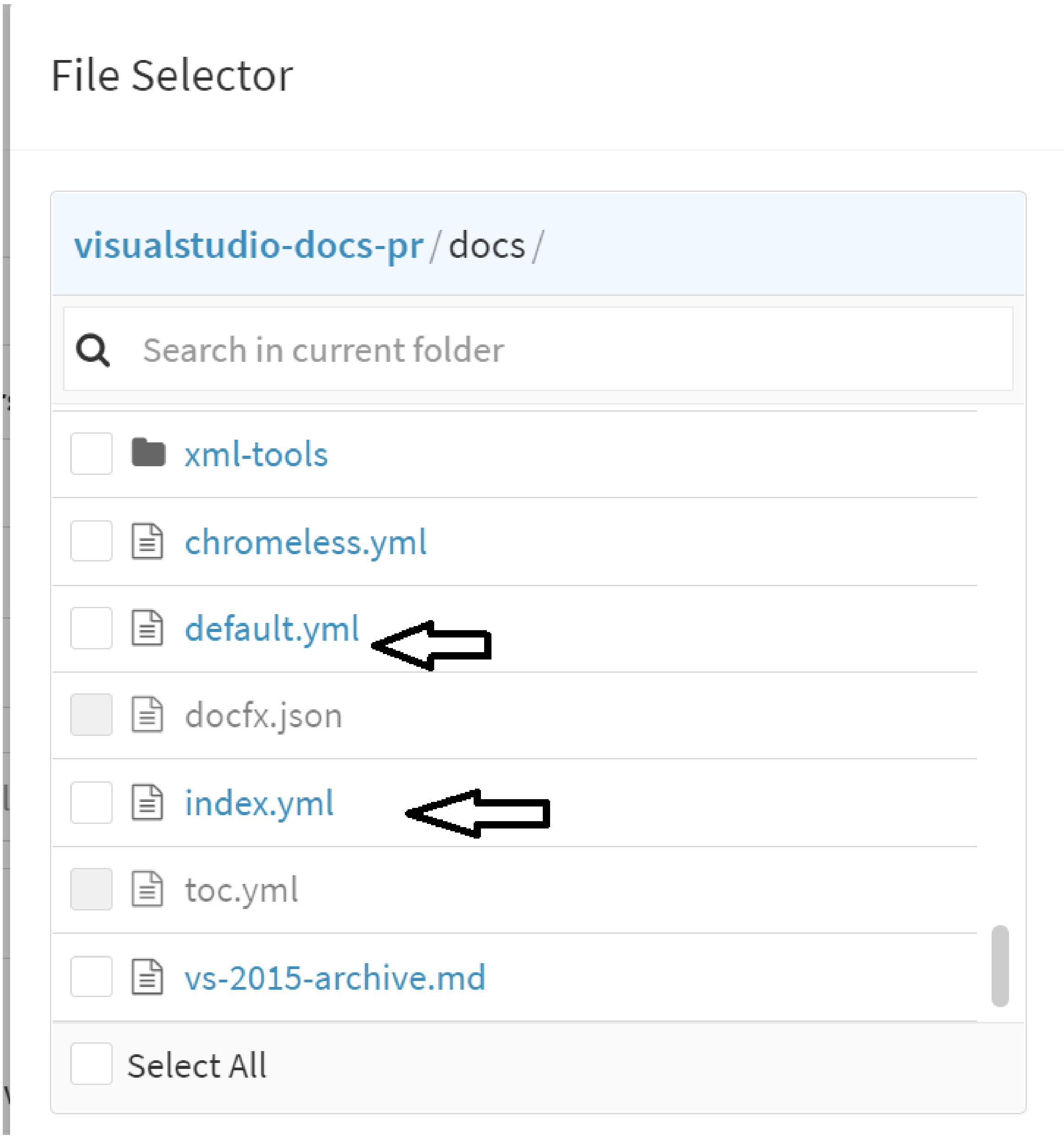
Archive YAML pages

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.



> **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 Content Localization Production PM Team 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

7/9/2021 • 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 obsolete services and retired product content should be moved to archive site regularly.

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 https://docs.microsoft.com/previous-versions/. For more information about archiving a few articles from a folder or a TOC node, see Archive and retire articles and folders using Docs portal.

Summary of the required steps

When we archive an entire docset, we don't move the physical location of the content. Nor do we change docset and repo name. Instead, we make a few edits in the configurations and update the base path in Docs Portal to achieve the required results.

Here's the list of steps that are required to present the entire docsets content to previous version site.

- **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)
- **Breadcrumb**
  - Add "Previous Versions" to breadcrumb
- **Docset base path and redirection** – from "/xxx" to "/previous-versions/xxx"
- **BI tracking**
  - docs_archive: manual (to distinguish from archive tool in BI data)

Modify files on the `main` branch

Make the following edits in the `docx.json` file under the `globalMetadata` section:

```
"build": {
    ...
    "globalMetadata": {
      ...
      "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.

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"` .

Modify the `.openpublishing.publish.config.json` file to add config `"open_to_public_contributors": false` as shown in the following example:

```
{
  "docsets_to_publish": [
    {
      "docset_name": "archive-feature",
      "open_to_public_contributors": false,
    }
  ],
}
```

Modify the breadcrumb on the `main` branch

In the `toc.yml` file, add `Previous Versions` to the breadcrumb. The following shows an example of the file before and the file after adding the breadcrumb:

Before

```
- name: Docs
  tocHref: /
  topicHref: /
  items:
  - name: .NET
    tocHref: /dotnet/
    topicHref: /dotnet/index
    items:
       ...
```

After

```
- 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 the breadcrumb to show `previous-versions` in the link, make sure to add that reference to all `tocHref` and `topicHref` lines in the `toc.yml` file.

Verify changes are ready on the Docs review site

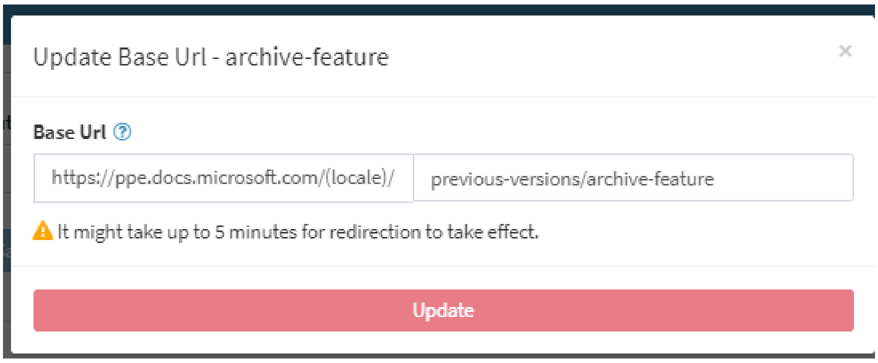
At this point in the process, you can verify the following criteria before you continue:

- Archive disclaimer is shown.
- Breadcrumb is correct.
- Feedback section and Edit button are disabled.
- Article URL is still the old one and not changed to include `/previous-versions`.

Change the base path

Now, change the base path from `/xxx` to `/previous-versions/xxx` on the Docs Portal.

1. Go to Docs Portal and find the docsets by using the search filter.
2. 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's an example:



:::image-end:::

This change automatically creates a redirection rule from `/xxx/*` to `/previous-versions/xxx/*`.

Merge `main` branch to `live`

After you've completed the above steps, merge the changes from `main` branch to `live` branch.

NOTE

Keep a few URLs from the current site handy to validate after the merge from `main` to `live`.

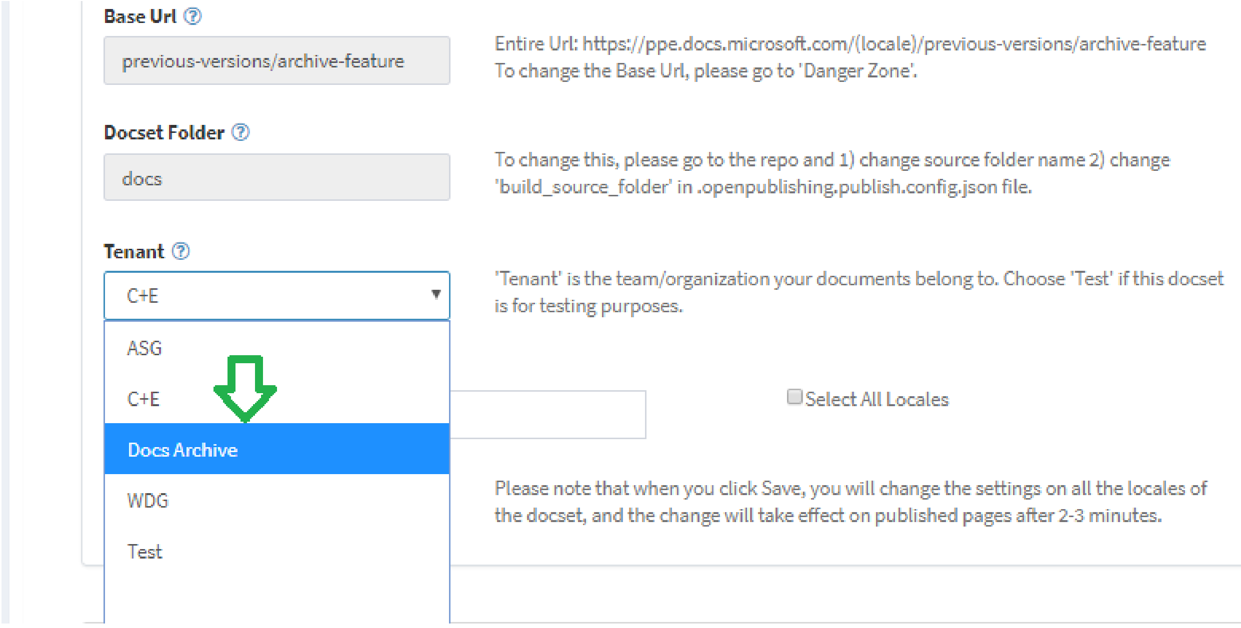
Verify on docs site

After the `live` branch build completes, verify the following items:

- New URL has `/previous-versions/xxx`
- Archive disclaimer is shown
- Breadcrumb is correct
- Feedback section and Edit button are disabled
- Old URL with `/xxx` is redirected to `/previous-versions/xxx`

Modify docset Tenant to "Docs Archive" via the Docs Portal

Go to the Docs Portal and view the docset settings. Go to the **Tenant** field and change the value to **Docs Archive**.



:::image-end:::

This modification doesn't change content location but it's helpful to track which docsets are archived.

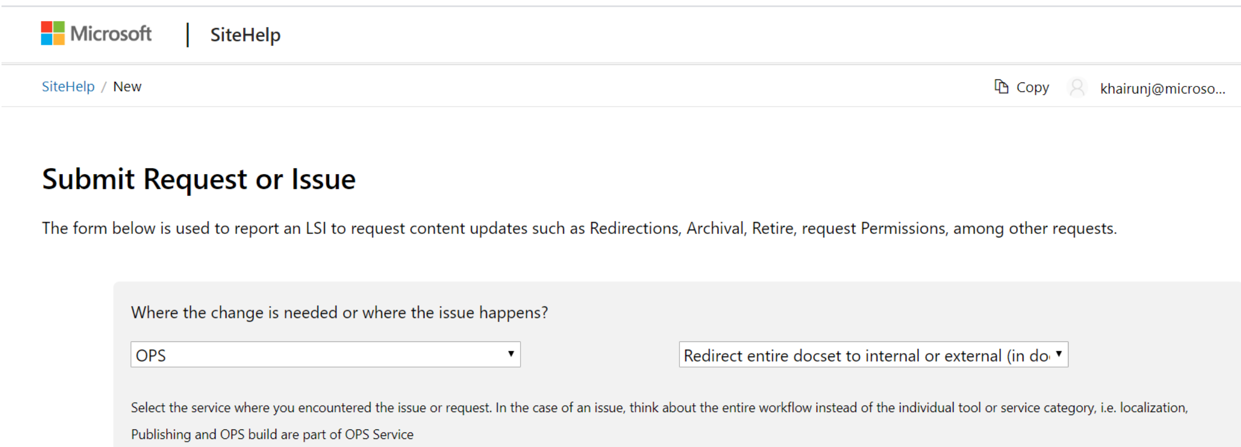
Archive a Repo

To archive an entire repo, you'll have to archive each docset one by one following the steps above.

After all the docsets are archived, make the repo private in GitHub settings.

Request to set up redirection to an absolute URL

If you need to set up redirects for all the articles on the docset to a specific URL, submit a request and the engineering team will make the change.



:::image-end:::

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 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

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

7/7/2021 • 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. 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

7/7/2021 • 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:

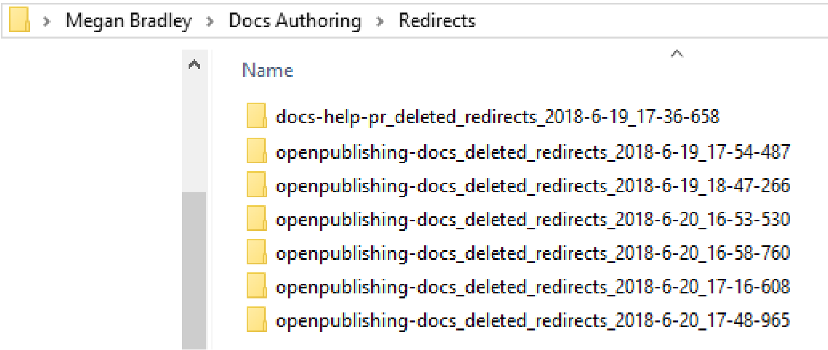
```
---
redirect_url: https://marketplace.visualstudio.com/items?itemName=docsmsft.docs-authoring-pack
redirect_document_id: false
title: Gauntlet VS Code extension
author: meganbradley
ms.author: mbradley
ms.date: 05/18/2018
ms.topic: contributor-guide
---
```

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:



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.

```json
{
    "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.

```json
{
    "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.

```json
    "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 contains 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://CsvJson.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

6/24/2021 • 3 minutes to read

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

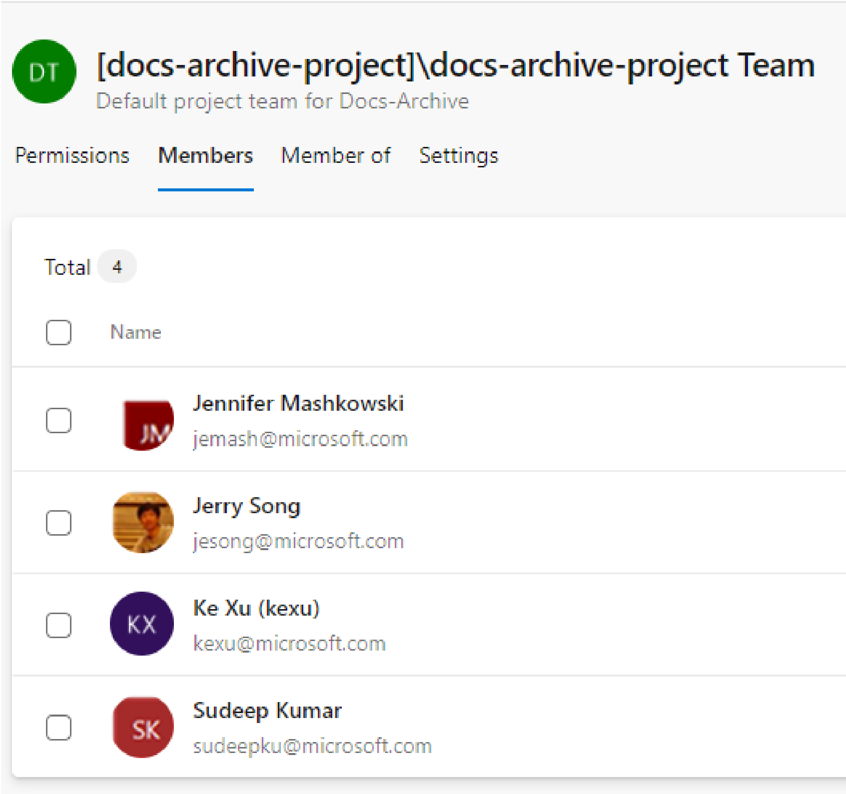
Permissions for managing previous-version content

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

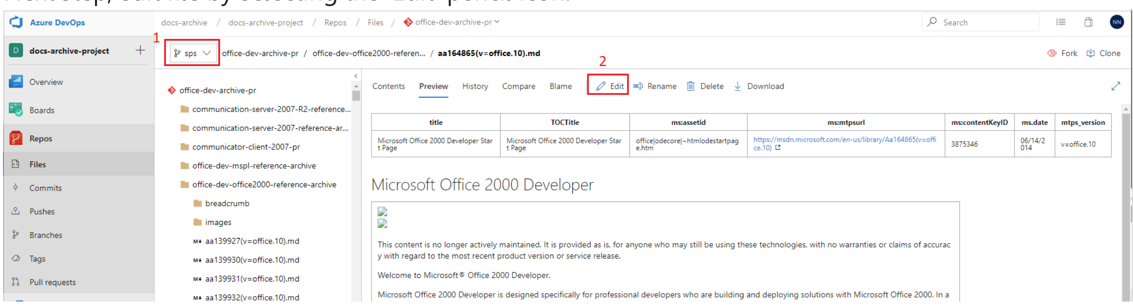
- If you need contributor-level permission but don't see the AAD SG or DevOps group for your content team, contact one of the project administrators:



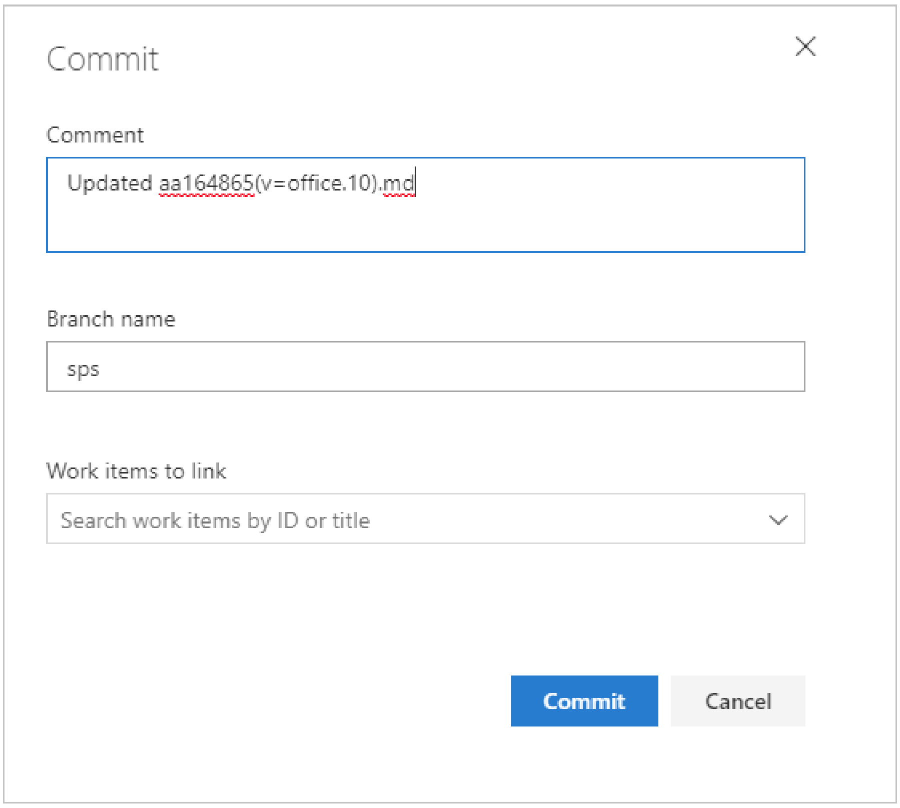
Process to update previous-version content

To make any critical updates to archived content, follow the below steps:

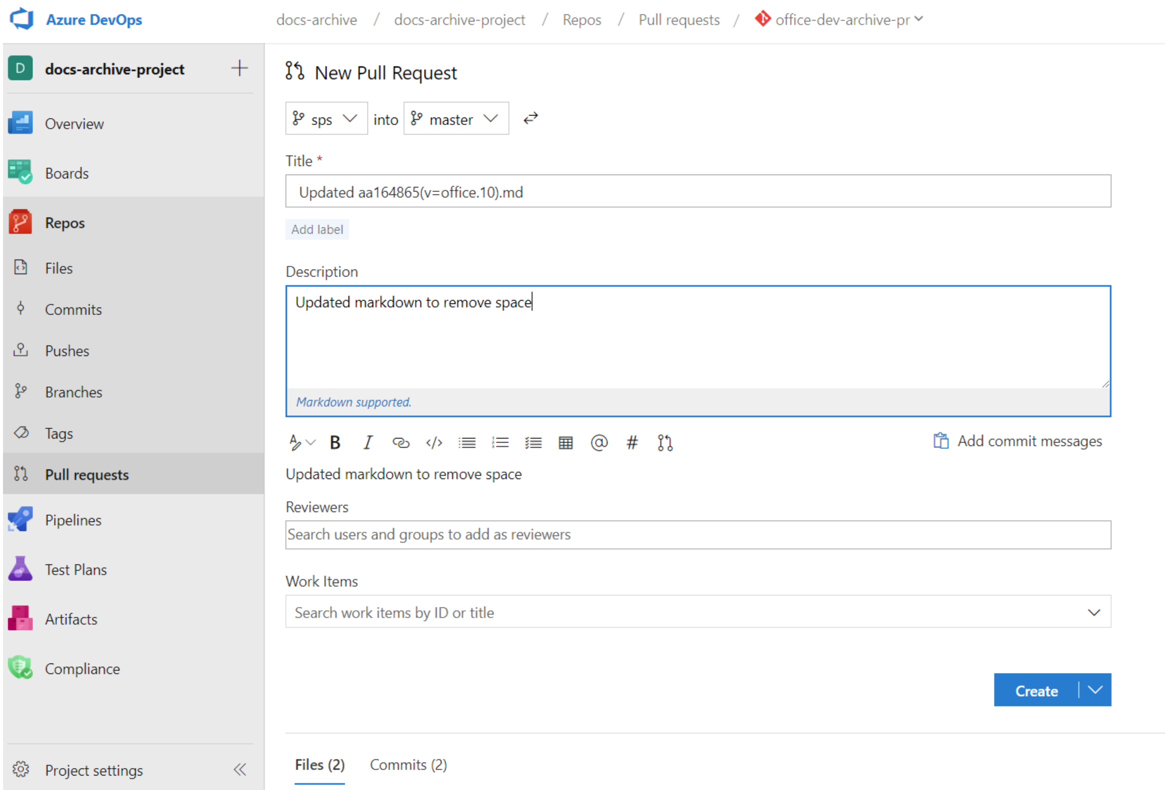
- Navigate to the Azure DevOps repo path of the file.
  - File location can be found at Location of previous-version content
  - File location can also be found from viewing source of the page from browser under meta name="original_content_git_url"
- Content updates can be done directly from browser 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.



- 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.



- 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

7/7/2021 • 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.

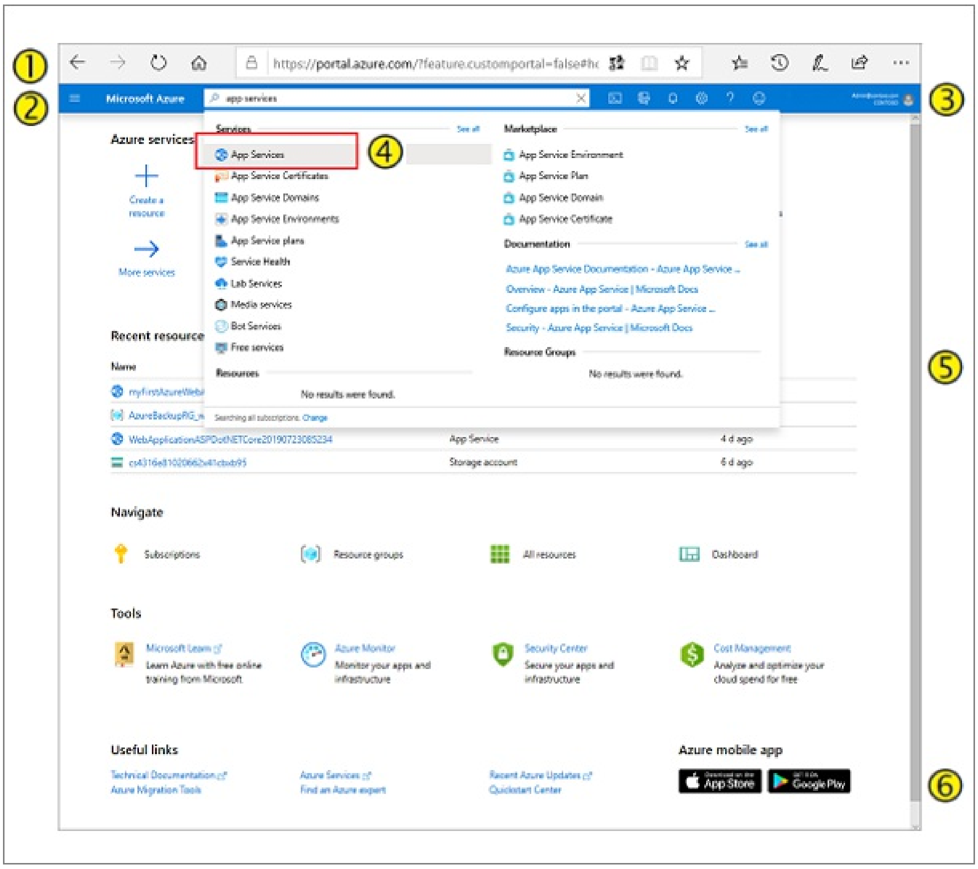
Create and format screenshots in documentation

7/7/2021 • 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:

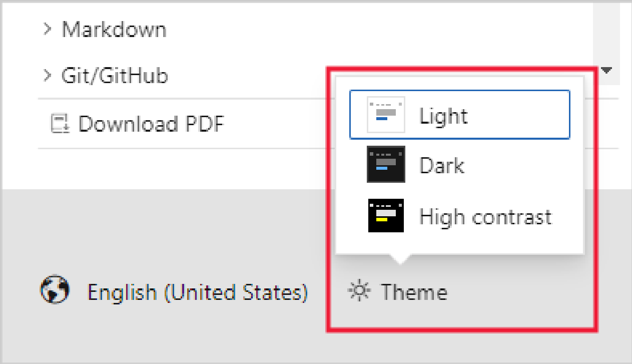
- In Paint.NET, select **Image** > **Resize**. Select **Maintain aspect ratio** to avoid distorting the image.
- In GIMP, select **Image** > **Scale Image**.

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

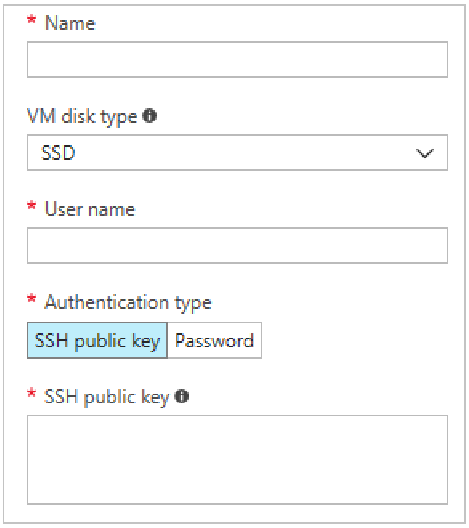
When you're working with conceptual art and icons, test with all three of our site themes: **light**, **dark**, and **high contrast**.

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:



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":::
```

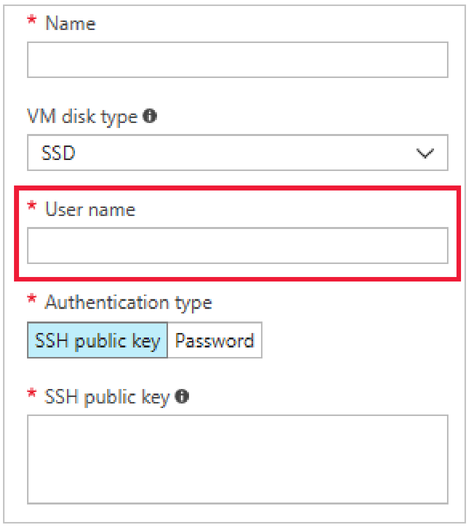
The border property is equivalent to the `mx-imgBorder` style for standard Markdown images:

```
> [!div class="mx-imgBorder"]
> ![alt-text.](media/my-cool-graphic.png)
```

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:



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

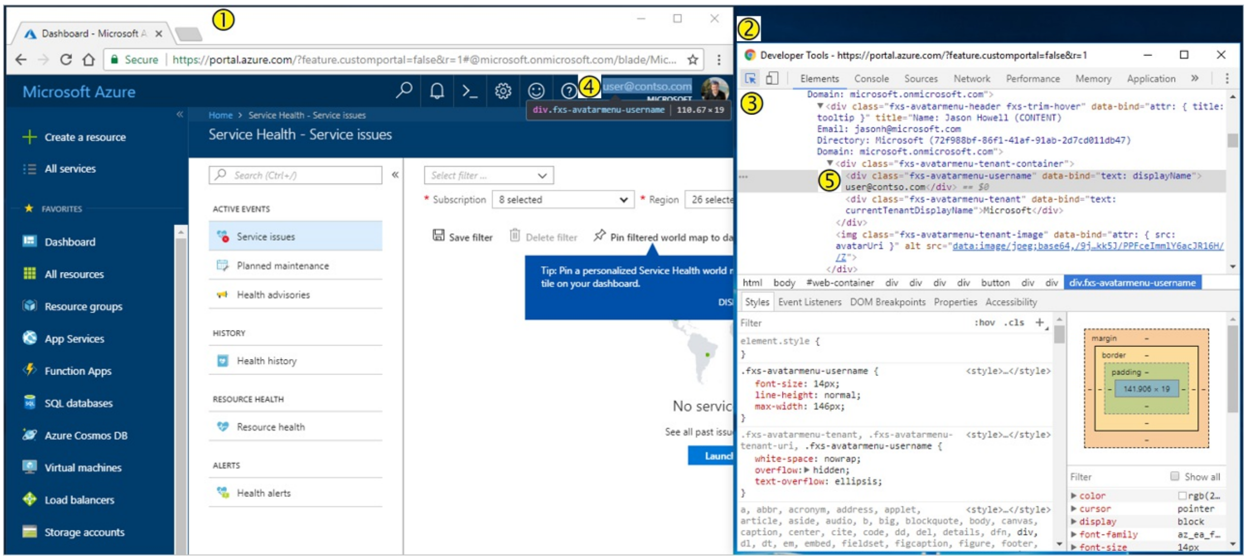
You should replace personally identifiable information with fictitious information, such as Contoso-based names and resources. Here are two resources to help explain acceptable values:

- 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 locates 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

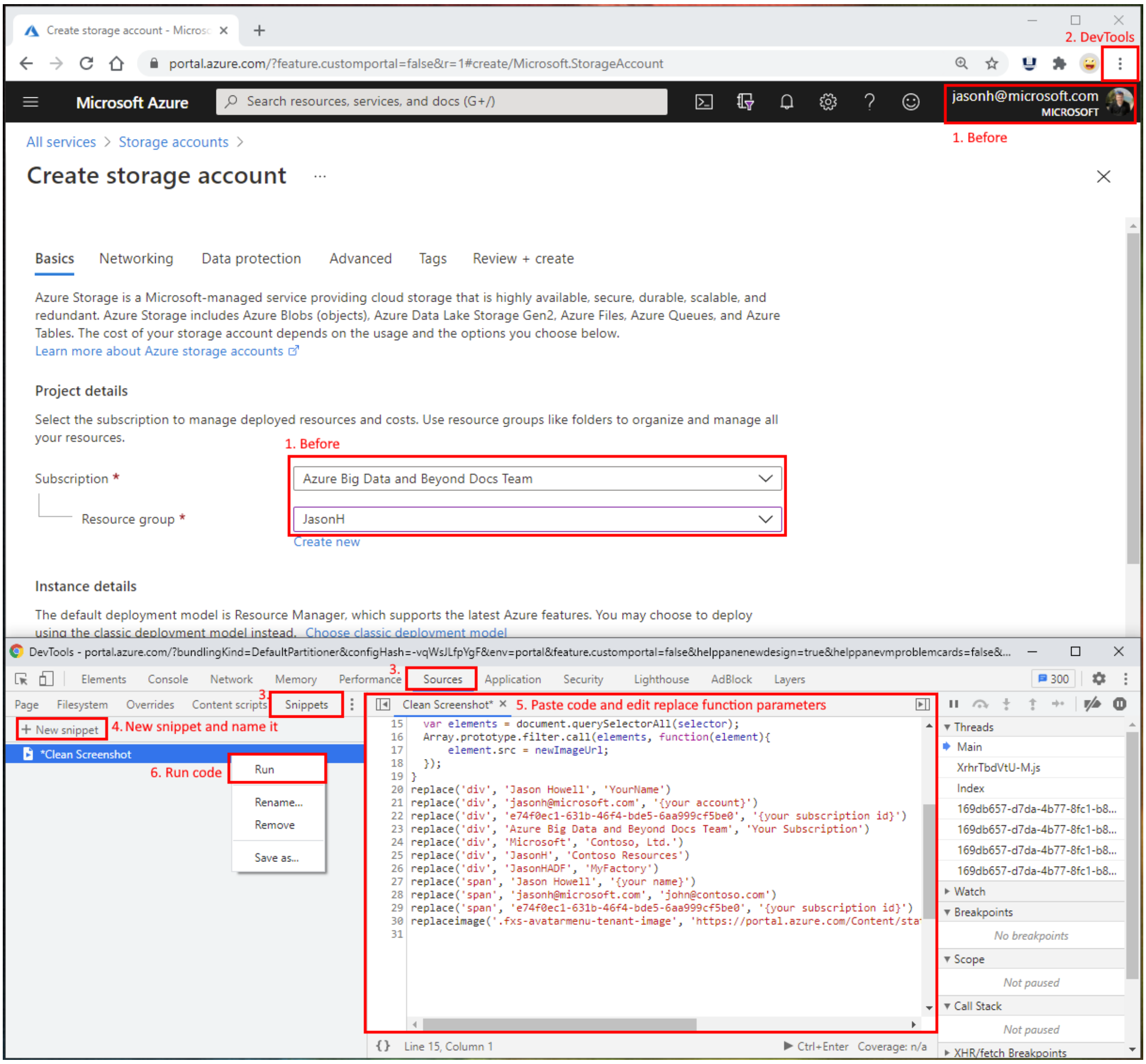
function replace(selector, text, newText) {
  var elements = document.querySelectorAll(selector);
  Array.prototype.filter.call(elements, function(element){
    if (text === element.textContent) {
      element.textContent = newText;
    }
  });
}

function replaceimage(selector, newImageUrl) {
  var elements = document.querySelectorAll(selector);
  Array.prototype.filter.call(elements, function(element){
      element.src = newImageUrl;
  });
}

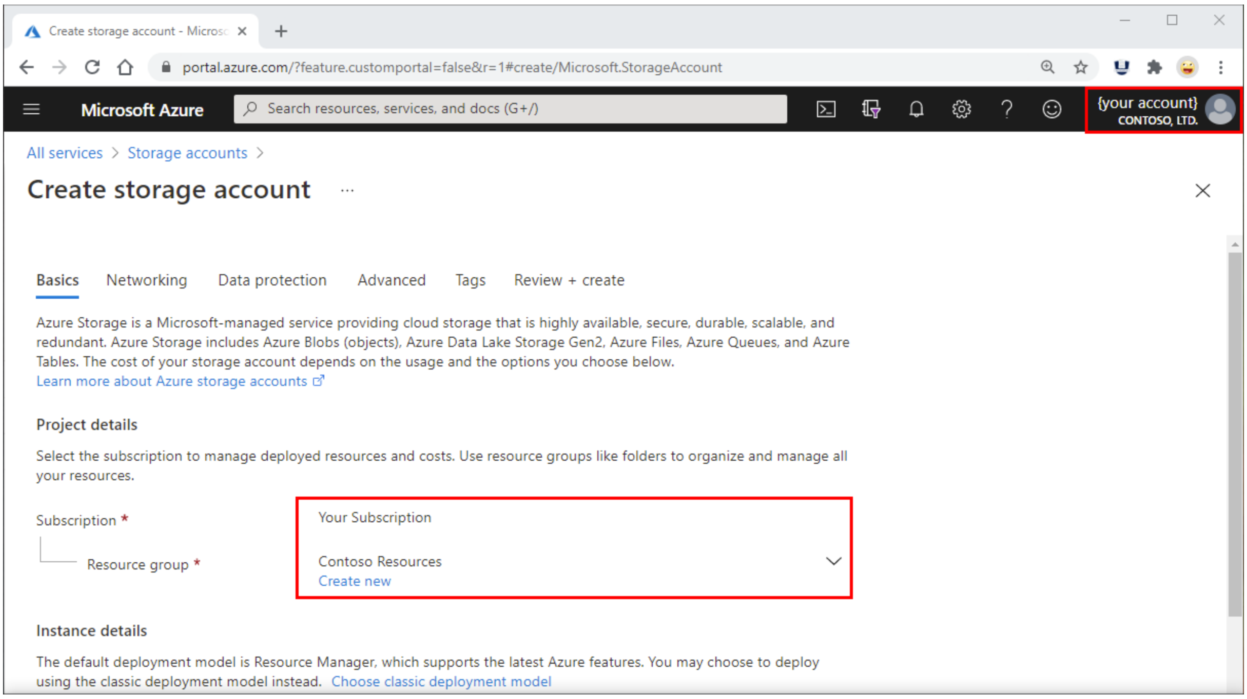
// Provide your real values in the second parameter to be replaced with the fictitious values in the third
parameter
replace('div', 'Your Name', 'YourName')
replace('div', 'youralias@microsoft.com', 'user@contoso.com')
replace('div', 'abcdef01-2345-6789-0abc-def012345678', '{your subscription id}')
replace('div', '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:



After script has run:



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.

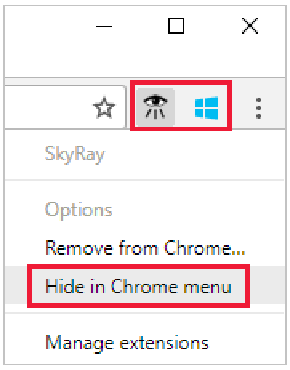
Hide browser customizations

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)

7/7/2021 • 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.

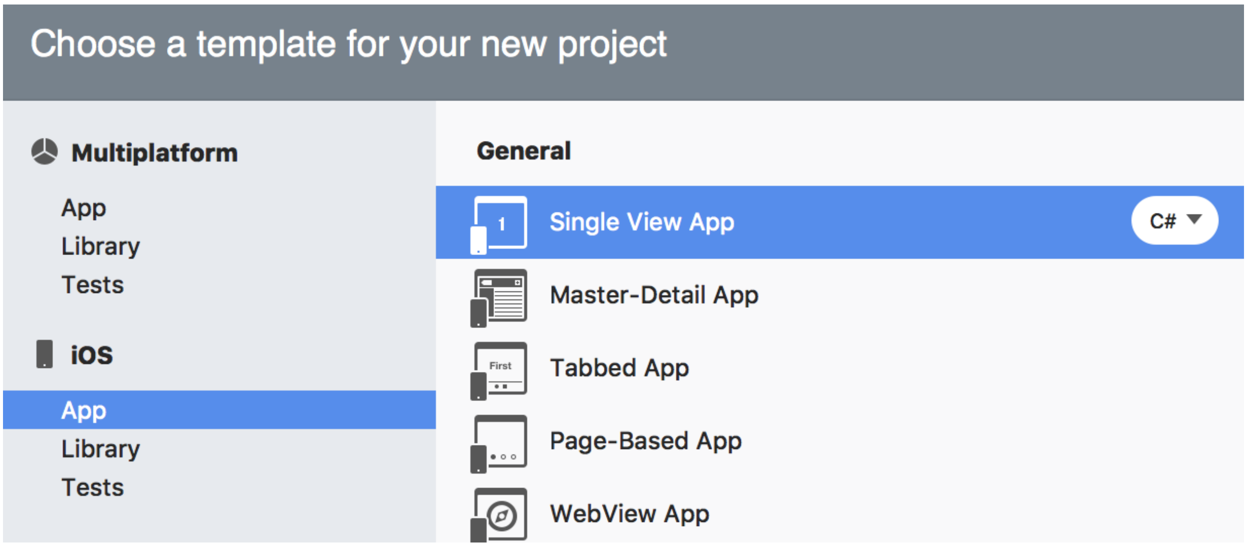
Examples

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

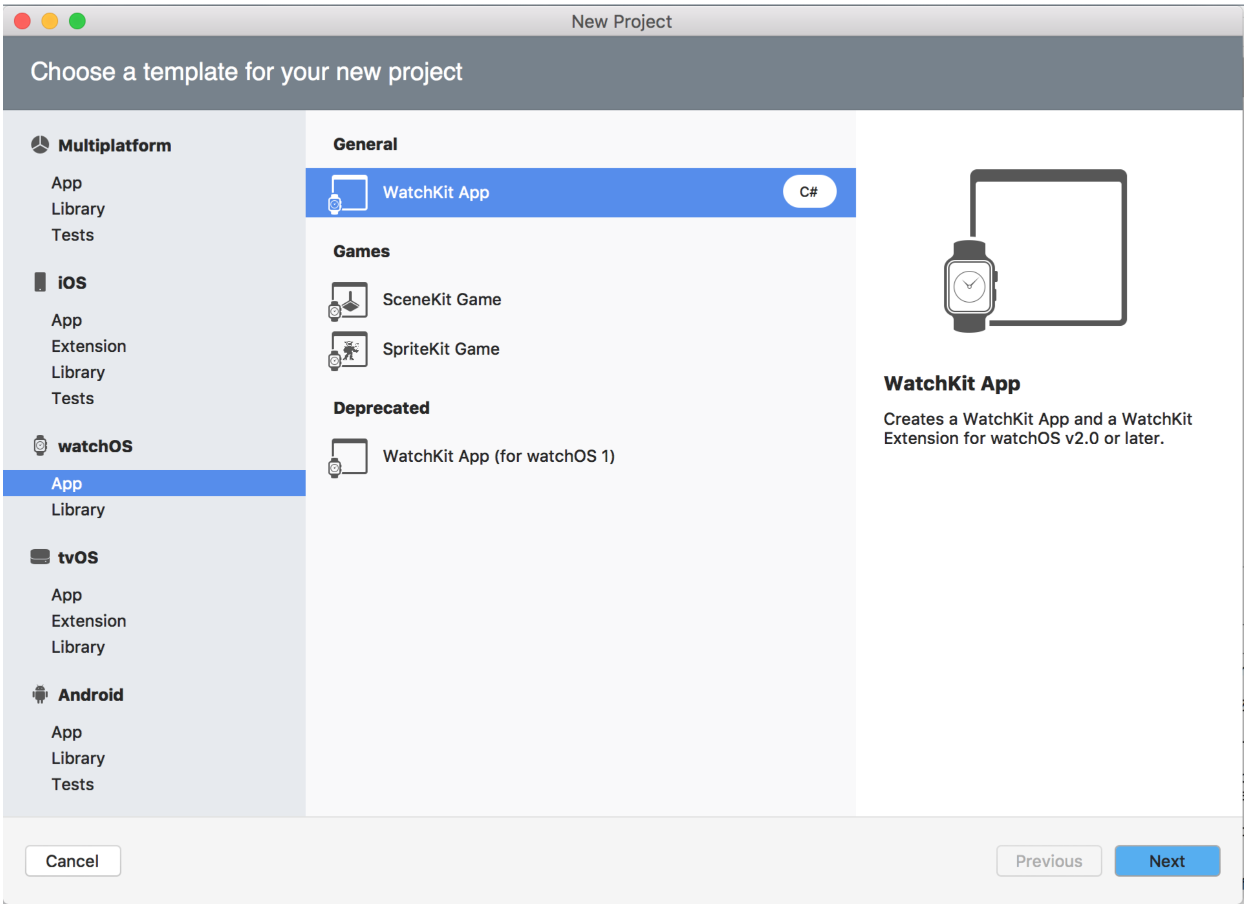
Here's an example of using a cropped image inline, and the full image expanded. The cropped image highlights the most relevant part of the full image.



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:



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

7/7/2021 • 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.

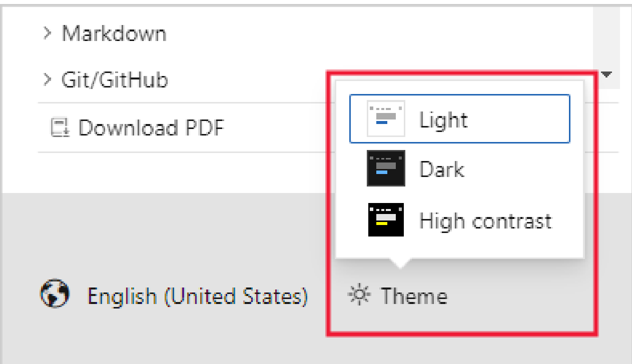
  Because icons are purely decorative, alt text is redundant and should not be used. If you get a build validation warning for missing alt text on an icon, the solution is to use the image syntax for icons as described in Images.

  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.

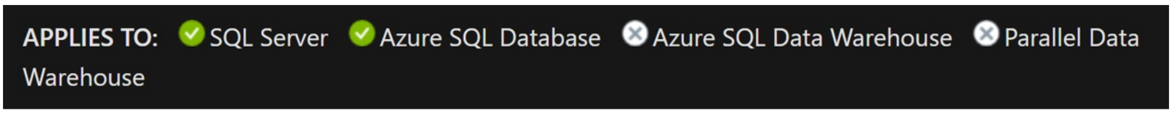
Test multiple themes and transparency

When working with conceptual art and icons, test with all 3 of our docs.microsoft.com themes: **light**, **dark**, and **high contrast**.

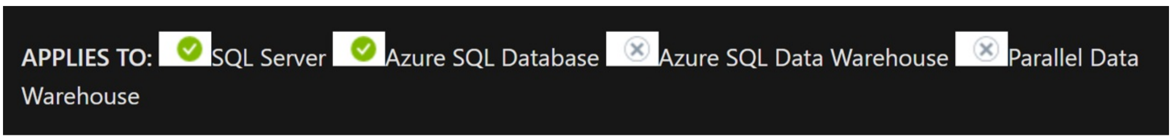
Images with backgrounds and transparency need to work with each theme.Conceptual art and foreground text can be unreadable on the Docs site set to the dark theme if a transparent background is used. Transparency is useful in branded icons and other non-text artwork.



- Correct use of transparency on icons:



- Incorrect background on icons:



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

7/19/2021 • 6 minutes to read

This article describes considerations for using animated GIF graphics, and helpful tools and approaches to make animations.

Why use animated GIFs?

- They can instruct users, illustrate problems for customer support, or mock up a new idea for the design process.
- They're easy to create.
- They're supported natively by effectively all browsers so your audience doesn't need to install a plug-in to see it.
- Compared to a sequence of still images, they're less cognitively demanding. Your viewer doesn't have to mentally interpolate frames between still images because the frames actually exist. For example, it may be easier to watch an animated video of how to tie a knot than to follow step-by-step drawings of the same process.
- 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 the 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's a bitmap without any image compression. That's why 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, and so on), 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. It should do so without expecting the viewer 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. The goal is to include anything that a visually impaired user would need to understand the content without seeing it. Including alt text will also help animated GIFs show up in search results and will help localization.
- **Localized**: Consider that it's difficult to localize animated GIFs into multiple languages. Most likely the animated GIF isn't 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

- Visual Studio Code uses many 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 you enable the game bar 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. Execute 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 recorded `.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. The CSI department contact is 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 many online services that help edit and optimize GIF files. Use these sites with discretion since they aren't Microsoft sites, and don't 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 lossiness. Gifsicle is a command-line tool that can significantly reduce your GIF's file size (also see the help manual). Lastly, this tutsplus 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 https://docs.microsoft.com/ doesn't support stopping animated GIFs, but that feature is encouraged for accessibility purposes. You may own the source code to the web site you're publishing the GIF files on. If so, you can add the following enhancements to improve the user experience with animated GIFs:

- **Play animated GIFs on click**: This tutorial, with an example page showing this method in use, and the code in GitHub repo (requires jQuery dependency)
- **Play animated GIFs on hover**: This tutorial requires Embedly API.
- **Add via plain vanilla JS**: This stackoverflow article explains how to load a static image and then switch to animated GIF on hover.
- **Pause/Play GIF**: 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**: Don't 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.

Alt tags or content text should supplement any animated GIF by explaining any content being demonstrated in the animation out of consideration for blind or visually impaired users.

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

- 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

7/8/2021 • 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	Making a screen capture video	How to use the Azure Cognitive Services .NET Speech SDK for recognition

VIDEO TYPE	DESCRIPTION	ADDITIONAL DETAILS	SAMPLE
Animation	Animation and motion design are used to explain difficult concepts, such as data flowing to the cloud	Making an Animated Video	What is cloud computing
Live action	One or more people speaking to a camera, interview-style	Making a One Dev Question video	How did Xamarin get its name and why do you all love monkeys so much?
Film-at-home	Record yourself speaking to a camera, interview-style, at home	Filming at home guide-doc Filming at home guide-video	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

Video Types for Learn

We have five video types specifically designed for Learn content. Visit Learn video guidance for guidance on where to place videos and script templates.

The following table summarizes the Learn video types.

VIDEO TYPE	DESCRIPTION
Welcome to Learning Path	Introduce the content in a learning path to help customers see why they should take it
Scenario	Visualize your module scenario to make your content more memorable
Concept	Teach a concept using a research-based method that helps learners understand

VIDEO TYPE	DESCRIPTION
How it Works	Show how a product or solution works to solve a common customer problem
Exercise Demonstration	Demonstrate the steps of an exercise to help learners verify they did their work correctly

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

Choose a video type for your module

7/8/2021 • 3 minutes to read

In this article, you will:

- Choose an instructional video pattern for your module
- Download a script template for your video

Overview

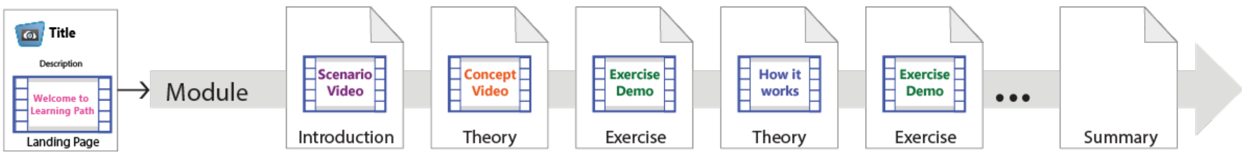
Video enhances learning content in a couple important ways:

1. **It's engaging!** Videos help learners build mental models, provide context, and internalize the content you're trying to teach.
2. **It's motivating!** Videos can provide relevance, increase autonomy, and create context. All of which increase motivation in learning.

To best engage our learners we want to follow specific principles for instructional videos:

1. **Short**:. Video duration should be no more than 3-6 minutes.
2. **Focused:** Keep the content tightly scoped to the learning goal.
3. **Learner-centric:** Use conversational language, be enthusiastic, and always tell the learner why the topic is important.

The following article presents suggested video types for a module based on the unit they might appear (Learning path landing page, introduction, learning content unit, or exercise) and provides links to download the script templates.



Landing page for learning path

Welcome to Learning Path

What is it? A Welcome to Learning Path video provides the rationale for why a learner would want to take the entire learning path. This video type provides motivation, and describes the overall learning objective for the path. See a Welcome to Learning Path video

Why use it? A Welcome to Learning Path video explains why the modules in a learning path are strung together the way they are. Doing so serves two important purposes:

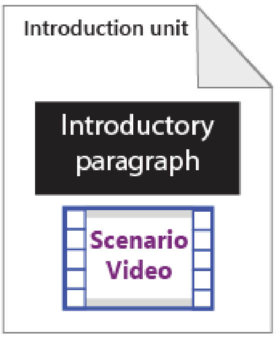
1. It provides the motivation and relevance necessary to help a learner complete the learning path.
2. It focuses the learners' attention on the right content. When they know the overall outcome that's expected, they are more likely to succeed in meeting the objectives of the learning path.

Download the script template

Introduction unit

The introduction sets the context for your module. Putting your content in the context of a scenario will make it more memorable to the learner.

Scenario video



What is it? A scenario video visualizes the problem you plan to solve in a module. Showcasing the scenario as a video will help the learner visualize and internalize the skills they will achieve as a result of completing the module. See an example scenario video

Why use it? A scenario serves a couple of purposes.

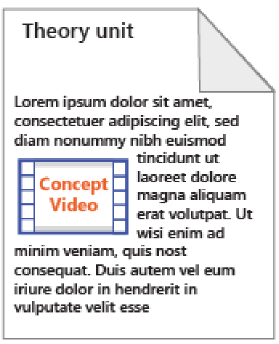
1. It connects the content to a real-world problem which provides relevance to the learner.
2. It provides a short story to engage the learner in the content. Both of which aid in engagement and retention.

Download the script template

Theory unit

Theory units are where we provide the knowledge necessary to gain a deeper understanding of the facts, concepts, and procedures we're teaching.

Concept video

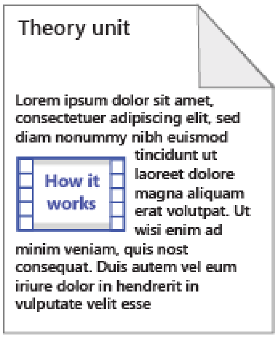


What is it? A concept video is focused on teaching one specific abstract idea or notion. Explaining concepts is fundamental to training and many concepts are easier to learn with the help of animated visuals.

Why use it? Describing a concept can require a lot of text but can be kept short and concise with a video. If you have a concept that is lengthy to describe with words and would be easier to illustrate with motion, we recommend you create a Concept video.

Download the script template

How it works video



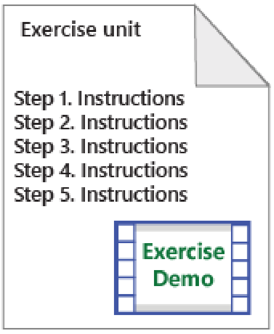
What is it? A How it Works video is a high-level overview of how a product or solution will solve a common problem.

Why use it? This type of video helps the learner visualize how a product or solution works. It can often be difficult or lengthy to describe with only text and diagrams. The addition of animation and a voice with inflection can help to motivate the importance of what you're teaching and make it easier to understand..

Download the script template

Exercise unit

Exercise units provide learners with practical experience in doing the tasks they need to learn.



What is it? This video is a demonstration of how to complete the exercise. It shows every step as it is described in the exercise.

Why use it? Exercise demonstration videos provide feedback to the learner about how they did the task. This video can help a learner identify where something may have gone wrong or provide the reassurance that their end state is correct.

Download the script template

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
How to choose a diagram type	You can use this decision tree to find the diagram type, diagram instructions, and templates to create a diagram.
How to draw a diagram from scratch	Follow these steps to draw a diagram on a blank white board, sheet of paper, or application pasteboard.
How to draw a diagram using a diagram type	Follow these steps to choose a diagram format to capture a technical concept.
How to provide feedback to a diagram	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.
Diagram cookbook	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.
Process mapping	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.
Diagrams as code	An alternative diagraming 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.
About technical diagrams and art	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 their extra elements in the diagram that do no 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 reach 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 have different things in your diagram? A unique thing may have distinct attributes.

6. Create a table of your relationships. Each relationship with 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/diagraming 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.

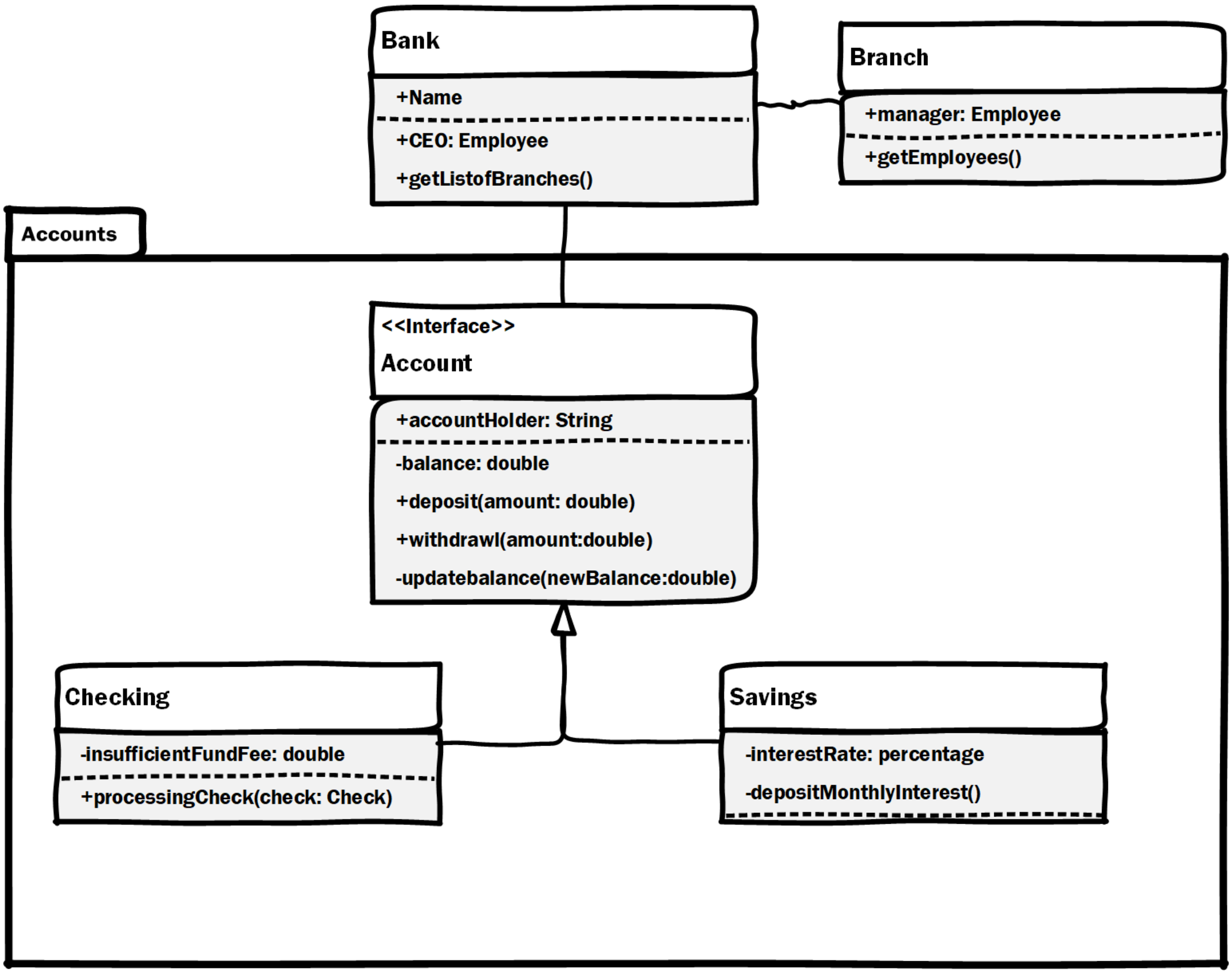
Purpose

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	• **Attribute** - An attribute is a property of an object. It may be represented by a data type. • **Operations** - An operation is an action that the class may be able to take. It is often referred to as a method. • **Interface** - Specifies the externally visible operations of a class, component, package, or other element without specifying internal structure. • **Enumeration** - Describes a data type consisting of a set of named values.

INGREDIENT TYPE	DESCRIPTION
Relationship	• **Dependency** - One object depends on another. • **Association** - This is a generic relationship between two objects. • **Aggregation** - One object is contained in another; however the object will remain if the container is removed. • **Composition** - One object is contained in another; when the container is removed, the contained object is removed. • **Inheritance** - One object acquires the attributes and operations of another class. The source class is a superclass and the class inheriting properties is a subclass. • **Implementation** - One class realizes the expression of another class. For instance, a document is an instance of a template.
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
- **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.

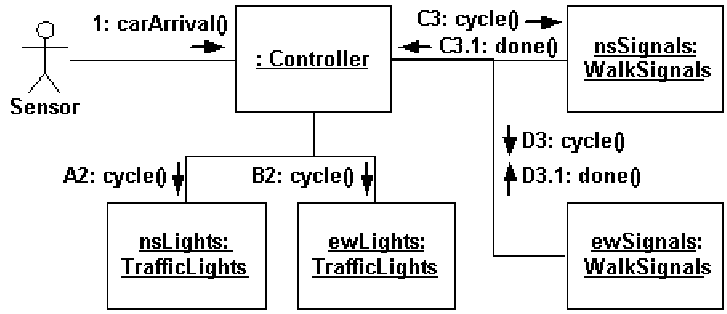
Purpose

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

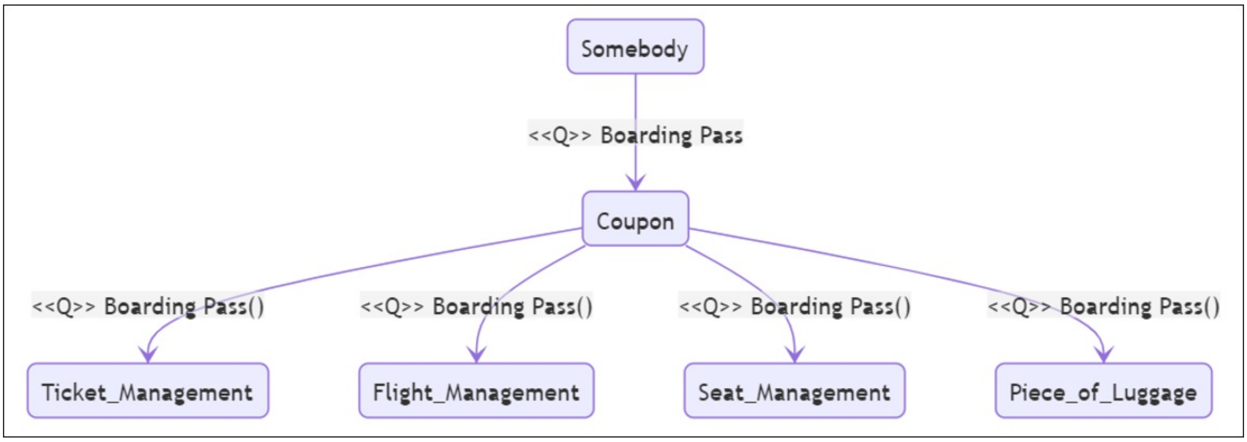
1. Open Visio. Select New and then type UML communication.
2. Select the blank template and select Create.
3. Drag the shapes from the Shapes pane that capture your system.
4. Connect the messages in your system.
5. You can wrap your diagram in the Diagram Overview.

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
- **UML** - https://www.uml-diagrams.org/communication-diagrams.html

<!-- ---------------------------------- -->

Component diagram

Depicts how components are wired together to form larger components or software systems.

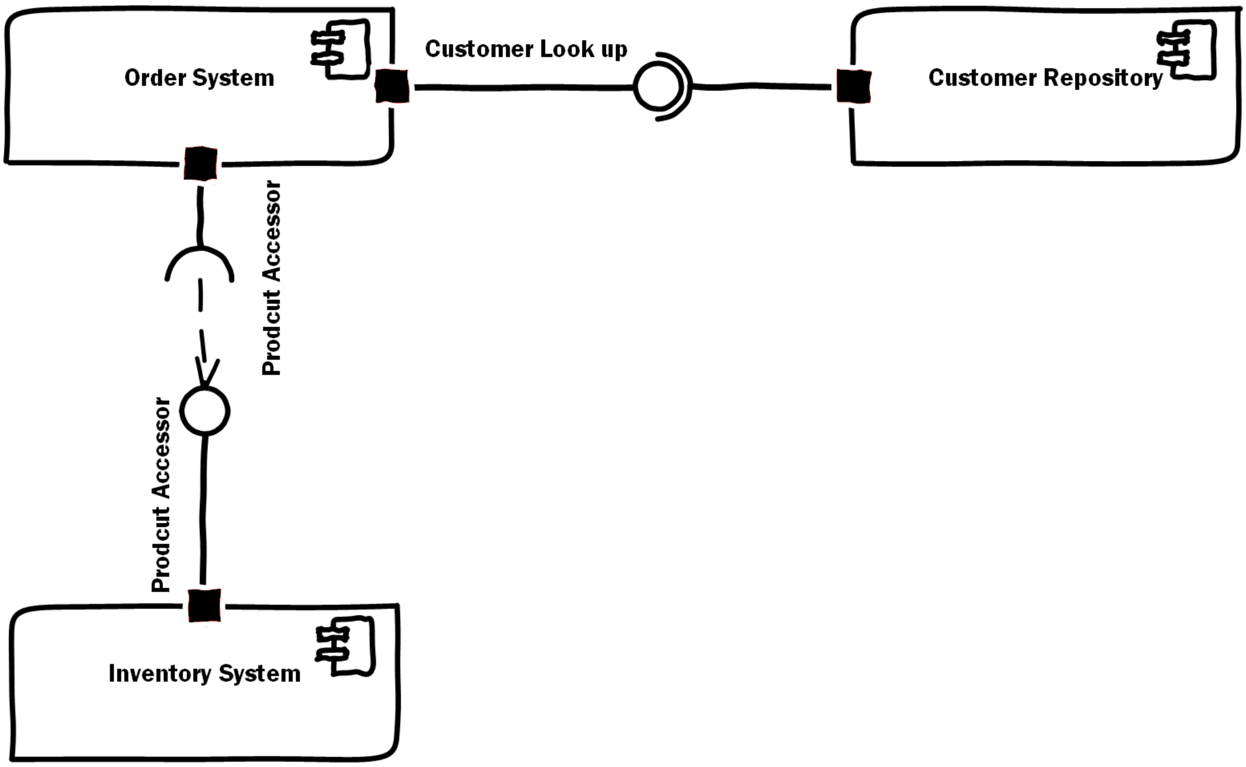
Purpose

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.

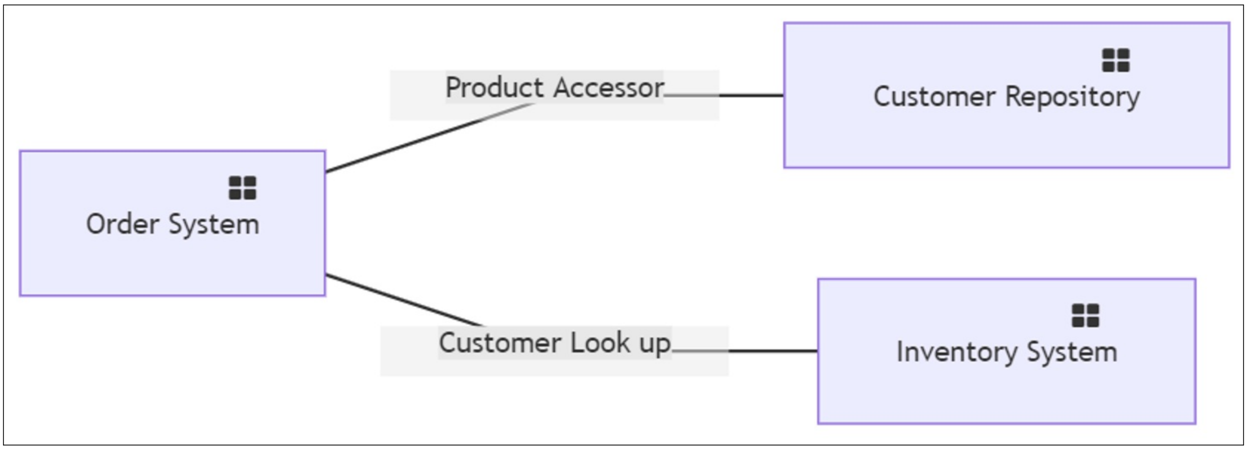
Example in mermaid

Mermaid doesn't have a specific diagram type. You can use the graph diagram.

```
    graph LR
        A["                fa:fa-th-large<br>Order System"]

B["                         fa:fa-th-large<br>Customer Repository"]

C["                       
fa:fa-th-large<br>Inventory System"]
        A-- Product Accessor ---B
        A-- Customer Look up ---C
```



Additional information

- **Agile Modeling** - http://agilemodeling.com/artifacts/componentDiagram.htm
- **Wikipedia** - 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.

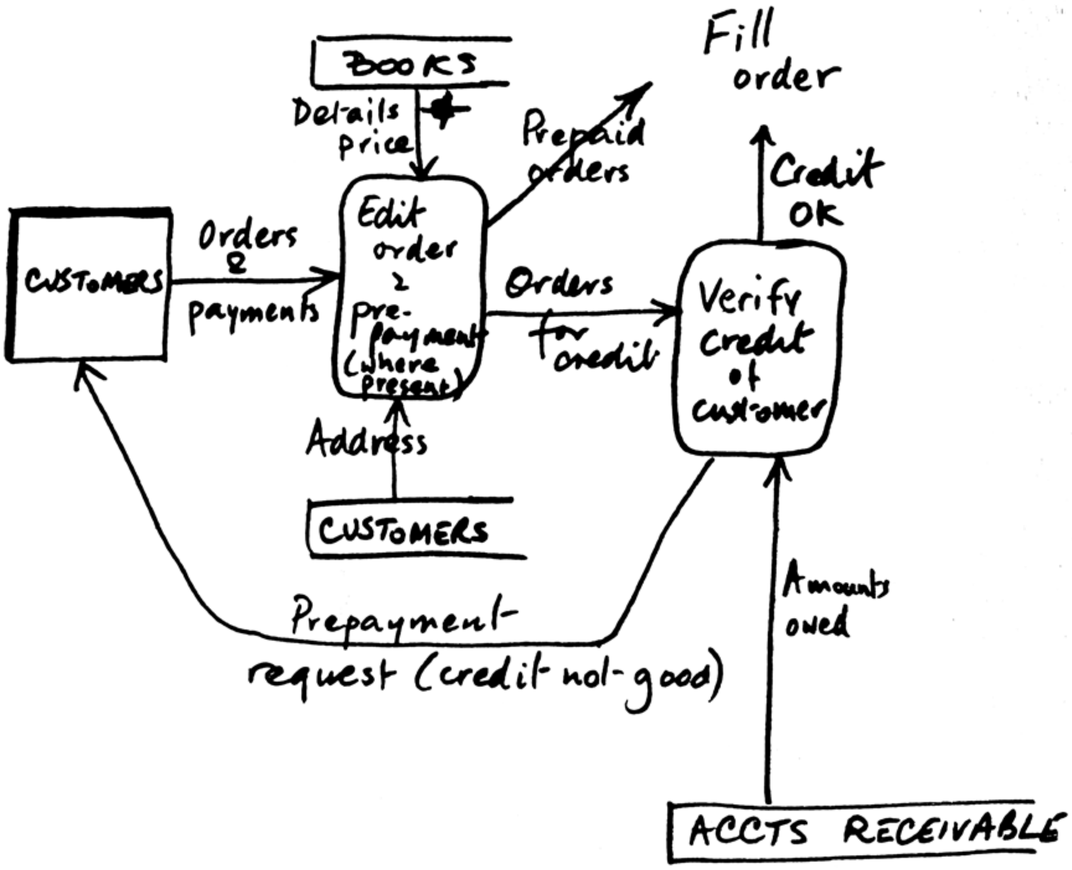
Purpose

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.

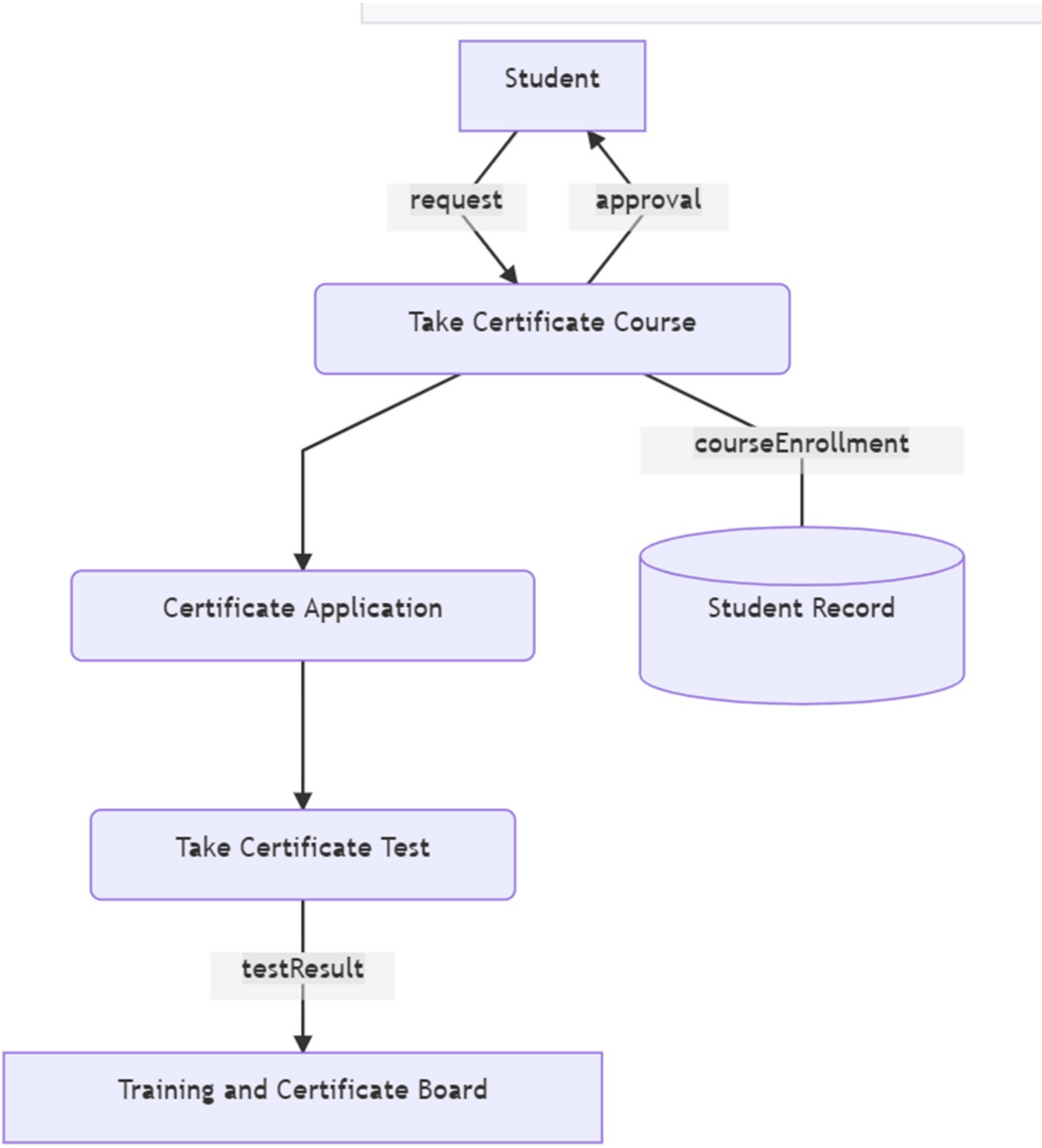
For the Visio instructions see: https://support.microsoft.com/office/create-a-gane-sarson-data-flow-model-diagram-0d7ca68a-0cd2-4022-8cc9-4003bbae3724

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
- **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.

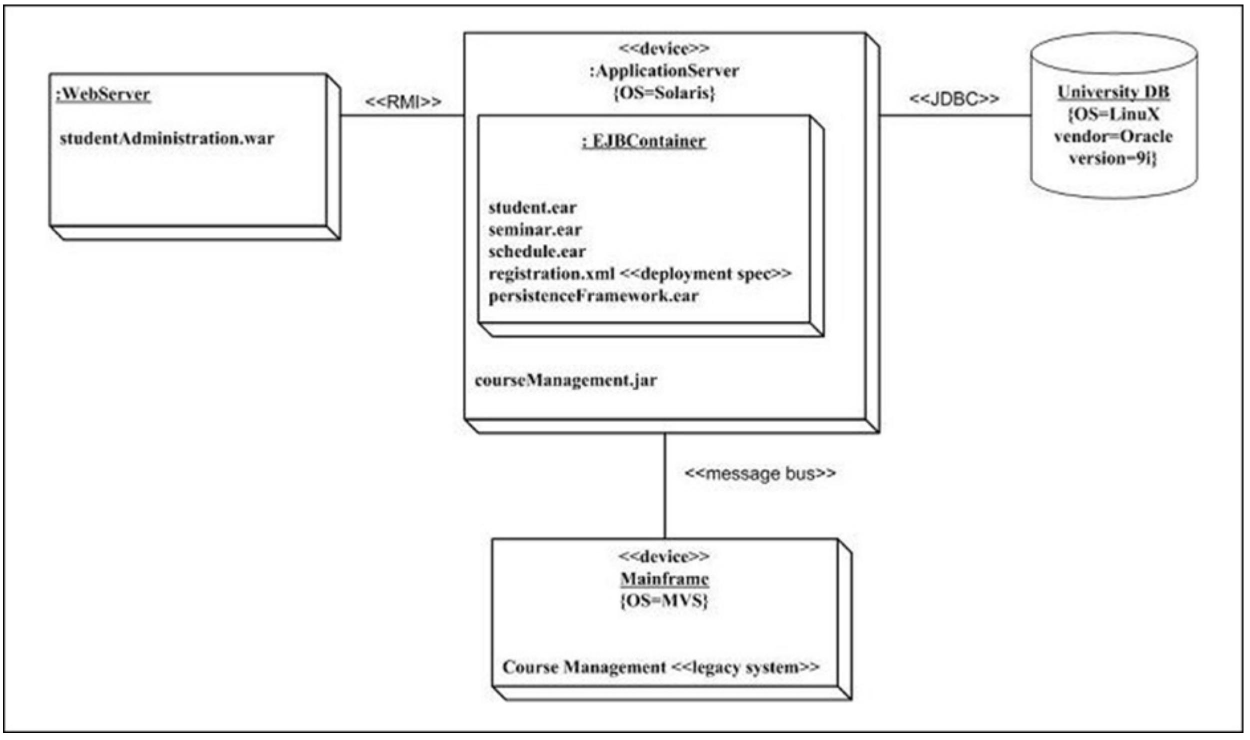
Purpose

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
- **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

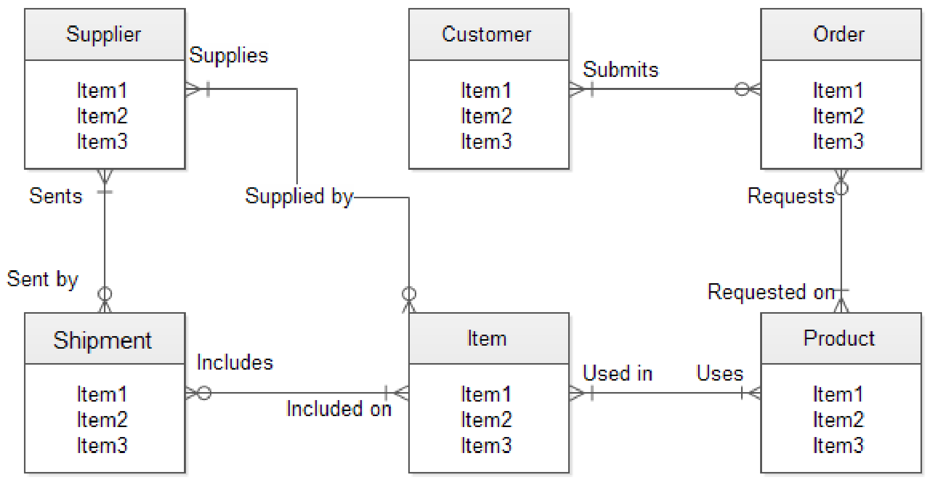
Purpose

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%E2%80%93relationship_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.

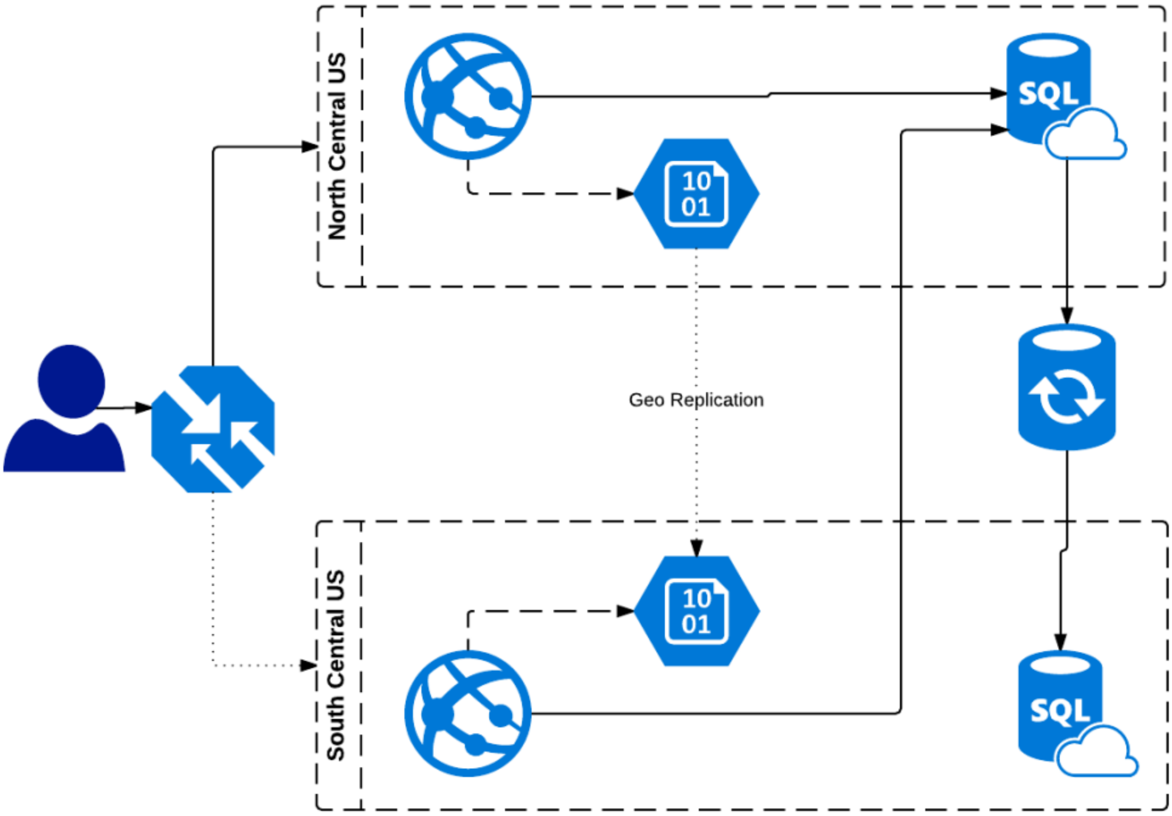
Purpose

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

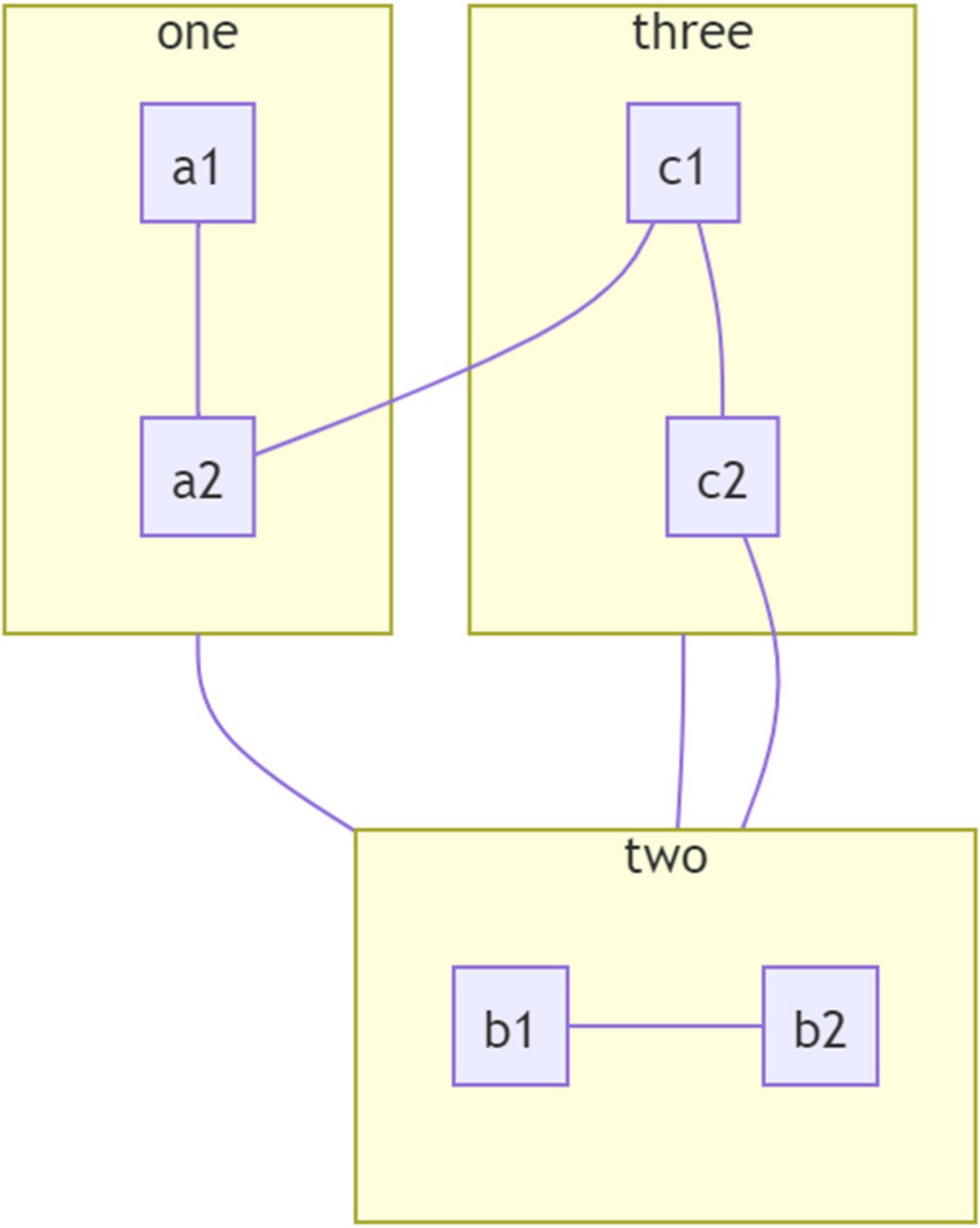
Steps to find the Visio

1. Open Visio. Select New and then type UML deployment.
2. Select the blank template and select Create.
3. Add you 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

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.

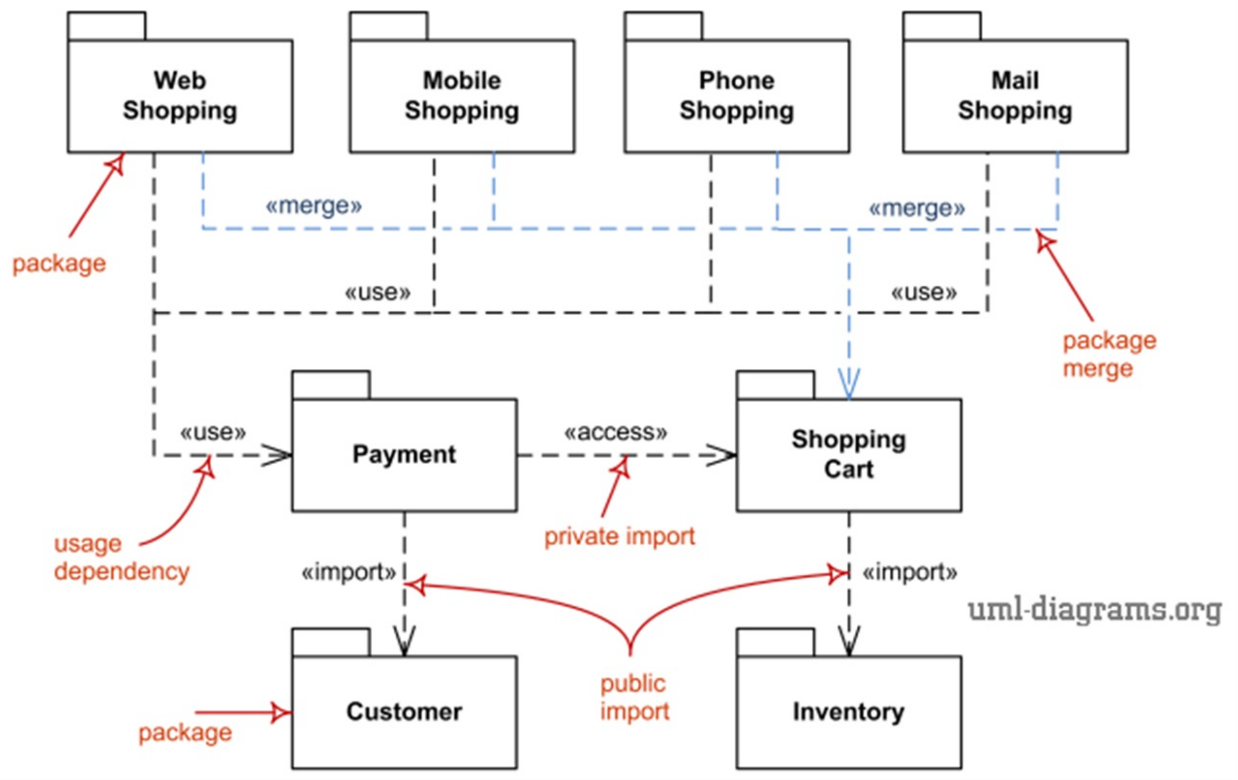
Package diagrams can be useful to do the following:

- 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
- **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.

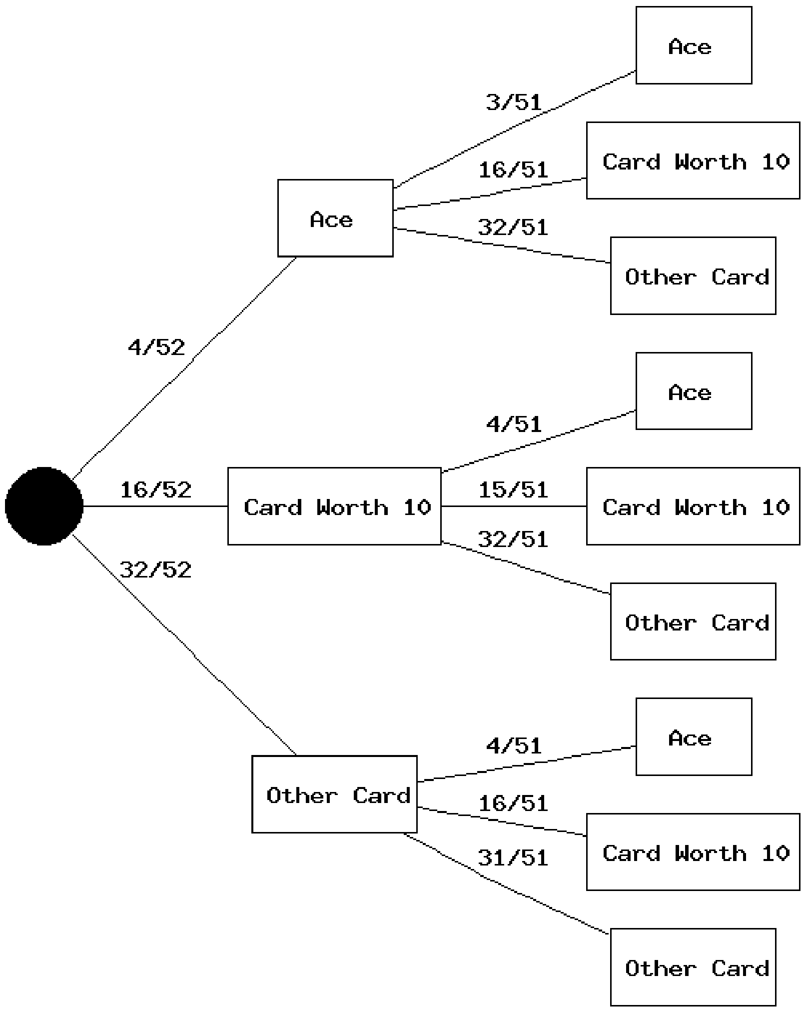
Purpose

A tree diagram is simply a way of representing a sequence of events or the hierarchy of related object.

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

Venn diagram

Describes

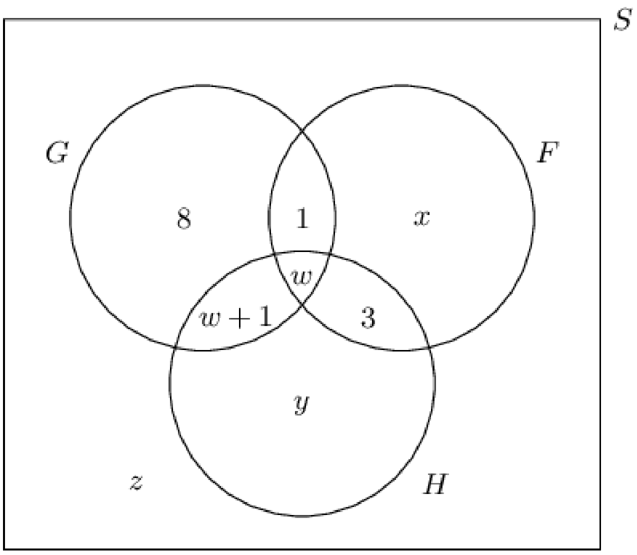
Purpose

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 overlap shows 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

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.

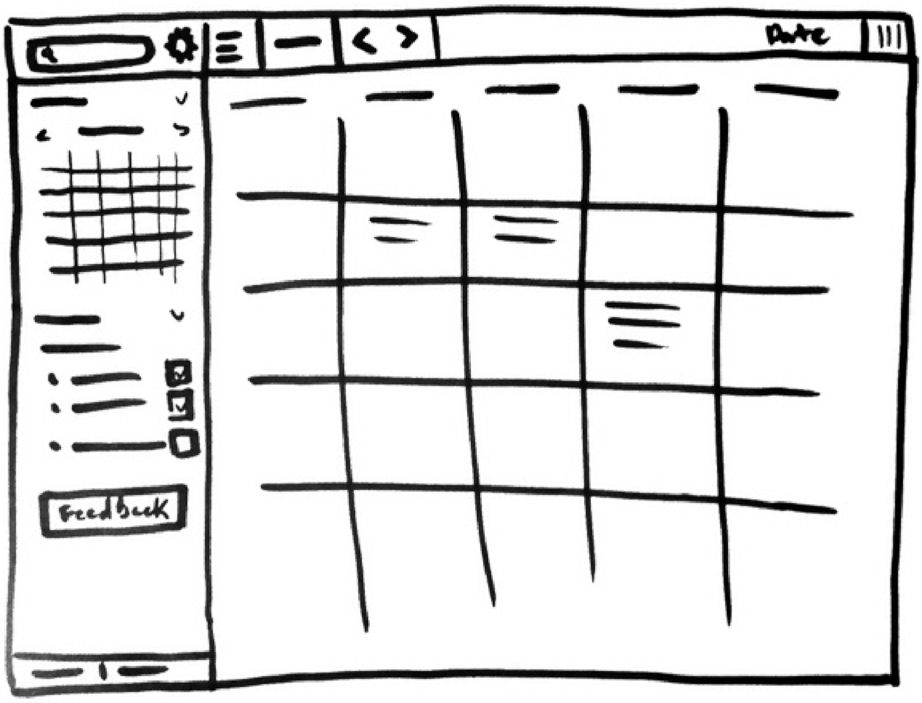
Purpose

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 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

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

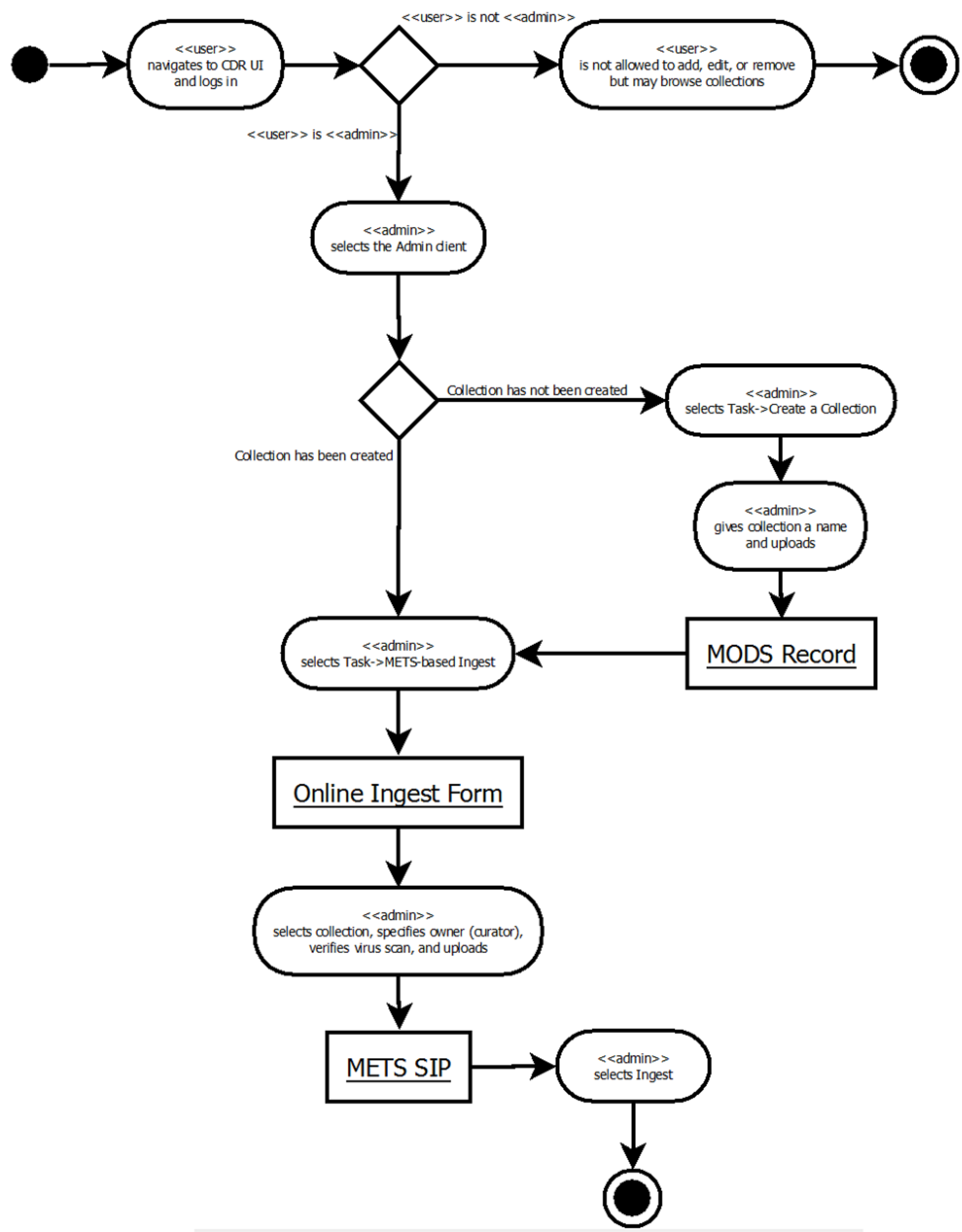
An activity diagram shows objects and choices representing the workflow of stepwise activities. This type of diagram is used to show an activity process from start to finish. This type of diagram displays the relationship between major steps in a process, it does not show minor details.

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
- **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

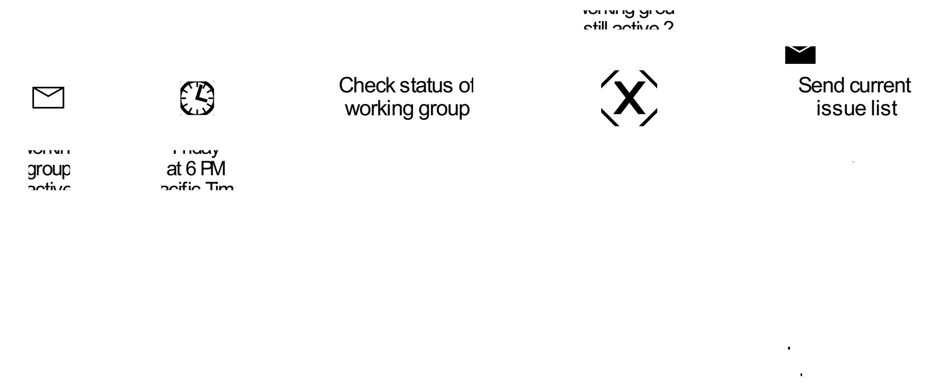
Use a BPMN diagram to represent a business process using the BPMN specification.

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.

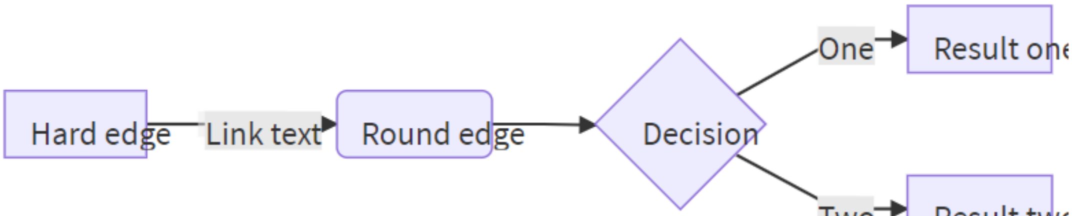
   The BPMN specification calls for three types of connectors: Sequence Flow, Message Flow, and Association. Connectors you add by using Auto-Connect or the Connector tool are Sequence Flow connectors by default. You can change the type by right-clicking the connector and then clicking the type you want 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

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.

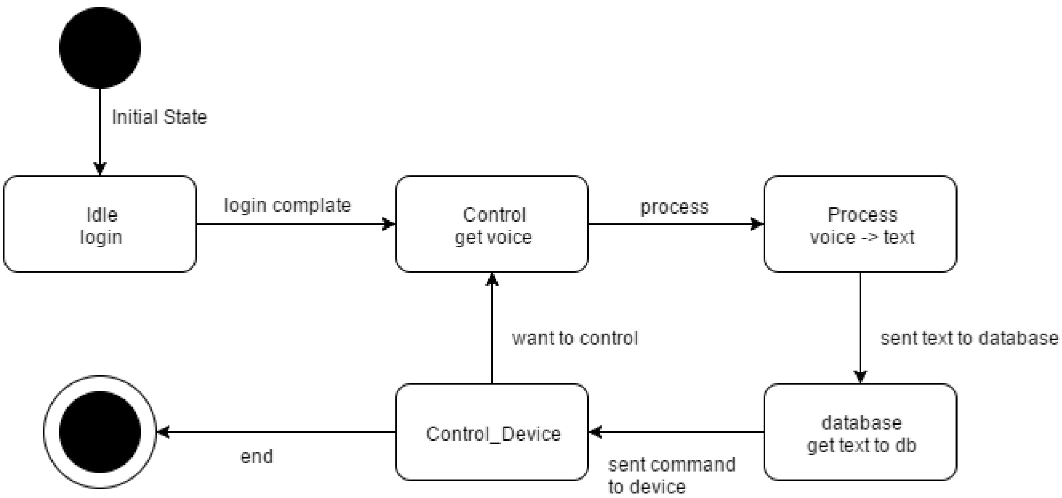
Purpose

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.

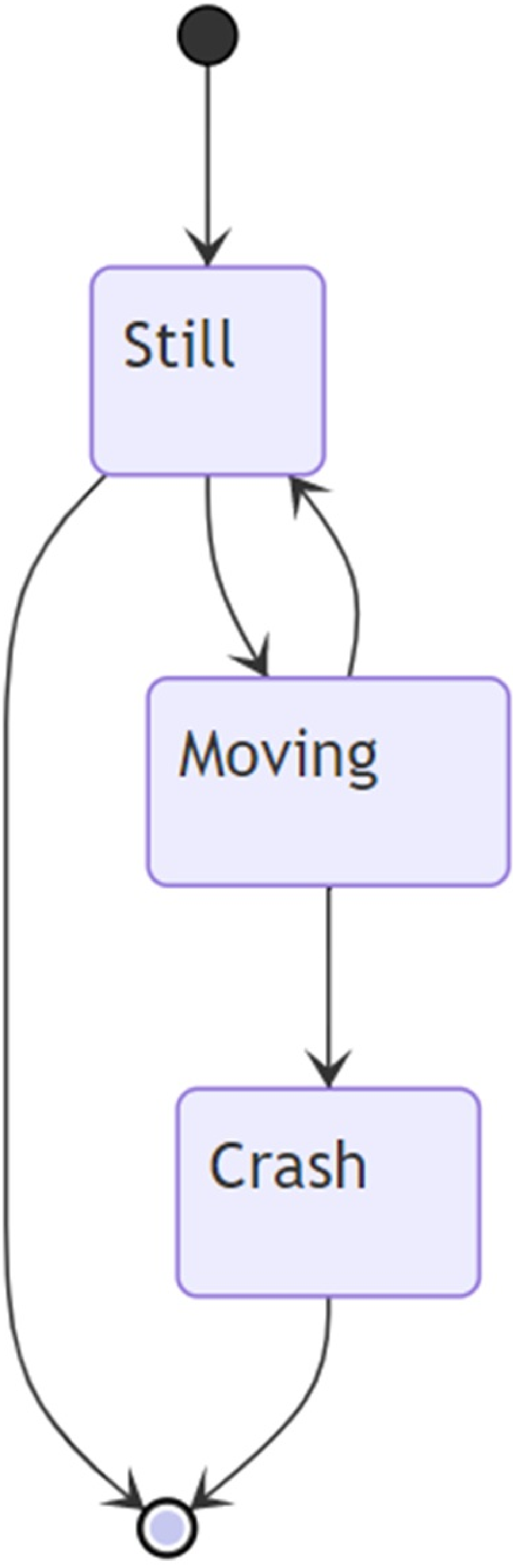
Steps to find the Visio

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
- **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.

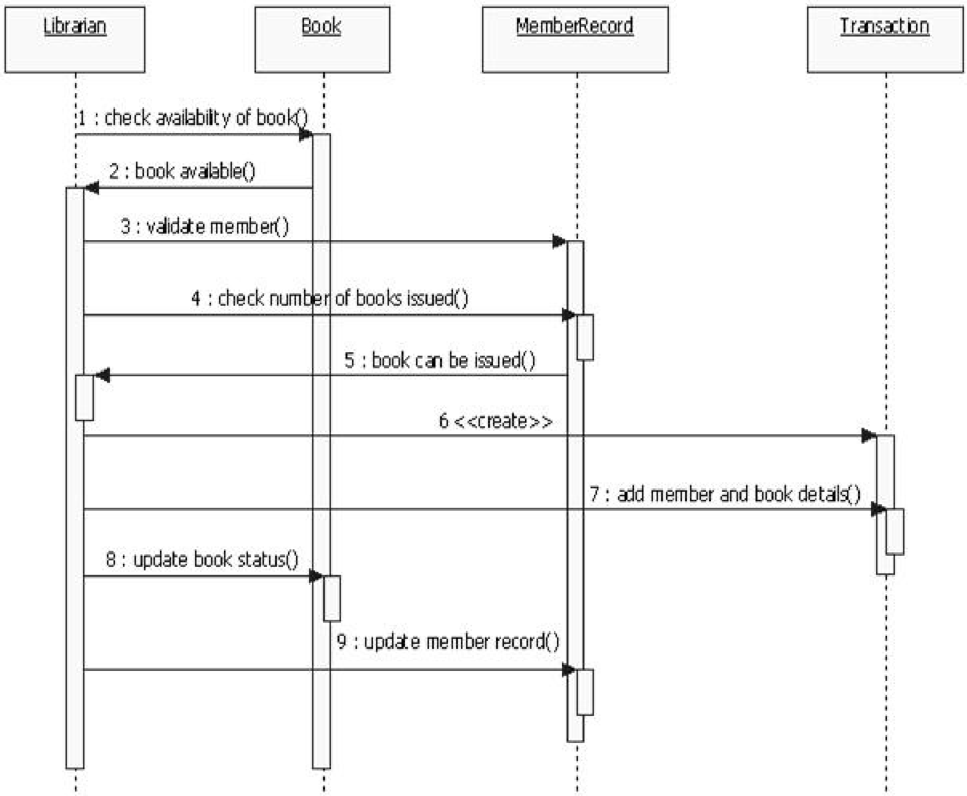
Purpose

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.

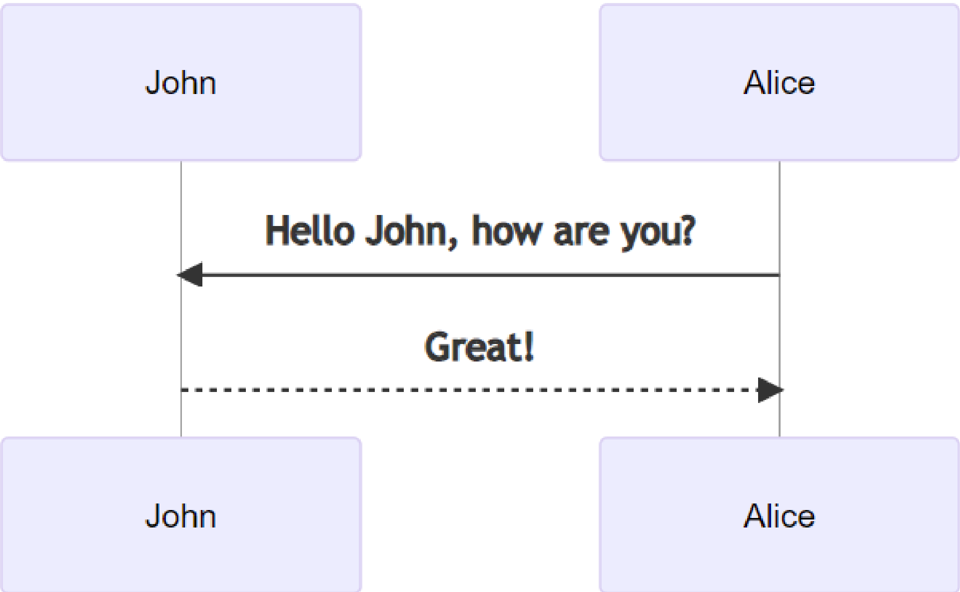
Steps to find the Visio template

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
- **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 from 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

6/24/2021 • 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.

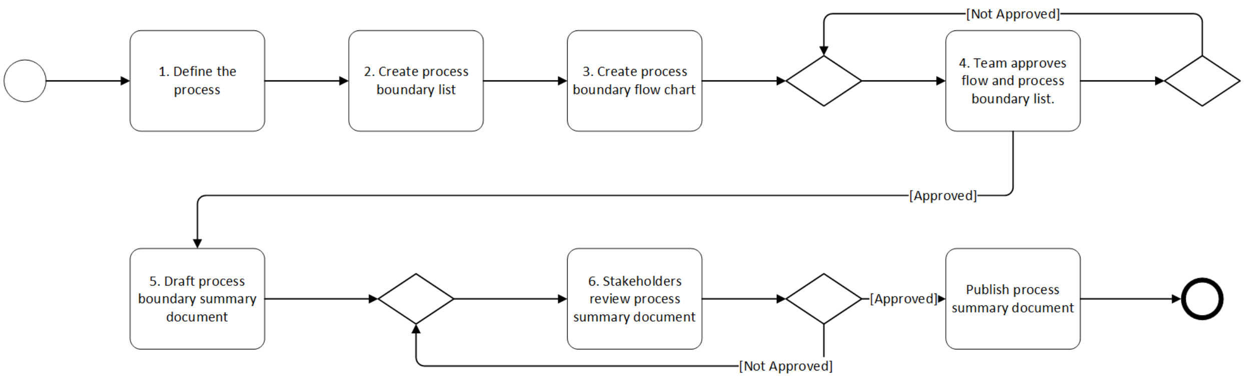
Overview

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 the material, equipment, information, people, money, or environmental conditions required to perform the process.

ITEM	DEFINITION
Input:	The materials, equipment, people, money, or environmental conditions that are required to carry out the process.
Output:	The product or service that is created by the process.
Problem/goal:	What problem or goal do you hope to address by analyzing the process?
Requirements:	What the end user of the process needs, wants and expects of the output.
Process Owner:	Name
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:	Role and Step at the beginning and end of the process. Where does it start and where does it end.
Metric:	How is the performance of the process measured (metric)?

Process team members

NAME	ROLE
Name	Title
Name	Title
Name	Title

Next steps

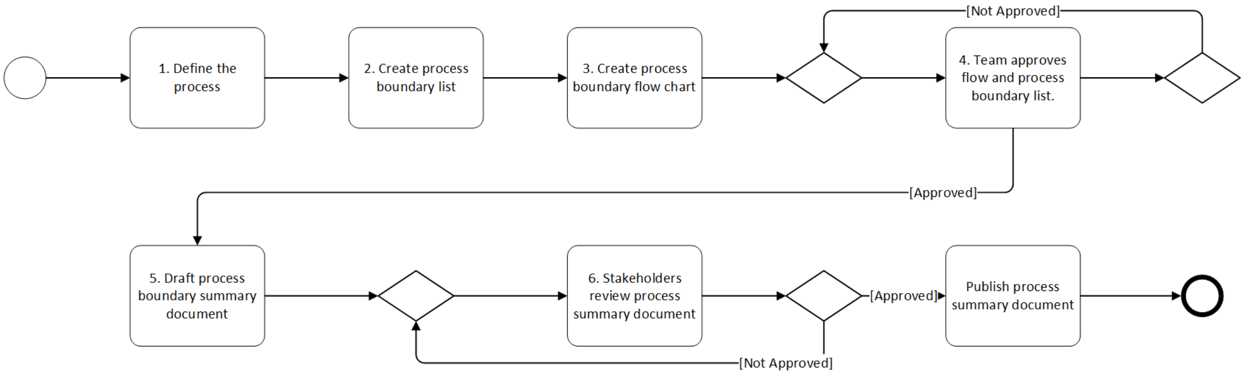
Process mapping overview.

Process map

6/24/2021 • 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 diagraming 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`:

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.dot.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

6/28/2021 • 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

7/7/2021 • 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 (because of 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, like with CodeFlow or other review processes?
- 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

7/19/2021 • 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, and so on, 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 requirement 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 that have runtimes that users may consider "unreasonable" for a sample. Keep in mind that the scripts provided are often used as cookbooks or other modified starting points. If your script takes a 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:	az eventhubs	You're writing about a command group and not referring to a single reference or command.
Reference:	az eventhubs namespace	You're writing about a reference and not referring to a single command.
Reference command:	az eventhubs namespace create	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
az group create	Creates a resource group in which all resources are stored.
az network vnet create	Creates an Azure virtual network.
az network vnet subnet create	Creates a subnet for an Azure virtual network.
az network vnet subnet show	Returns a subnet for an Azure virtual network.
az cosmosdb create	Creates an Azure Cosmos DB account.
az group delete	Deletes a resource group including all nested resources.

Syntax formatting

Use the following short-list of applied guidelines that are designed to coincide with PowreShell 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 autogenerate 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

7/19/2021 • 4 minutes to read

The following checklist helps ensure scripts and samples that use Azure PowerShell are consistent. For more information on submitting standalone scripts, 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/requires-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-azurermvm) 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. A cmdlet can have multiple ways to provide required arguments. When it does, 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 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 `azurepowershell-interactive` to add the Try It button to the snippets:

```
```azurepowershell-interactive
Get-Command -Verb Get -Noun AzVM* -Module Az.Compute
```
```

When NOT to use Try It

Don't use `azurepowershell-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 `azurepowershell` 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

6/18/2021 • 2 minutes to read

Technical focused reviews are no longer supported for C&L. The MS technical 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 technical principles.

The five writing principles

7/7/2021 • 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

7/7/2021 • 2 minutes to read

WRITING PRINCIPLES	CONSIDERATIONS	INTENT
Use everyday words	• Word choice is an important part of voice. • Be less formal, not less technical. • Use natural language. Don't use cute or casual language. • Imagine you're talking to a peer. • Use technical words when they're the right words. • Use words that your audience uses. • Explain new technical concepts using real-life examples.	• Everyday words express the intent the way a reader does • Use keywords in the title and description • Use keywords in the section headers
Write concisely	• Use everyday words. • Use locators (>) for "select, select, select" instructions. • Don't repeat words that are in the UI. • Read the article out loud. Do you stumble over some words or sentences? Are you out of breath? If so, simplify. • Consider using art. • Delete words that aren't essential.	• It's easier to be concise when you're focused on one thing • Use intent data to suggest what you can delete or move elsewhere
Easy to scan	• Write concisely. • Consider using art. • Use tables, lists, and bolding. • Put your section headers to work. • Take the birds'-eye view. • Put the most important thing first.	• The article's main intent is impossible to miss • The title and description tell me I'm in the right place • The section headers lead me through the article • There's no unnecessary information getting in my way • The article is organized so I can find my answer

WRITING PRINCIPLES	CONSIDERATIONS	INTENT
Show empathy	- Be a partner, not a judge. - Talk like you're helping a coworker. - Don't imply the problem is their fault. - If the situation is genuinely frustrating, acknowledge it. - You don't have to be their buddy, but don't be a jerk. - Leave disclaimers to the lawyers	- The tone of the article supports the customer - The article addresses the customer's point of view and what matters to them

Everyday word list

7/7/2021 • 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	Don't use. For links to web pages, use something like `For more information, see [title of page as link](link.md)` .
Click on	Select **UX element**.
Click the X tab	Select **X**.
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	`More about [specific topic](link.md)`
navigate	go
obtain	get
operation	action, task
patch	fix
perform	do
provision, provisioning	set up, setting up
purchase	buy

FORMAL	EVERYDAY
repair	fix
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

7/9/2021 • 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

6/24/2021 • 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 continous 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.

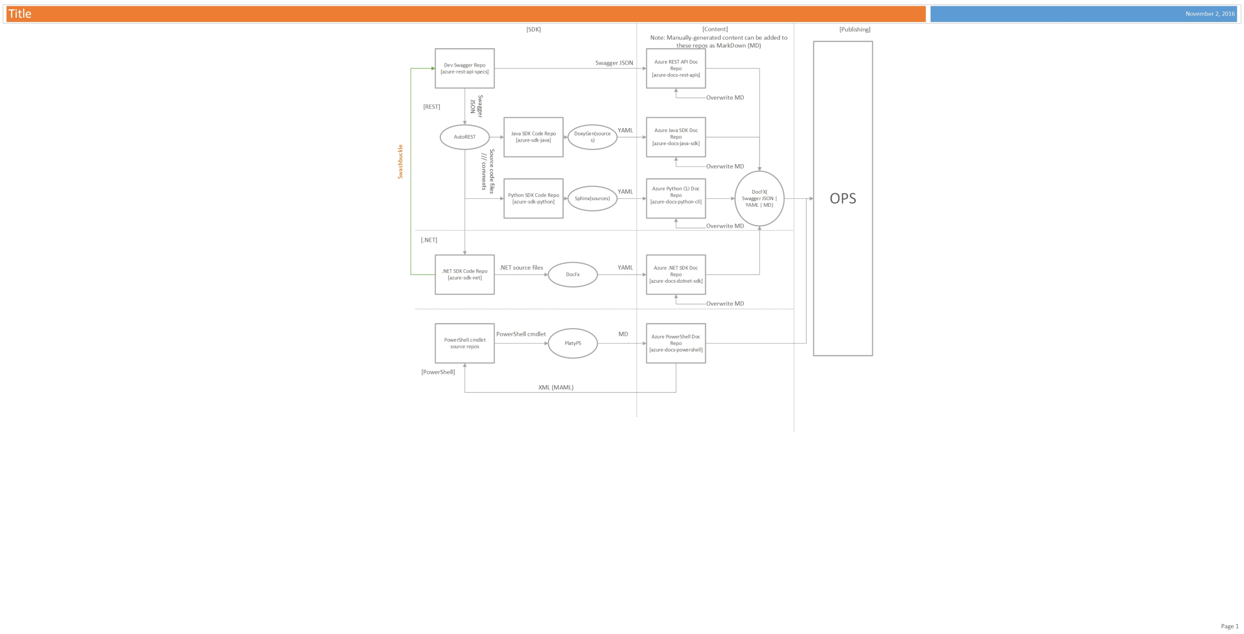
Developers

- 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



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 continous 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.

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 cas 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` and named like `azure-docs-<model>-<lang-or-framework>-<some optional discriminant tag>`
- 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 you 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 referecne
- **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

7/19/2021 • 10 minutes to read

In this article, we address common questions about contributing to and understanding reference content on https://docs.microsoft.com/. For detailed information about publishing a new API reference, see API Documentation Onboarding.

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. From there, the onboarding team and Docs Dev Experience PM team will respond.

One exception to this process is Azure PowerShell reference content. 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 ticket 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 isn't 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'll help you decide where your documentation belongs.

Repo ownership and administration

Do I need a separate repository to host my code reference?

Not always. For an Azure product, we'll build your content into existing repos. If your project isn't an Azure project, we'll create a new repo for you. When applicable, your new repo might be created in the GitHub organization for that language if there's one (for example, https://www.github.com/dotnet).

Product stakeholders and the onboarding team decide the appropriate action when they intake 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. Then, we hand the repo over to you, and you're expected to manage your own ongoing builds, updates, and so on. 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're expected to understand the publication process and are responsible for all builds.

Why are some reference docs editable and not others?

The decision is up to the product team. Some product teams lock down their docs, others don't, and some teams overlook that setting altogether!

For content that's autogenerated from an open-source repo, we can set up a 1:1 mapping. The setup will make it so that any updates are reflected in the docs. Let's say a customer submits a PR against your content and you accept the change. Then, you must manually update the source code.

If you host your code in a private repo, or the package is configured some other way that prevents a 1:1 mapping, you can't 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, and 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	Java
C#	C#
JS/TS	JavaScript

LANGUAGE	UNIFIED TOC LOCATION
Python	Python

If your reference content appears in the **Other** category at *the bottom of the TOC*, you've 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, and so on)?

Main reference repos for Azure build 4-5 times a week. If you make a change in the SDK repos/source code, that change won't go live automatically. Instead, it goes live with the next build. To publish your content sooner, request an out-of-band publish via aka.ms/publish-on-docs or the Docs Support Teams channel

Teams with self-managed repos set their own schedule and which tasks are automated versus manual. This flexibility can cause a problem for product team members who don't know exactly when their changes will go live. Be sure to train your team and communicate your builds.

For information or training on how to set up 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 won't go live until your release date. You'll have this time so that 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's possible to build preview packages. We can publish to monikers that aren't released so you can review your content before it's live. This process is for package-based systems. If you're .NET and don't have a NuGet package, there's 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'll explain these options and restrictions as we handle your scenario.

My reference content isn't rendering correctly on the review or main site. What do I do?

You can post to the Docs Support Teams channel for input on your backend content. If you can't 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 versus independently?

APIs for Azure services most likely will be added to existing Azure-specific guides. If the API isn't for an Azure service, it might be added as a new node in an existing guide. For example, the existing guide may be the .NET guide. In this case, your source files would land in a new folder of the existing repo. Or, your API will live as a guide within your 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?

Don't add conceptual files to your reference repository in most cases. Instead, you'll host all conceptual content in the same repository as your product documentation. If you don't have associated product documentation, we'll create a separate repo to host your conceptual content.

We host this content separately for better content governance. This setup helps 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). This way the content appears 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 isn't specific to a product or service, so there's 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:
   - Multi-cloud use of the Azure SDK for Python
   - Overview of Azure PowerShell
   - Service Bus Java libraries (Overview)

When an API or SDK integrates with a Microsoft product or service, a full conceptual guide may be available for reference. Product teams should add conceptual articles like tutorials, how-tos, and so on, to that product's guide. Some examples of conceptual articles include:

- 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 doesn't involve the SDK.

- Machine Learning service crosslinks: ML.NET mention in a Resources article 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 tell what repo it's in because it's not open for edits. 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'll 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'll 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'll 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 sieze 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

7/19/2021 • 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 artifacts

There are several documentation artifacts for an Azure client library: the README (used in multiple places), a library index and overview, its API reference, a Quickstart article, and conceptual documentation.

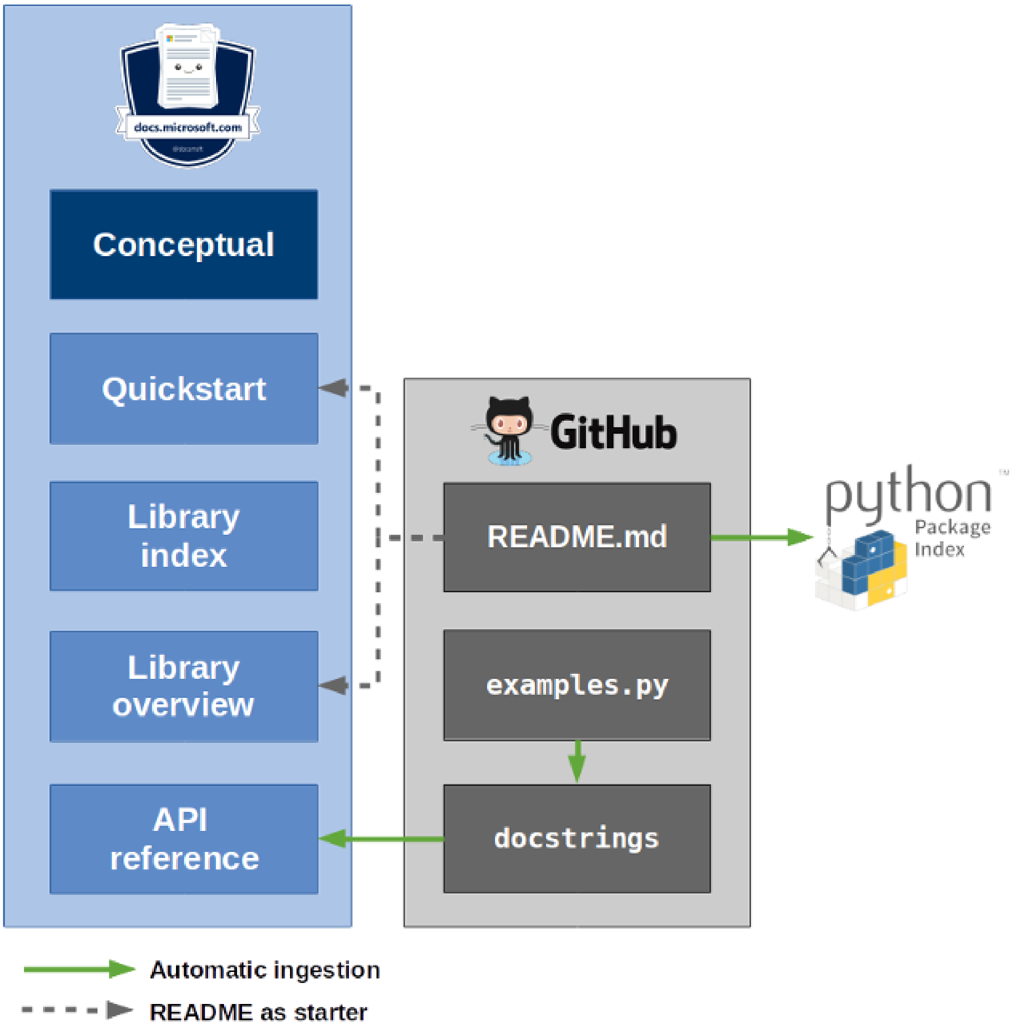
TYPE	DESCRIPTION	OWNER(S)	EXAMPLE(S)	TEMPLATE
README	`README.md` in the root of the library's repository or its directory within a repository, used for: • Library README (GitHub) • Package distro page (NuGet, PyPi, npm, Maven) • Library overview • Starter for Quickstart	Azure SDK team, product team	Cosmos DB client library for Python README	README template

TYPE	DESCRIPTION	OWNER(S)	EXAMPLE(S)	TEMPLATE
API reference	API reference documentation for a library's types and their members. Autogenerated 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	• Service Bus Python reference • `examples.py`	N/A
Library index	The library index is the landing page for a service's library reference on https://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	Library index template
Library overview	The library overview is the landing page for a single library's reference on docs.microsoft.com. The library's README is used as the landing page.	Content developer, product PM	TODO	README template
Quickstart	The Quickstart gets a developer up and running with the client library.	Content developer, product PM	TODO	Client library Quickstart
Conceptual	Long-form technical documentation for the service. Concepts, quickstarts, tutorials, and how-tos.	Content developer, product PM, public	Cosmos DB documentation	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 do 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. **Do not** 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: • **Install the package** (required) • **Prerequisites** (required) • **Authenticate the client** (if applicable)
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. Show how to understand what they mean 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 https://docs.microsoft.com/ and IntelliSense-like hover help in IDEs.

Follow the documentation guidance for the language in which your library is written. At a minimum, include the following information:

- **All types, their members, and parameters are documented** to 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 seen 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 or members in the reference documentation. The links should go from the README and conceptual docs, and back.
- 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

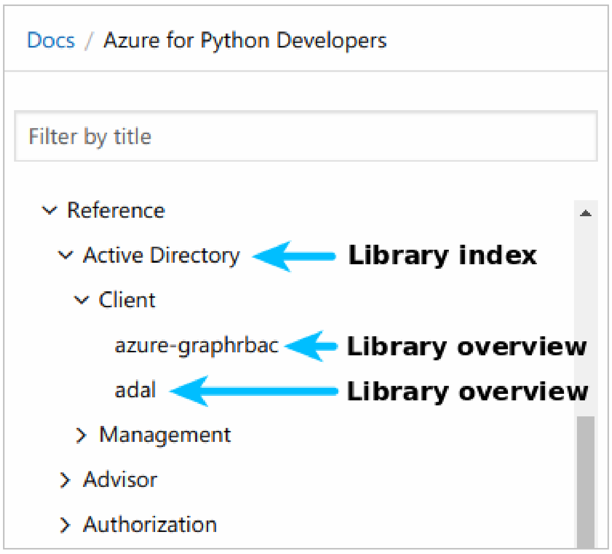
See the Service Bus client library for Python API reference for an example of library reference documentation that follows this guidance.

Library index

The library index is the landing page for Azure service's client libraries. It lists the available libraries in a language specific to the service. The index also provides a short description 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. The library should help a developer get up and running quickly with the library. It includes package installation, client authentication, and code snippets. The information in the overview will help with most operations developers need to do 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. The Quickstart also helps the developer complete key coding tasks in their development environment.

The intent of a library Quickstart *isn't* to introduce a service and its features. 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 technical content appears as articles on https://docs.microsoft.com/. For example, Azure's conceptual documentation is 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. The section includes a matrix of language-native documentation tools that are used in reference generation. For example, Sphinx for Python, TypeDoc for TypeScript, and so on:

Supported platforms

Documenting your code

Language-specific guidance for in-code documentation (triple-slash comments, Python docstrings, and so on.):

- .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

7/8/2021 • 4 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
- Use style guides for content on docs.microsoft.com

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:	az eventhubs	You're writing about a command group and not referring to a single reference or command.
Reference:	az eventhubs namespace	You're writing about a reference and not referring to a single command.
Reference command:	az eventhubs namespace create	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
az group create	Creates a resource group in which all resources are stored.
az network vnet create	Creates an Azure virtual network.
az network vnet subnet create	Creates a subnet for an Azure virtual network.
az network vnet subnet show	Returns a subnet for an Azure virtual network.
az cosmosdb create	Creates an Azure Cosmos DB account.
az group delete	Deletes a resource group including all nested resources.

Syntax formatting

Use the following short-list of applied guidelines that are designed to coincide with PowreShell 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 autogenerate 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

7/19/2021 • 2 minutes to read

The library index is the landing page for an Azure service's library reference on https://docs.microsoft.com/. It lists and describes the libraries available for the service. The page also links to the library reference landing pages. For more information, see Document your library - library index.

Index page structure

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 description, for example see Cosmos DB docs landing page.

After the title and introduction, present a table in the *Libraries* section that lists each library. In the table, list the packages available for the service and the supported library versions. Use an H3 header above each table that includes the version number. Include the package version number to help the user know if their package is up to date.

> **TIP**
>
> A "library" is generally the intersection of a service and language. Many libraries publish multiple "packages".

Use the following Markdown starter for your table:

```
| 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)    |
```

> **NOTE**
>
> The **Samples** column can be skipped if there are no dedicated samples for the packages as with libraries for resource management.

Libraries for resource management

Use the following library to work with the Azure Storage resource provider:

NUGET PACKAGE	REFERENCE
Microsoft.Azure.Management.Storage	API Reference for Microsoft.Azure.Management.Storage

Libraries for data access

The following example shows 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.Files.Shares	API Reference for Azure.Storage.Files.Shares	Samples for Azure.Storage.Files.Shares
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

7/8/2021 • 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".

See also

- Use style guides for content on docs.microsoft.com

Coverage Requirements: Introductory Content

7/1/2021 • 6 minutes to read

In this article, you'll learn about the required *introductory content* for each of the Microsoft products and services published on docs.microsoft.com. The requirements **apply to all content teams that contribute on docs.microsft.com** including:

- DevRel Content & Learning (C&L)
- Business Application Group (BAG)
- Microsoft 365 (M365)
- Product Engineering groups

What is introductory content?

By **introductory content**, we mean the content that a new user needs to get up and running on their own with a product or service offering. This content might be published as product documentation, Learn modules, and/or Patterns and Practices (for example, well-architected framework or cloud adoption framework) guidance.

It differs from Azure Product Launch Readiness (PLR) in the following ways.

AZURE PLR	INTRODUCTORY COVERAGE
Azure only	All Microsoft prodcucts/services
Compliance launch readiness requirement	Organizational standard (not a product/service ship-blocker)
Docs only	Docs, Learn, PnP

FY22 Objective Key Result

This Objective Key Result (OKR) is shared by Content teams (C&L, BAG, M365) and Product Engineering groups.

Coverage OKR: Deliver content that helps people succeed.

- **Key Result #1**: **"Introductory Content Coverage"** - Increase the percentage of Microsoft products with "getting started" content on docs.microsoft.com from X% to 100% by end of FY22.
- **Key Result #2**: **"Time to Coverage"** - Increase the percentage of Microsoft products with "getting started" content available on docs.microsoft.com within Y days of GA launch from X% to 100% by end of FY22.

Read more about this OKR.

What should I do with the report?

> **IMPORTANT**
>
> The coming report and the Coverage score is **not** to be read as a report card. Instead, they are intended to be used as a signal of coverage gaps, opportunities, and resource gaps.

The score will be used by leadership; however, the report will show you how you are tracking against the coverage goals for introductory content.

In FY22, content teams are required to review the coverage report and develop a coverage plan with their manager. For now, the report is currently available as living Excel sheet for limited products and services. Tracking data is currently limited to 69 Azure services and 22 BAG products and growing quickly.

Some individual contributors may find parts of the Coverage report useful. Work with your manager to understand your team's priorities.

> **NOTE**
>
> The content coverage score for "Introductory content" will be tracked on monthly basis and accessible through a Power BI report. The Power BI reporting is coming soon and would scale baseline to remaining products in couple of iterations over the course of next 1-2 months (Tentative ETA: FY22Q1).

Temporary Excel OKR tracker (report coming soon)

M4 =AVERAGEIF(D4:L4,"<>NA")

			Introductory and Getting Started Content (Asset or Content Types)									
			Docs	Docs	Docs	Docs	Docs	Docs	Learn	Learn	P&P	
#	Customer Solution Area (CSA)	Azure Product/Services	Landing Page	Overview	If QuickStart apply – one for Portal	If Quickstart apply - atleast one ~~SDK~~ API based	If Quickstart apply - atleast one ARM template	Tutorial (incases where Quickstart is not applicable)	Evaluate/Intro to module	Try Module	CAF /AAC /WAF	Coverage Score
1	Infrastructure & Application	API Management	1	1	1	1	1	1	0	1	1	89%
2	Infrastructure & Application	Application Gateway	1	1	1	1	1	1	0	1	1	89%
3	Security	Azure Active Directory	1	1	1	1	0	1	0	1	1	78%
4	Infrastructure & Application	Azure Advisor	1	1	1	NA	1	1	0	0	1	75%
5	Infrastructure & Application	Azure App Service	1	1	0	1	1	1	0	1	1	78%
6	Infrastructure & Application	Azure Backup	1	1	1	1	1	1	0	1	1	89%
7	Security	Azure Bastion	1	1	1	0	0	1	0	1	1	67%
8	Data & AI	Azure Bot Service	1	1	1	1	0	1	0	1	1	78%
9	Data & AI	Azure Cognitive Search	1	1	1	1	1	1	1	1	1	100%
10	Data & AI	Azure Cosmos DB	1	1	1	1	1	1	1	1	1	100%
11	Infrastructure & Application	Azure Cost Management and Billing	1	1	1	0	1	1	0	1	1	78%
12	Infrastructure & Application	Azure Data Explorer	1	1	1	NA	1	1	0	0	1	75%

Tracking your work

When you create + update content to meet our Coverage OKR, tag your Azure DevOps work items with `okr-coverage-fy22`. This tagging will enable us to keep track of work in progress.

Azure requirements

Review what you need to provide **FROM ALL THREE MODALITIES**.

Find our exception process and list at the end of this article.

Required Docs content

TOPIC TYPE	MS.TOPIC AND MS.CUSTOM VALUE	TOC	DETAILS
Landing page	landing-page or hub-page	NA	NA
Overview	overview	NA	NA
Quickstart for Azure portal	quickstart, and mode-portal (ms.custom)	NA	A quickstart for Azure portal, a QuickStart can be orientation, set up, installation or deployment

TOPIC TYPE	MS.TOPIC AND MS.CUSTOM VALUE	TOC	DETAILS
Quickstart for API	quickstart, and mode-api (ms.custom)	NA	A quickstart that is API-based (PowerShell, CLI, language SDKs such as Python etc.)
Quickstart for ARM	quickstart, and mode-arm (ms.custom)	NA	A quickstart for ARM template
Tutorials	tutorial	NA	If Quickstarts don't apply, then a tutorial

Required Learn content

At least two Learn modules determined by Learn team's owned tooling.

TOPIC TYPE	MS.TOPIC VALUE	TOC	DETAILS
'Introductory' module	NA	NA	That enables user to decide if the product meets their needs. It is typically a "Introduction to" module that includes - What is it? How it works? When to use it?
'Try' module	NA	NA	That enables a user to use product in a simple hands-on experience but real-world context. It can be a sandbox, BYOS (bring your own subscription), simulation (with hotspots for interaction), or a screen cast (no interaction). Currently, learn modules are not categorized by Evaluate and Try. Doing baseline assessment may require manual scrapping.

Required Patterns & Practices (PnP) Content

At least one introductory article (`ms.custom = internal-intro`). Below is the criteria to determine whether a specific content qualifies as introductory article:

- Does the content have an architectural diagram that aligns with the use case?
- Does the content justify to the reader the product decision based on the use case?
- Does the content encourage the reader to think about cost, reliability, perf, operations, and security with easy to understand examples?
- Does the content provide deployment assets for the entire use case so that the reader can "try out" the scenario for learning purposes?

BAG requirements

For BAG, including Dynamics 365 and Power Platform, review what you need to provide FROM ALL MODALITIES LISTED HERE.

Find our exception process and list at the end of this article.

Required Docs content

TOPIC TYPE	MS.TOPIC OR MS.CUSTOM VALUE	TOC	DETAILS
Landing/hub page	landing-page or hub-page	NA	NA
Product Overview	intro-internal (ms.custom)	NA	NA
Getting started	intro-internal (ms.custom)	NA	How to try/buy, deploy/configure, and use the “top” features that could be identified during the doc planning.

Required Learn content

At least one getting started Learn module per product at GA (determined by Learn team’s owned tooling). Ideally, at least two modules:

1. Intro to module
2. Try module Ideally, we would like to have both modules (determined by Learn team’s owned tooling). For any reporting data issue, contact Adrian Stevens and Jesse Dietrichson.

Other C+AI product requirements

These requirements apply to non-Azure and non-BAG products delivered under C+AI, such as:

- .NET
- Visual Studio
- SQL Server
- and so on

Required content: Products in this category must have the following Learn and Docs content.

- One of these docs:

TOPIC	MS.TOPIC VALUE	TOC	DETAILS
A landing or hub page	landing-page or hub-page	NA	NA

- Two or more of these docs:

TOPIC	MS.TOPIC VALUE	TOC	DETAILS
Overview or "What is..." content	overview	NA	Title or H1 or TOC title begins with "What is"
What's New topic	NA	NA	Title or H1 or TOC title begins with "What's New" or "What's Cool"
Installation topic	NA	NA	Title/H1/TOC title begins with "Install"

TOPIC	MS.TOPIC VALUE	TOC	DETAILS
Get Started topic	NA	NA	Title/H1/TOC title begins with "Get started"
Migration content	NA	NA	Title/H1/TOC title begins with "Migrate" or "Migration"
Deployment content	NA	NA	Title/H1/TOC title begins with "Deploy" or "Deployment" is captured here too
Quickstart	quickstart	NA	NA

- One or more of these modules:
  - Introductory Learn module about the product, service, or feature (determined by Learn tooling)

All of the above is required.

M365 Developer Requirements

Docs

TOPIC TYPE	MS.TOPIC VALUE	TOC	DETAILS
Landing page for product group	landing-page or hub-page	NA	NA
Quickstart per product (if applicable)	quickstart	NA	NA
Overview per product	overview	NA	NA
Tutorial (if applicable)	tutorial	NA	NA

M365 Commercial Requirements

Coming soon

Exceptions

If you think a product or service does not require separate introductory content and/or all components of introductory content as prescribed in this article, email to Shweta Agarwal. In your email, be sure to provide the business reason for the exception.

Current list of Exceptions

- Azure Data Lake Storage - Landing page doesn't apply
- Azure Data Share - API-based quickstart, Learn intro to and try modules doesn't apply
- Azure Advisor - API-based quickstart doesn't apply
- Azure Bastion - API-based quickstart doesn't apply
- Azure Data Explorer - API-based quickstart doesn't apply

- Azure Sentinel - API and ARM-based quickstarts, Learn try module doesn't apply
- Azure Security Center - API-based quickstarts, Learn try module doesn't apply
- Virtual Machine Scale Sets - Intro PnP article doesn't apply
- VPN Gateway - ARM-based quickstart doesn't apply

Reporting issues

If the OKR reporting data (here, Power BI coming soon) is incorrect for any product or service, email Alex Buck for Docs and PnP and Jesse Dietrichson for Learn so we can fix our automation.

Keep Shweta Agarwal in CC and manually fix the metadata tag (ms.topic or ms.custom) of corresponding content.

Unified Content Model

6/25/2021 • 20 minutes to read

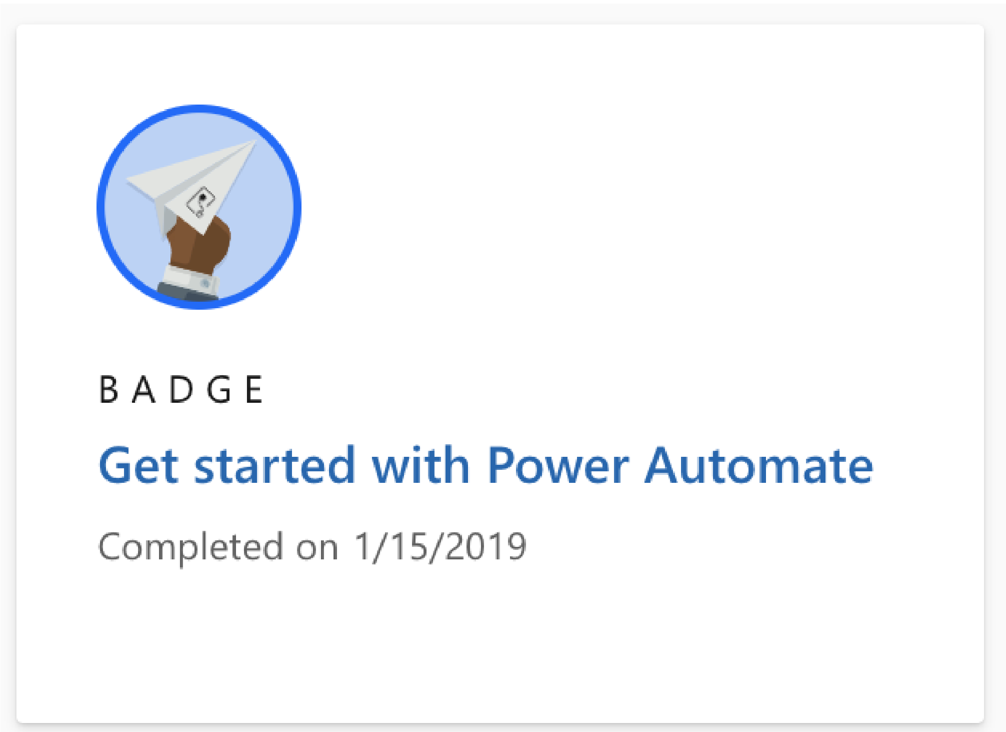
The unified content model is an effort to ensure the content on Docs.microsoft.com is structured in a way that supports the business and customers. Structured content is more sustainable, easier to reuse, more efficient to manage, and supports sophisticated technical and customer experiences. The unified 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` 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



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

Use answers to present responses to user generated questions. They should not be used outside of the context of Q&A.

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

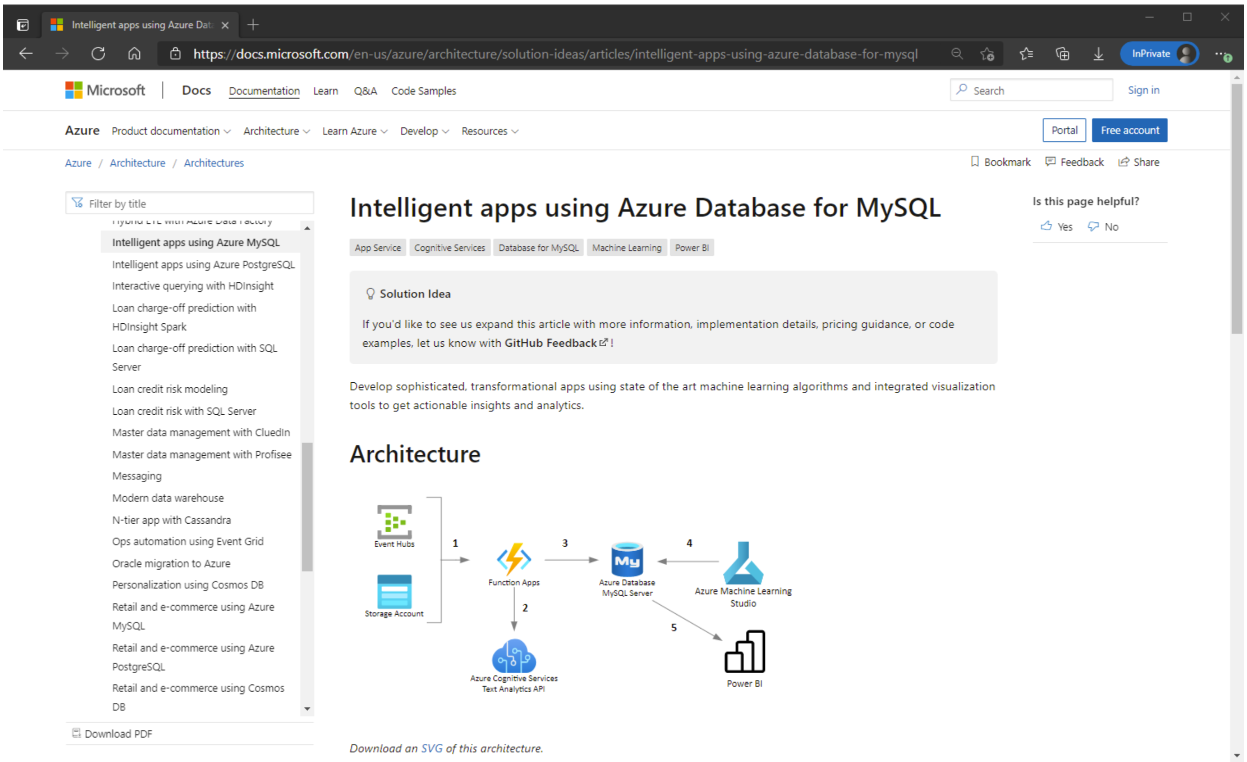
- none

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



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

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

- Certification UID
- Description
- Optional Pre-requisites
- Pre-requisites
- Title
- Status

Taxonomies used on Certifications

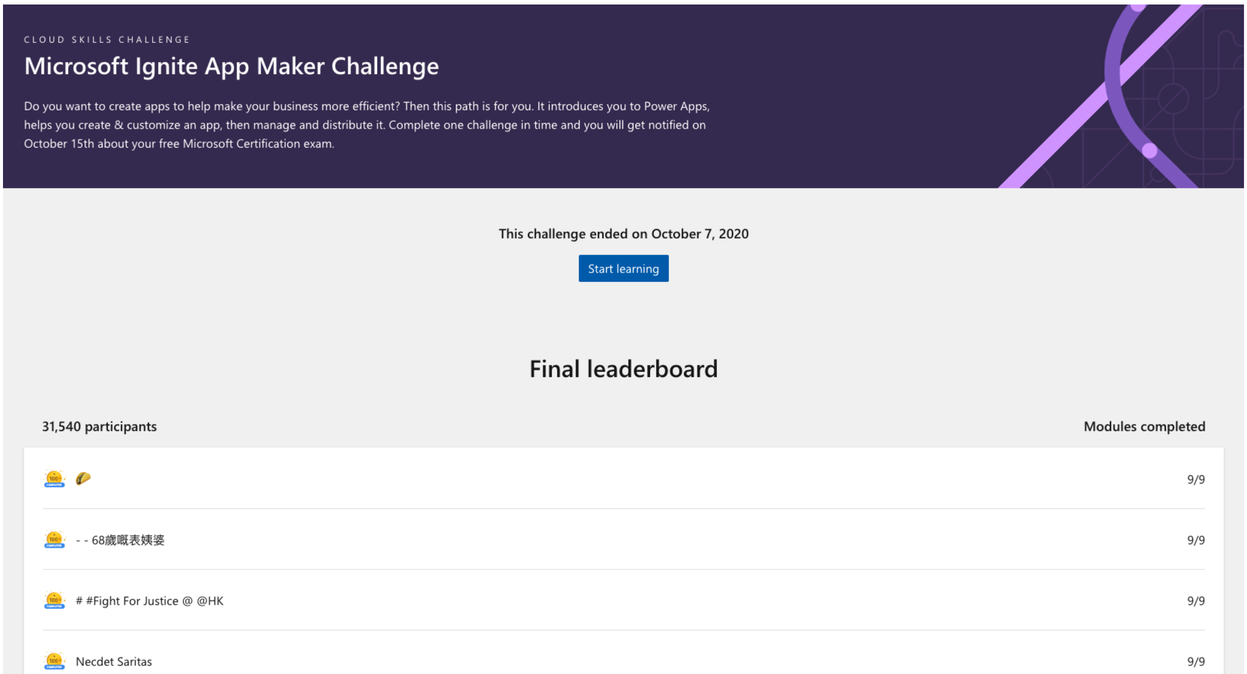
- Product
- Role
- Level
- Certification Type

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



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

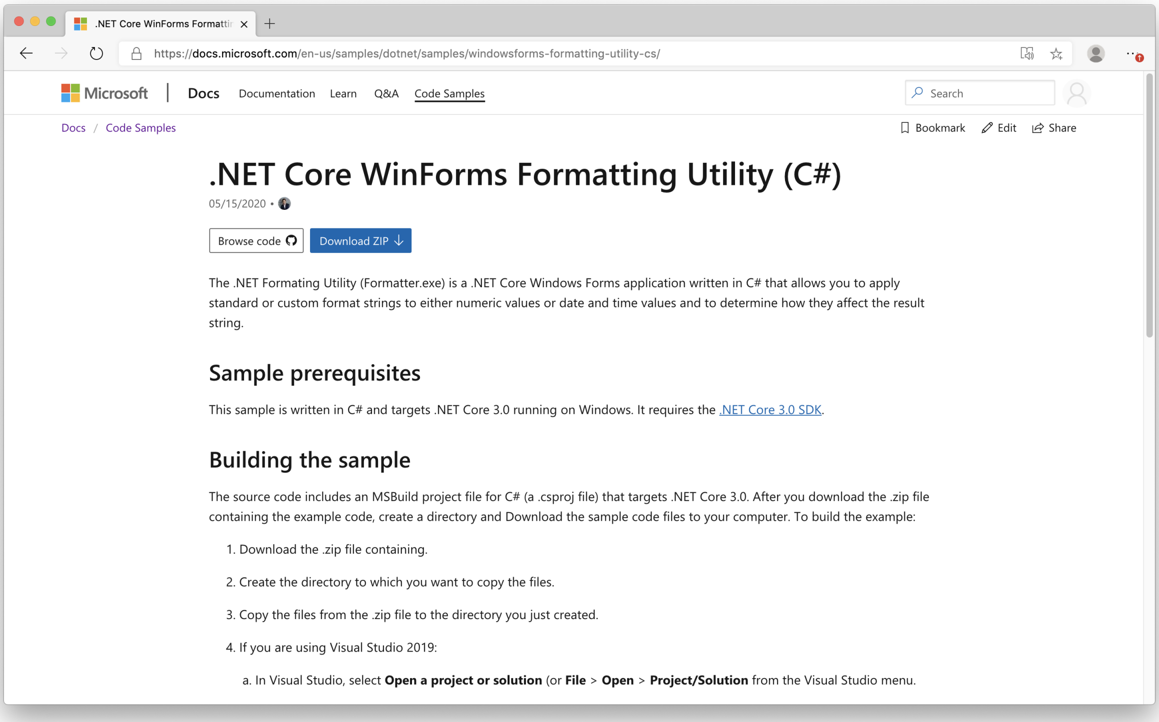
- none

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



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

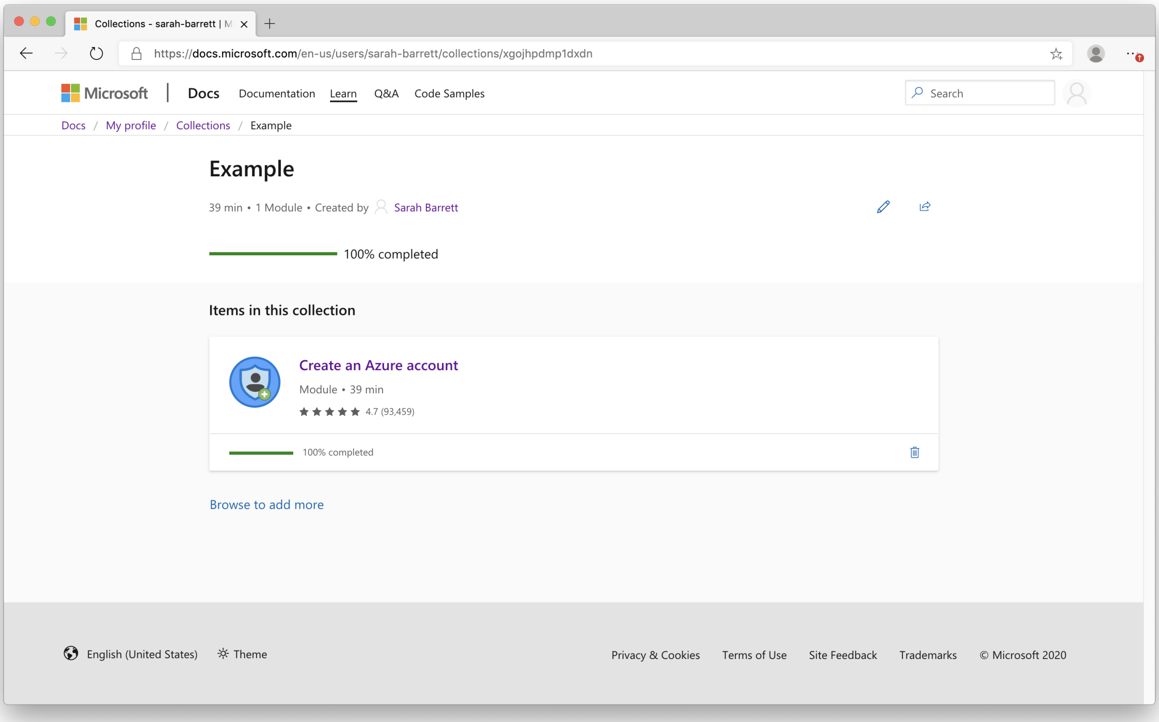
- Product
- Platform

Relationships with other content types

- Code samples in this collection
- Source code for this code sample

Collection

Status: Live



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
- Known issues 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 colleciton
- 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. Only valid for Q&A currently.

Elements of Comments

- Body

- Date
- Most recent version date
- Upvote count

Taxonomies used on Comments

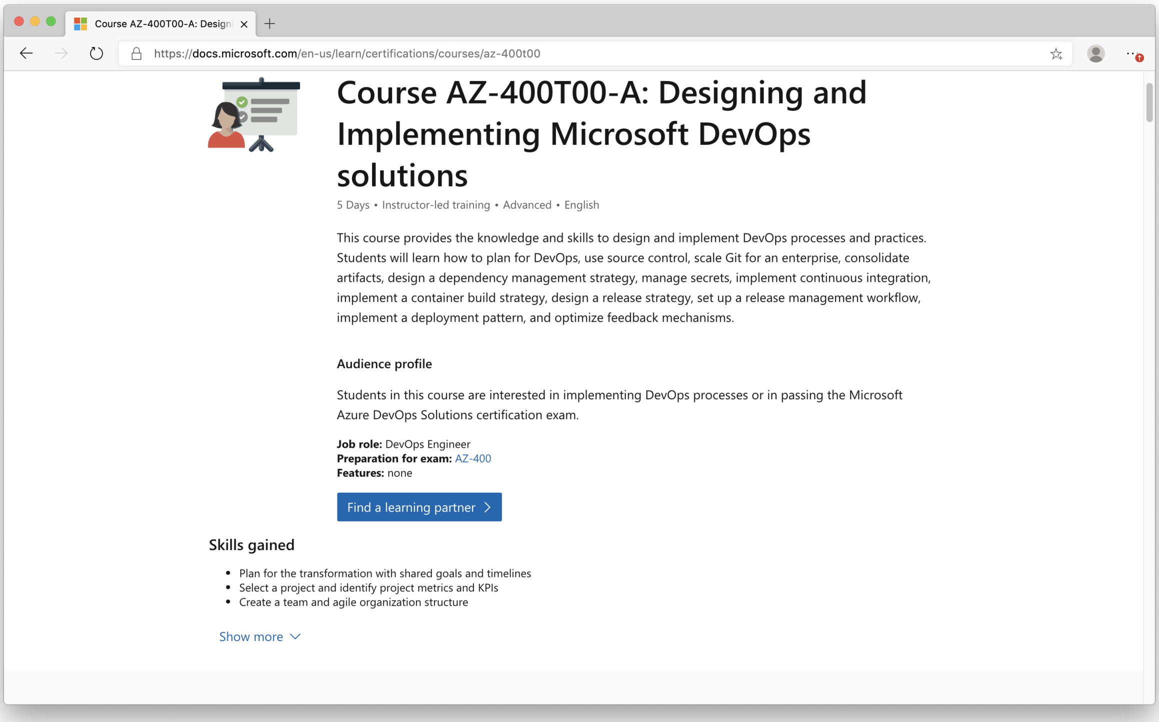
- none

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



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 colleciton

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

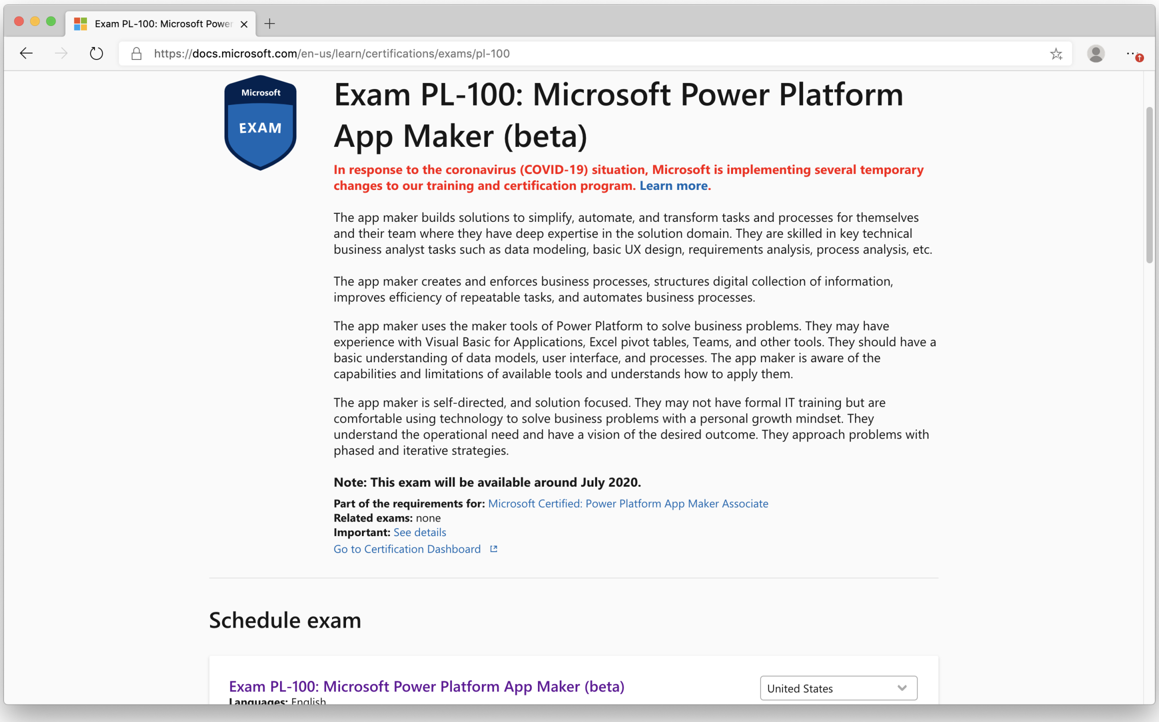
- none

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



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
- Exam that this exam session is for
- Certification this exam counts toward
- Exam this collection supports
- Exam(s) required for this certification

Exam Session

Status: Development
The specific instance of an exam (version taken at a day/time from a specific provider) that a user has scheduled/paid for or attempted. Triggered by a user scheduling an exam.

Live Example

No current example

When to use Exam Sessions

To trigger a specific instance of an exam session tied to a user. This allows us to snapshot what version of an exam a user registered for or took, and this is where we hang "appointment" data.

Elements of Exam Sessions

- Title
- Exam status
- Compatibility check link (online mode only)
- Provider
- Appointment datetime
- Confirmation code
- Mode (online or in-person)
- Testing location
- Testing URL (online only)
- Result (pass, fail, no-show)
- Link to score report

Taxonomies used on Exam Sessions

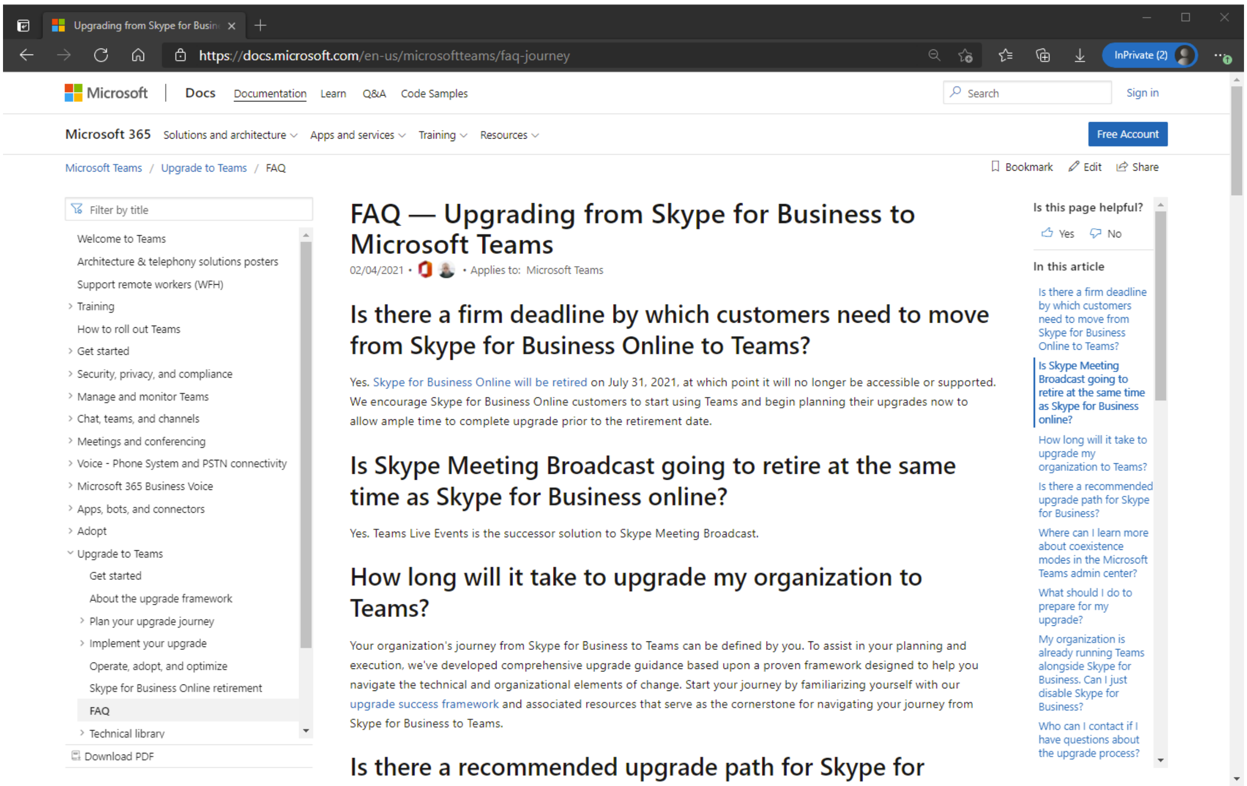
- none

Relationships with other content types

- Exam sessions for this user
- Exam that this exam session is for

FAQ

Status: Live



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

Known Issue

Status: Planned
A defect in a system that Microsoft is aware of.

Live Example

No current example

When to use Known Issues

Use known issues when you need to present a list of problems we are aware of, and optionally, are trying to fix. In-product links to explain problems should go to known issues or filtered lists of known issues.

Elements of Known Issues

- Created date
- Description
- Last modified date
- Resolved date
- Title
- Public reference ID

Taxonomies used on Known Issues

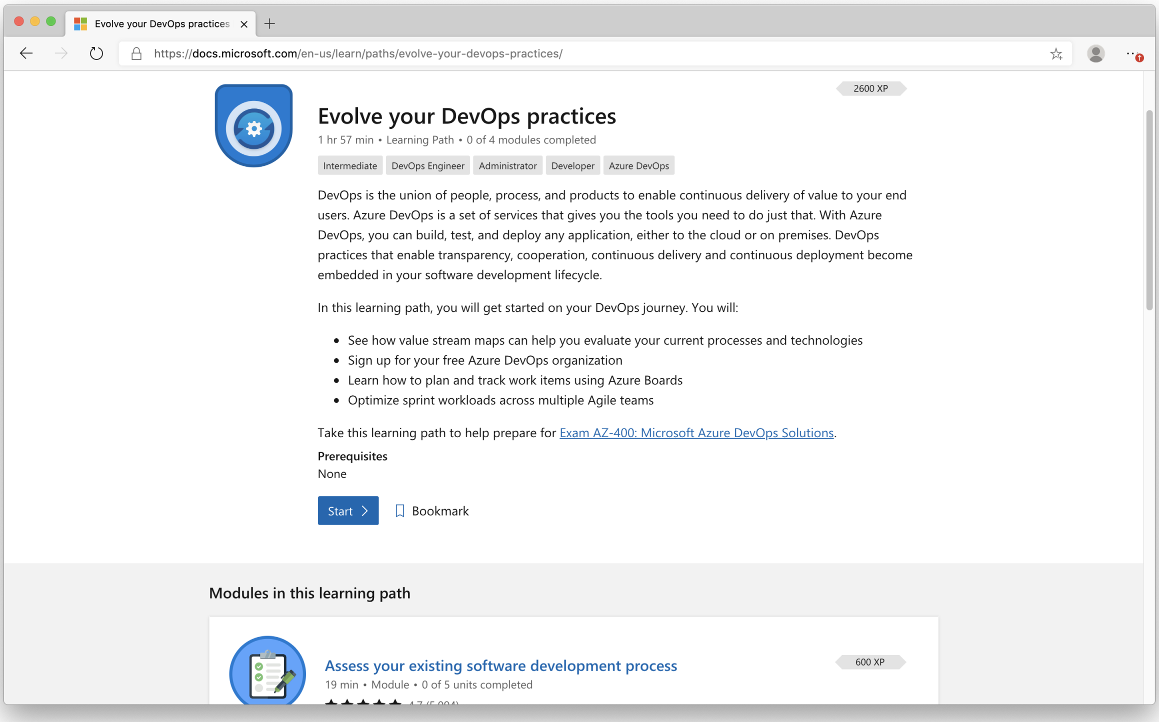
- Product
- Version
- Known Issue Status

Relationships with other content types

- Known issues in this collection

Learning Path

Status: Live



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

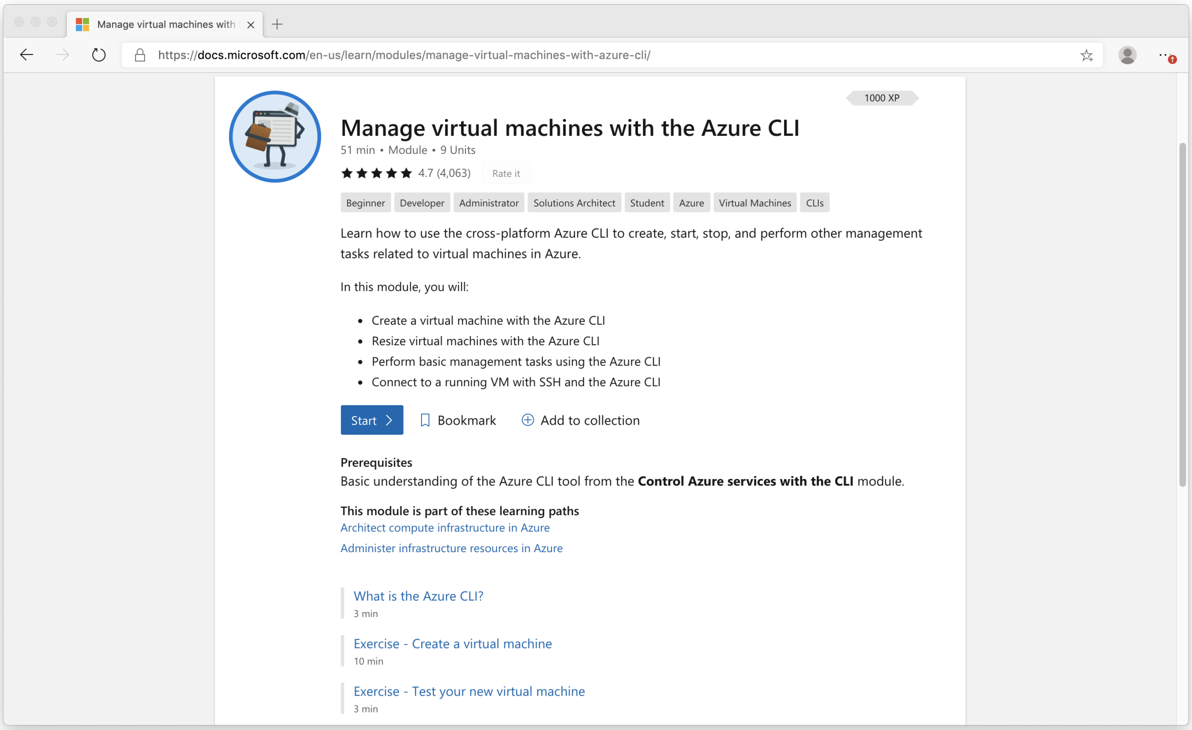
- Product
- Policy Type

Relationships with other content types

- Lifecycles in this collection

Module

Status: Live



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

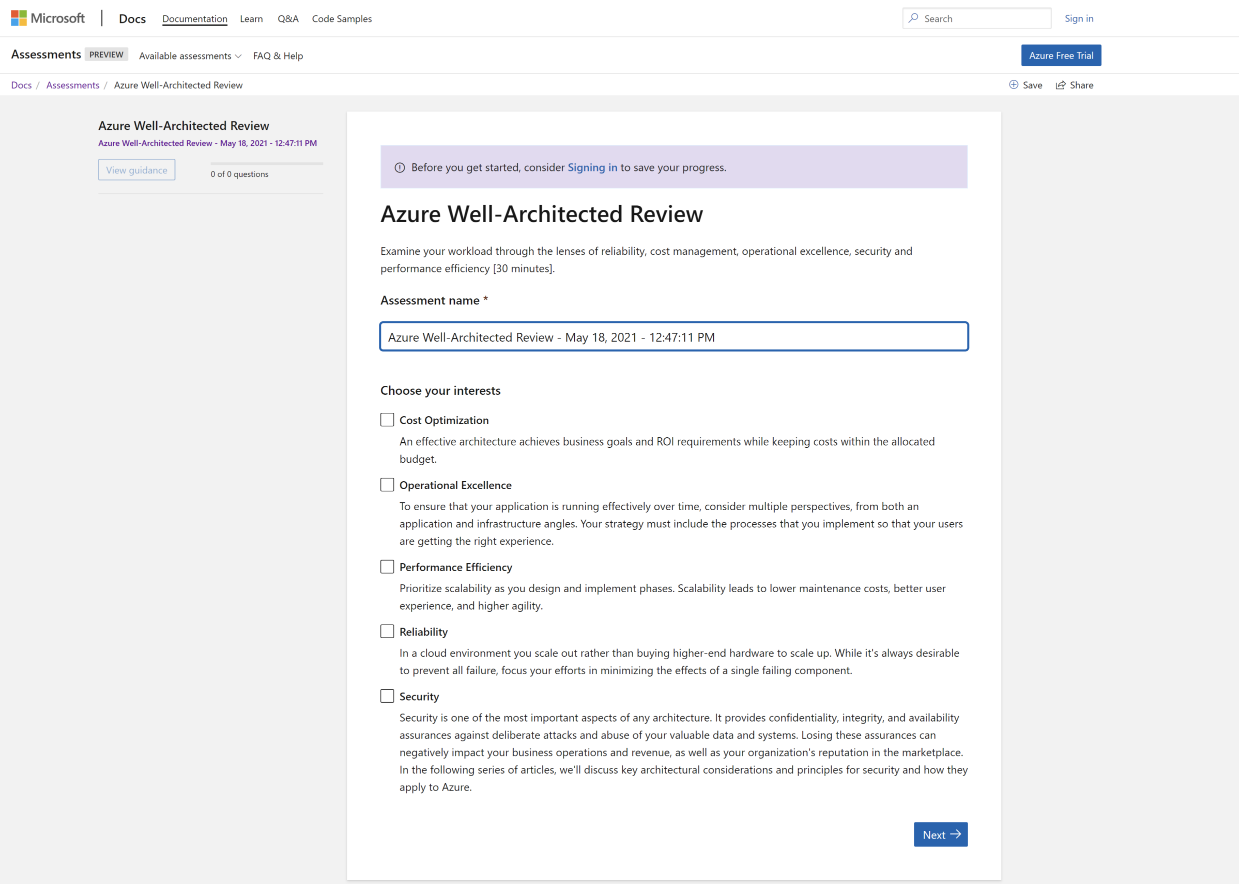
- none

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



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

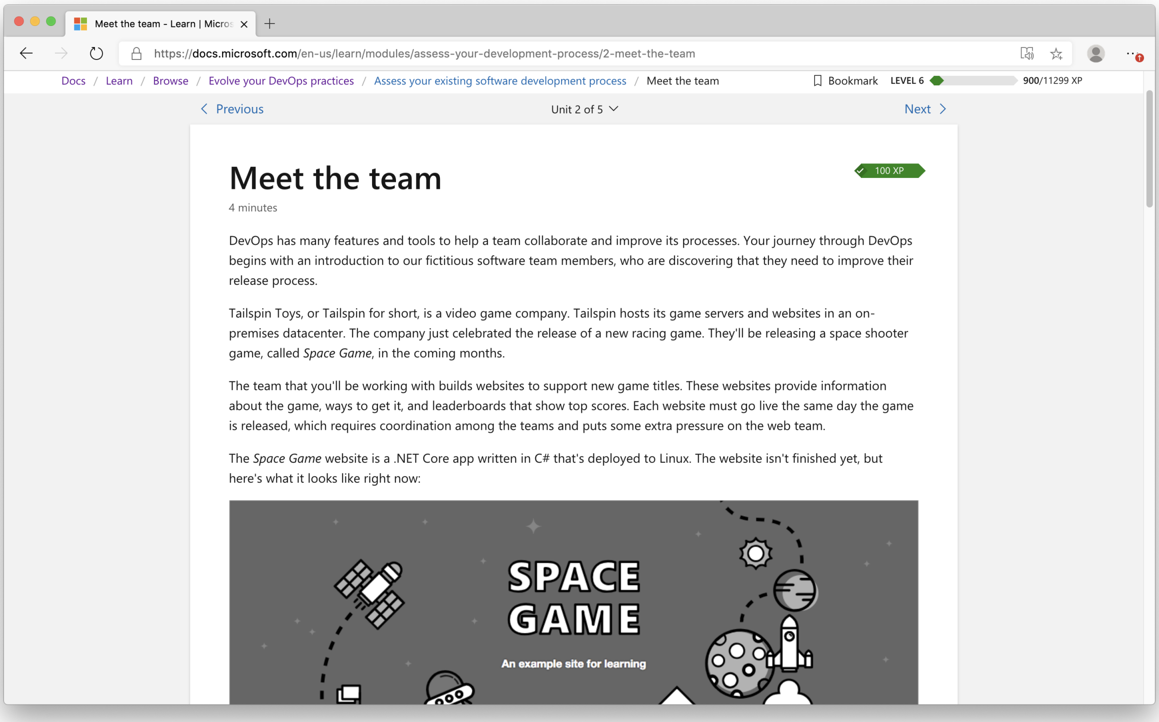
- none

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



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

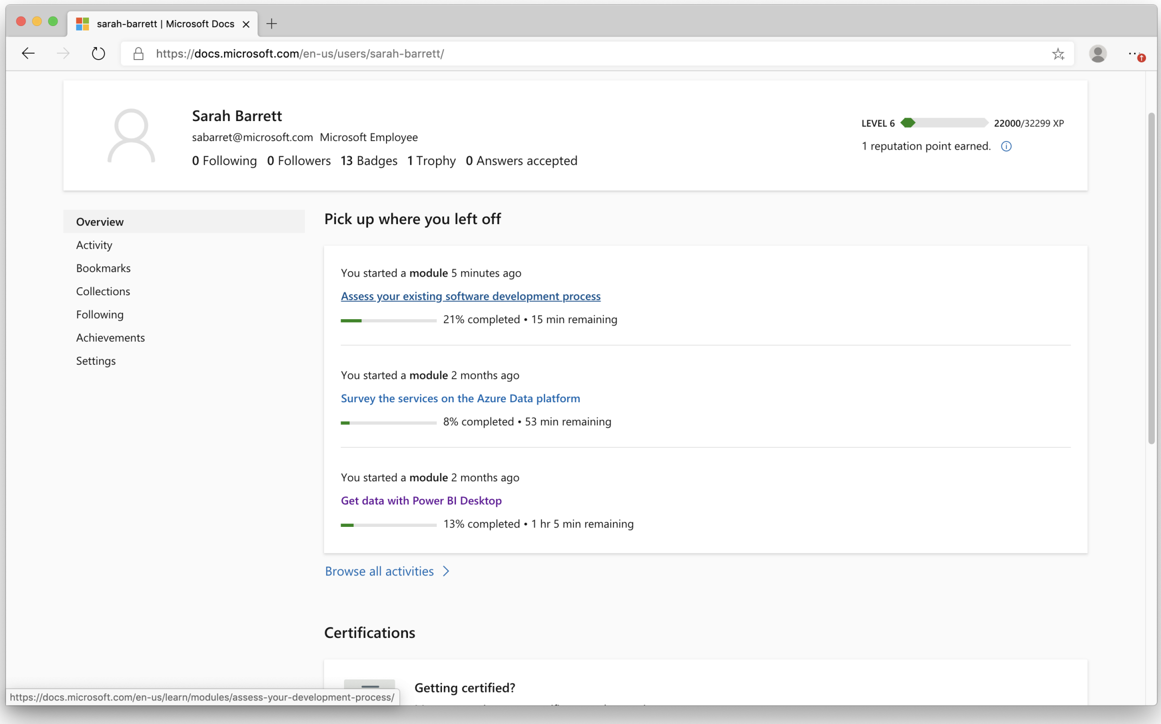
- none

Relationships with other content types

- Units in this module
- Module this unit is in

User

Status: Live



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
- Exam sessions for this user
- 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

- Attached content
- Certification UID
- End Date
- Start Date
- Status
- Title
- Expiry date
- Achievement date
- Renewal date

Taxonomies used on User Certifications

- Certification Type
- Level
- Product
- Role

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 then 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

7/7/2021 • 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?

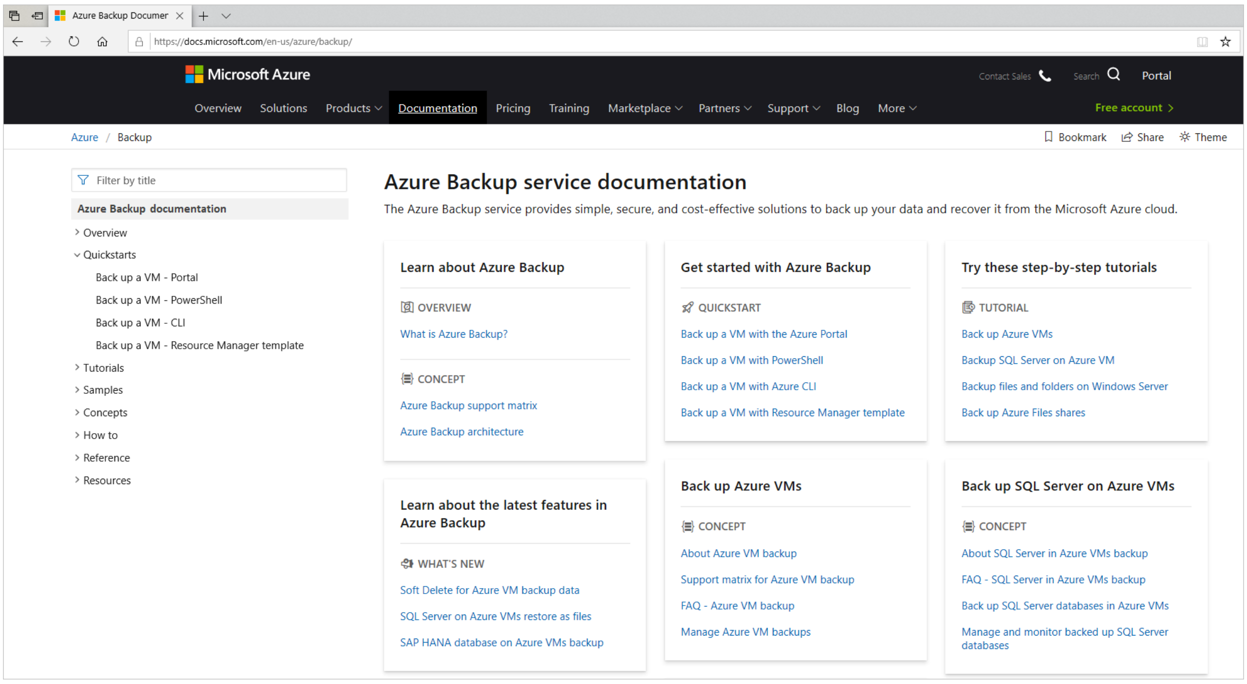
Most services have one TOC. For a larger product with multiple technologies, it might work better to have multiple TOCs, and then tie the TOCs together by linking to them from a product hub page. Deciding whether to keep all the technologies in the same TOC or break them up into multiple TOCs depends 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:

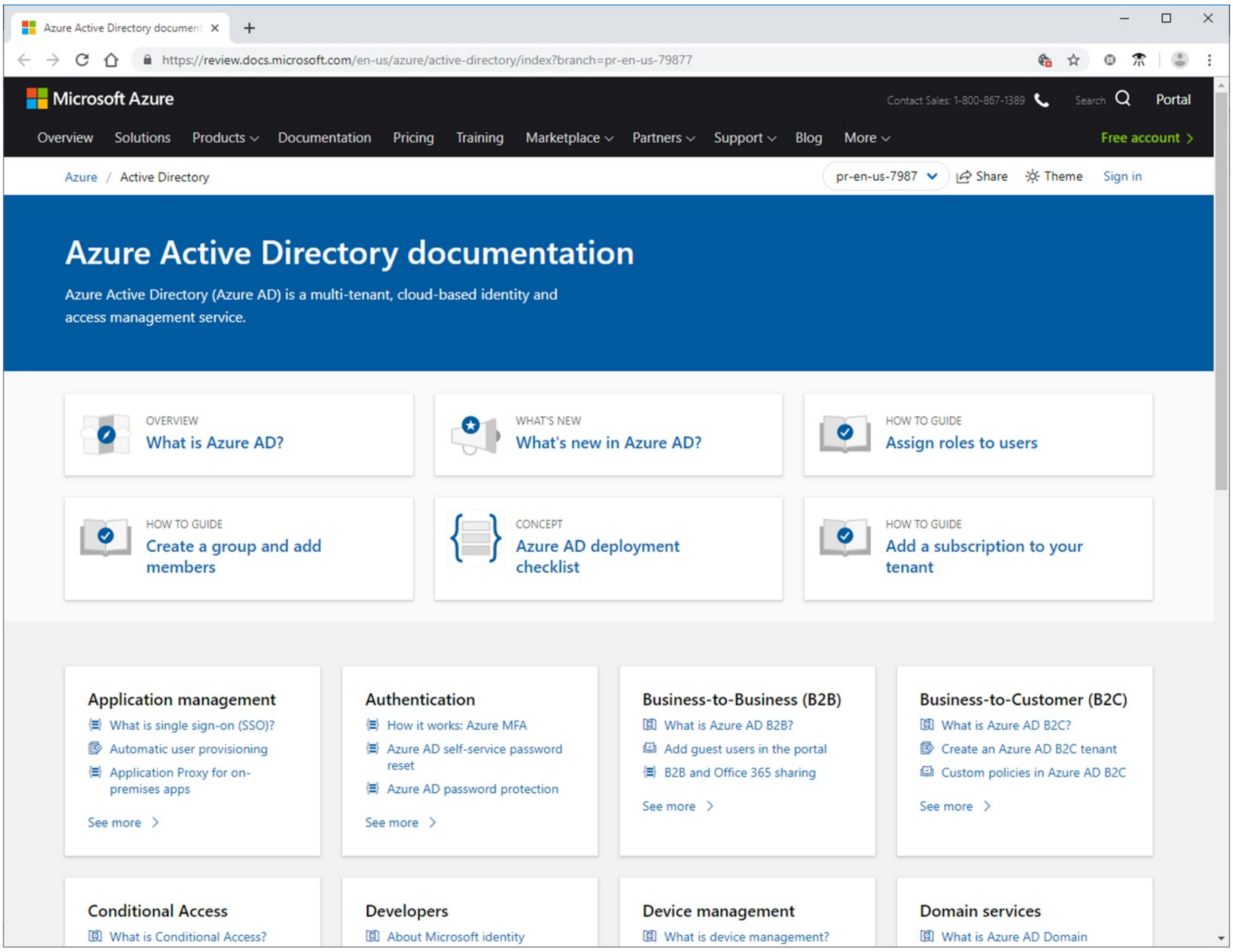
<Service name> documentation

\> 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.



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

7/7/2021 • 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:

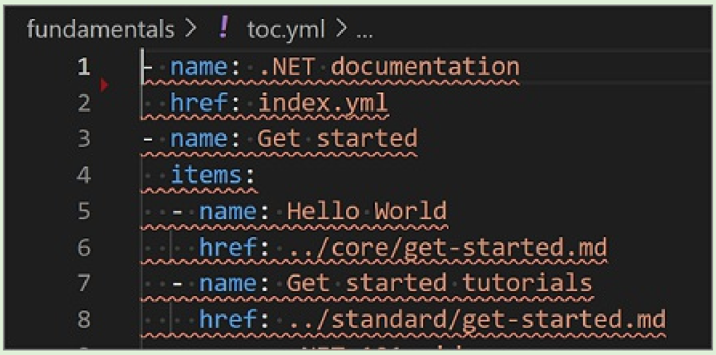
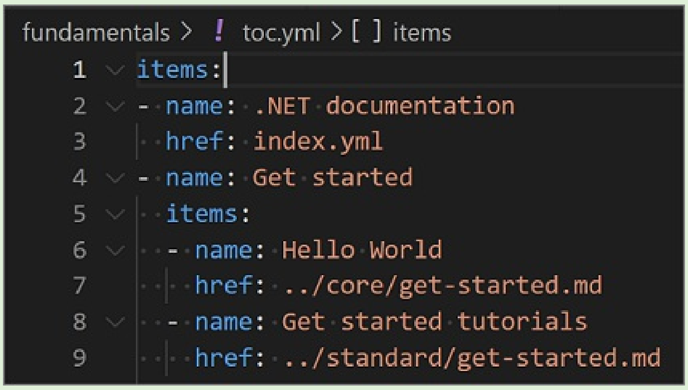
```
items:
- name: Tutorial
  items:
  - name: Introduction
    href: tutorial.md
  - name: Step 1
    href: step-1.md
  - name: Step 2
    href: step-2.md
  - name: Step 3
    href: step-3.md
```

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, asynchonous, 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.yamllint.com/`.
> - Add an `items:` tag as the first line in the file to appease the linter.
>
>   Before:
>
>   
>
>
>   After:
>
>   
>

Here's a larger example YAML structure, which includes more configuration options:

```yaml
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: VMScaleSets-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.

```json
"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.

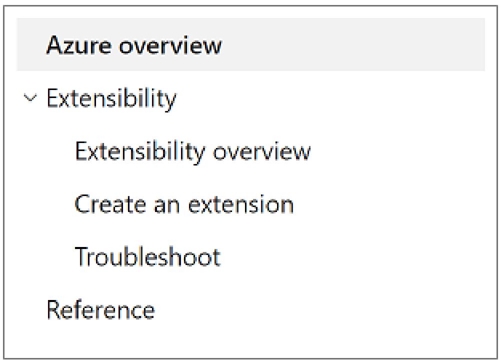
`toc.yml`:

```
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:



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

7/7/2021 • 12 minutes to read

This article explains how to set up your table of contents (TOC) and breadcrumb files so that when your TOC links to articles in another folder in your repo or another docs.microsoft.com repo, your TOC maintains its context.

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, which 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'll need to create or modify three files:

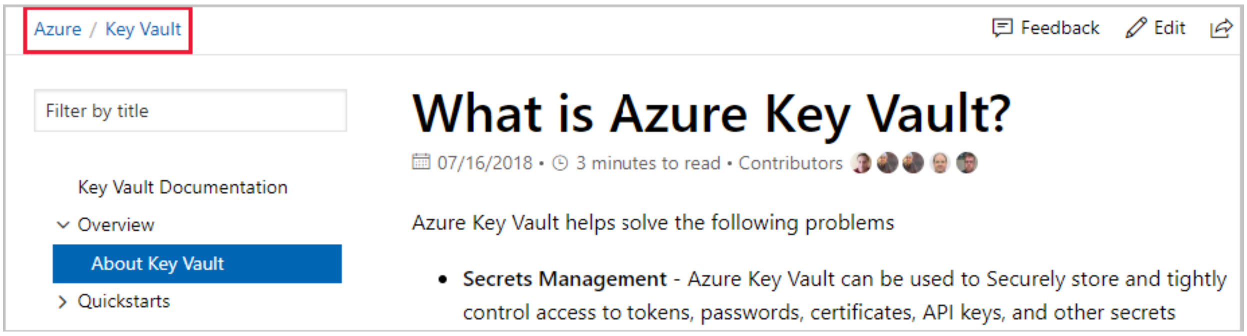
- **/breadcrumb/toc.yml** - To add mapping that displays a custom breadcrumb for an article that isn't in your folder.
- **toc.yml** - To add a link to a file in another folder or repo.
- **/context/context.yml** - [optional] To provide a short string to append to your URLs. This file is also helpful 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're trying to maintain. We'll show this configuration in more detail in the following steps. You can use a text editor, such as VS Code, to work on these files.

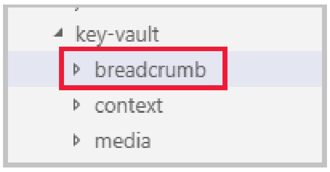
> **NOTE**
> It's best to create and publish breadcrumb and context files first in a separate pull request (PR) before making your TOC changes. This is especially important if you are using links to another repo - your breadcrumb changes will not preview if those files are not already published. These files are not seen by users unless they are digging though 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's 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.



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 don't 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'll 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'll 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 selects the link to the SQL Database 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 'll 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
azure-docs-pr	articles/sql-database/sql-database-always-encrypted-azure-key-vault.md	https://docs.microsoft.com/azure/sql-database/sql-database-always-encrypted-azure-key-vault

> **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 selects 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 selects 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'll point to later when you build your TOC file. For this example:

FILE	FILE LOCATION	PUBLIC URL
Breadcrumb	articles/key-vault/breadcrumb/toc.yml	https://docs.microsoft.com/azure/key-vault/breadcrumb/toc.json

   At this point, you're 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'll 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'll 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 selects 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		https://docs.microsoft.com/sccm/apps/deploy-use/create-deploy-scripts
Azure-docs-pr	articles/sql-database/sql-database-always-encrypted-azure-key-vault.md	https://docs.microsoft.com/azure/sql-database/sql-database-always-encrypted-azure-key-vault

2. Gather the full public URLs (without locale) of your breadcrumb TOC file and your documentation TOC file (in your repo/folder). You'll need them for testing later. For example:

FILE	FILE LOCATION	PUBLIC URL
Breadcrumb	articles/key-vault/breadcrumb/toc.yml	https://docs.microsoft.com/azure/key-vault/breadcrumb/toc.json
TOC	articles/key-vault/toc.yml	https://docs.microsoft.com/azure/key-vault/toc.json

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** 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're 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
```

c. For both scenarios above, you can link to a bookmark within an article by adding the bookmark syntax to the end of the URL. For example, if you want to link to the *Run a script* chapter of the *SCCM Run Scripts* article, you'd append the URL like this:

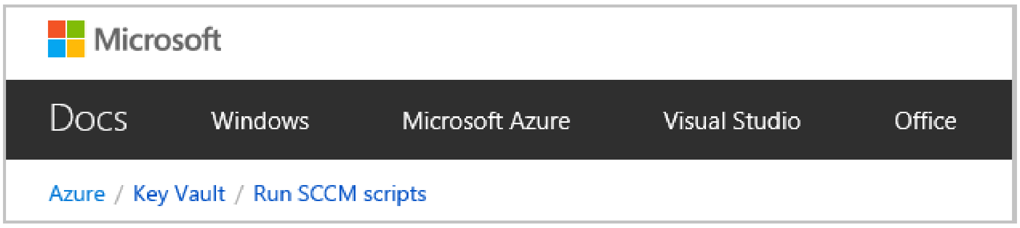
```
- name: SCCM Run Scripts
  href: /sccm/apps/deploy-use/create-deploy-scripts?toc=/azure/key-vault/toc.json&bc=/azure/key-vault/breadcrumb/toc.json#run-a-script
```

IMPORTANT

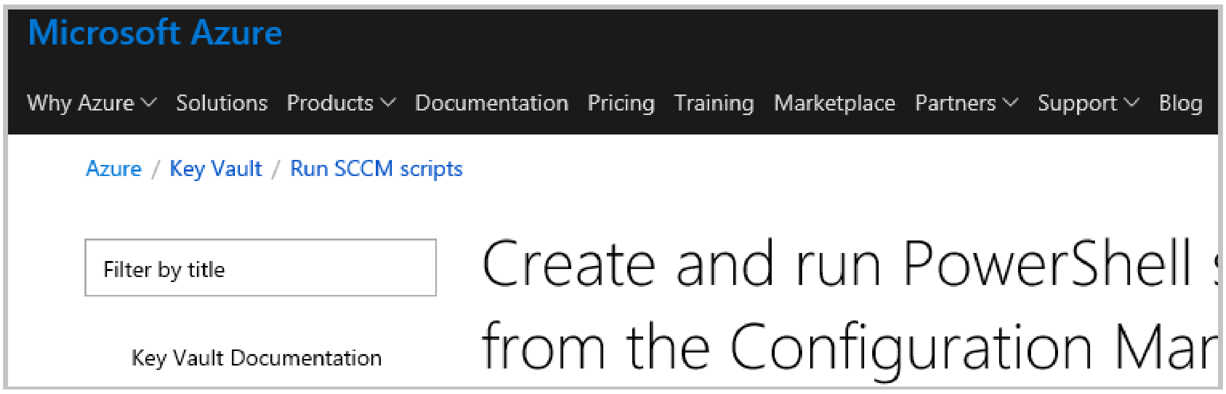
- The `href` value is not case-sensitive.
- The `href` value should not include the language code, `/en-us/`.
- DO NOT USE `maintainContext: true`.

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



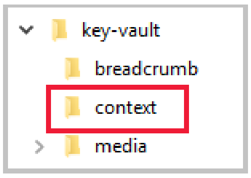
You can force the chrome appearance as well by using a context file. It's a user experience best practice to maintain a consistent chrome along with the TOC and breadcrumb.

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'll 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.

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

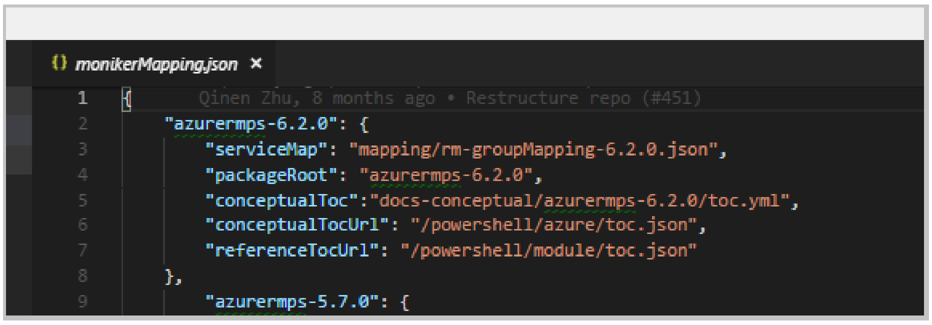
6. After you verify the context JSON file, update your documentation TOC to use the shorter contextual link. You no longer need to append the forced breadcrumb and toc paths to the link. Instead, add the shortened context string, for example: `?context=/azure/key-vault/context/kv-context`. This seems to work 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, once you've selected the new link and it has resolved in the browser, you'll no longer be in your branch. Any other new contextual links won't work unless you go back to your branch. For each new link that you need to test, you'll 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.



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

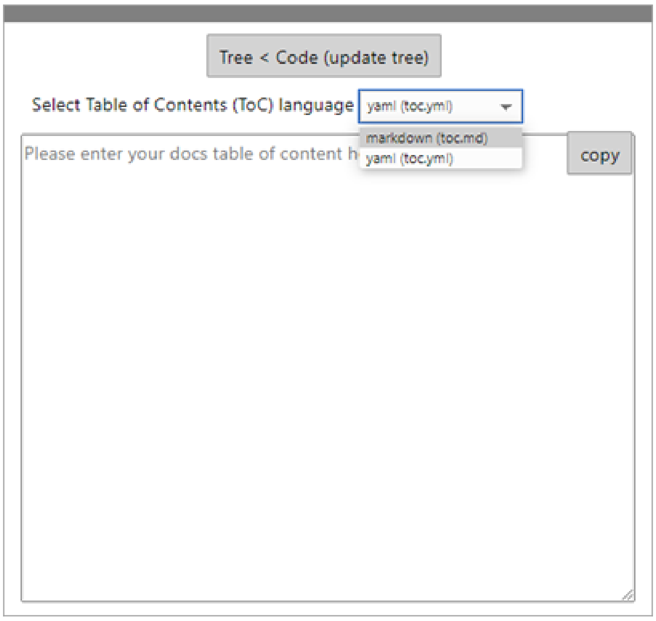
7/7/2021 • 2 minutes to read

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.



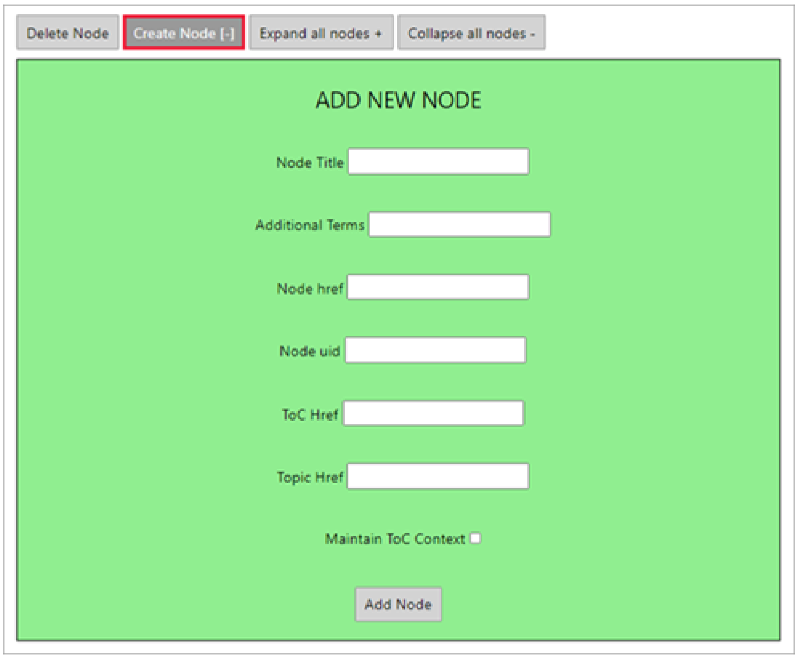
> **NOTE**
> You can't generate a new Markdown TOC.

Manage entries

This tool lets you create, delete, and edit the TOC entries.

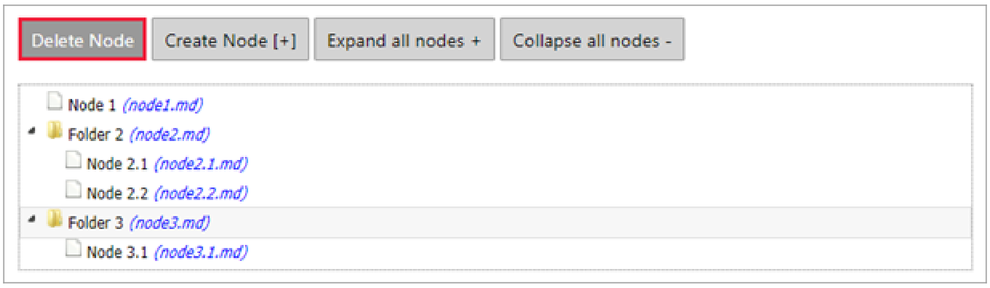
Create a new TOC entry

To add a new TOC entry, select **Manage Tree [+]** > **Create Node [+]**. The **ADD NEW NODE** section appears:

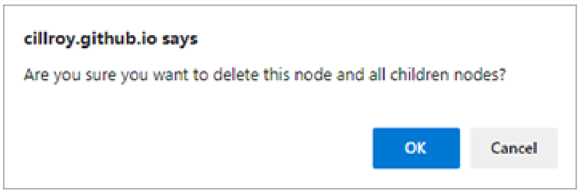


Delete an existing TOC entry

To remove a TOC entry, select **Manage Tree [+]**, select the node you want to remove, and select **Delete Node**:

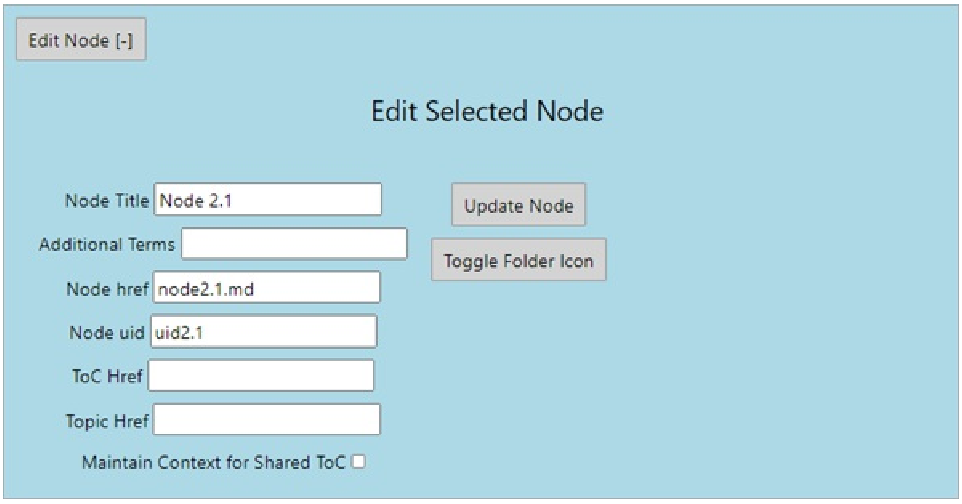


A dialog box appears asking you if you're sure you want to delete the node and its children:



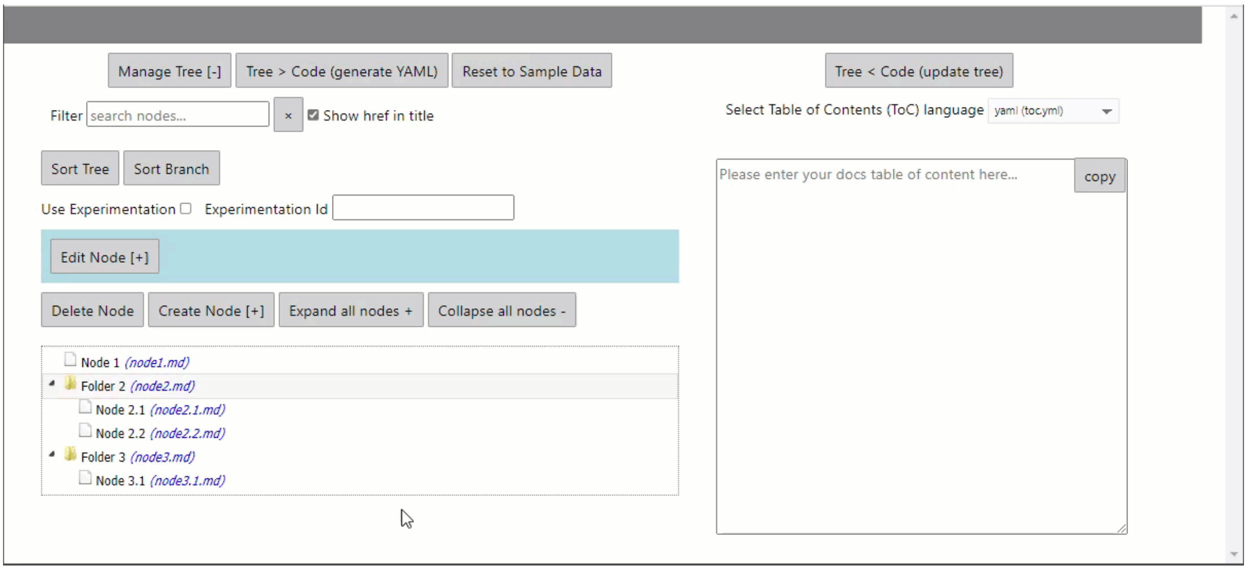
Edit an existing TOC entry

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:



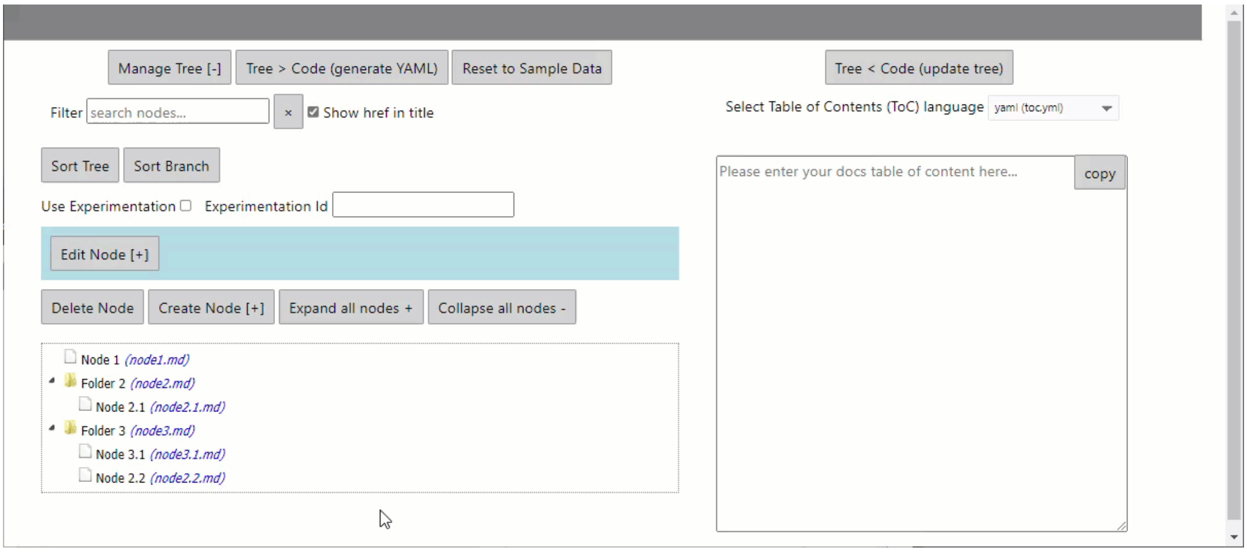
Drag entries

You can drag a node to any position within the TOC structure:



Sort TOC

You can sort the TOC at the branch level or sort the entire TOC:

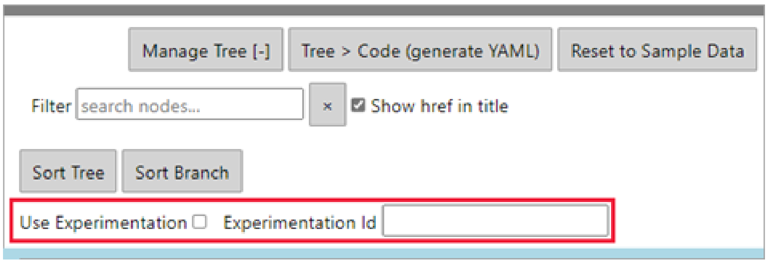


Support for advanced attributes

The TOC Visualizer provides support for advanced attributes.

Experimentation ID

Select the **Use Experimentation** option and enter an **Experimentation Id** value.

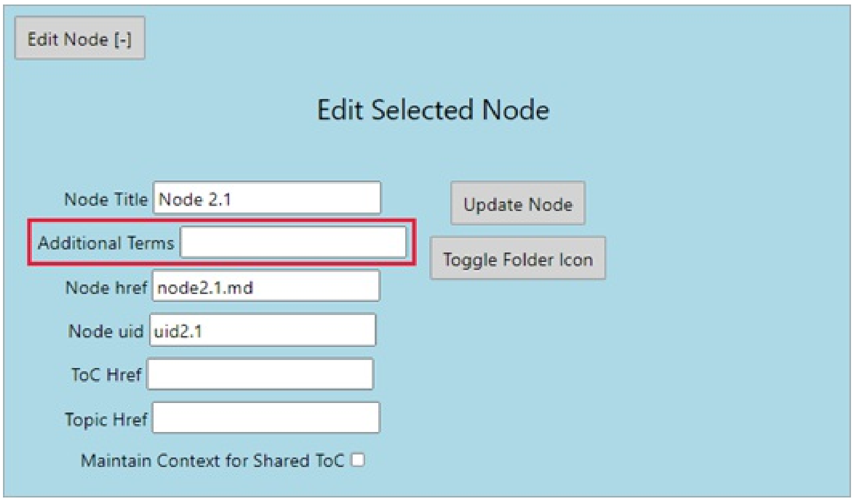


When you generate your YAML TOC code, you'll see a `metadata` block at the beginning:

```
metadata:
  experimental: true
  experiment_id: "<your-experimentation-id>"
...
```

Additional terms for TOC filtering on Docs

Enter an **Additional Terms** value.

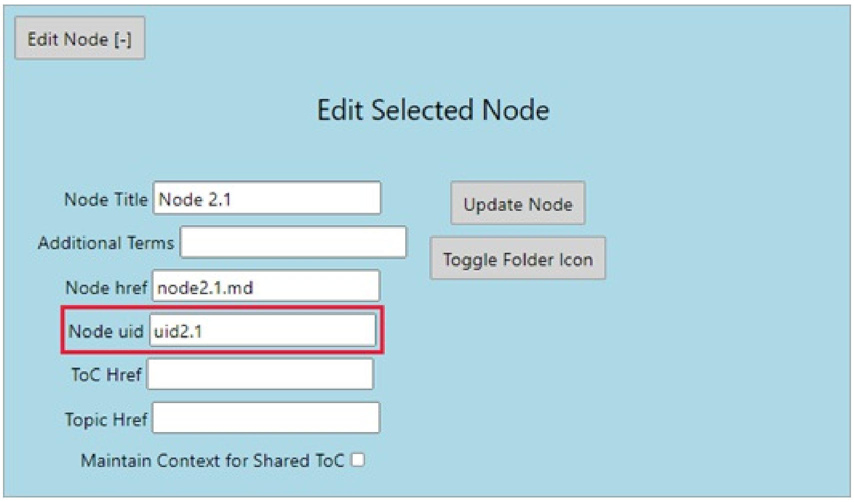


When you generate your YAML TOC code, you'll see a `displayName` value in the node block:

```
...
- name: Node 1
  displayName: <your-additional-term-value>
  href: node1.md
  ...
```

Node UID

Enter a **Node uid** value.

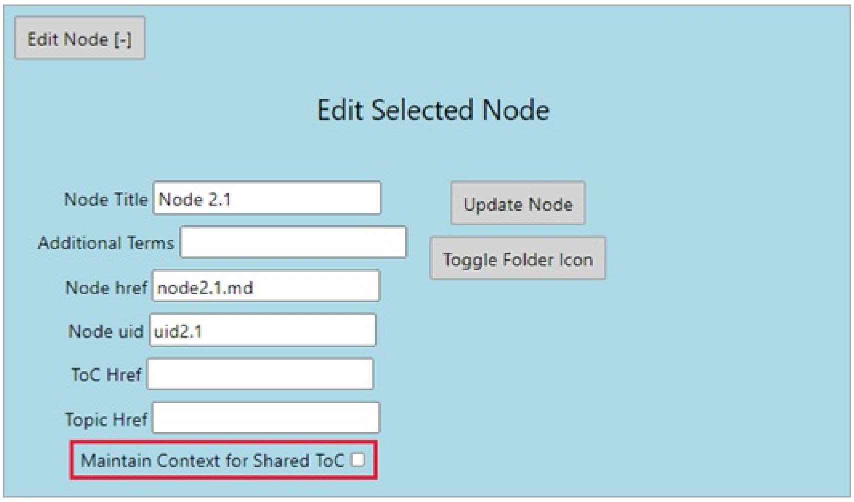


When you generate your YAML TOC code, you'll see a `uid` value in the node block:

```
  ...
  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.



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

7/7/2021 • 24 minutes to read

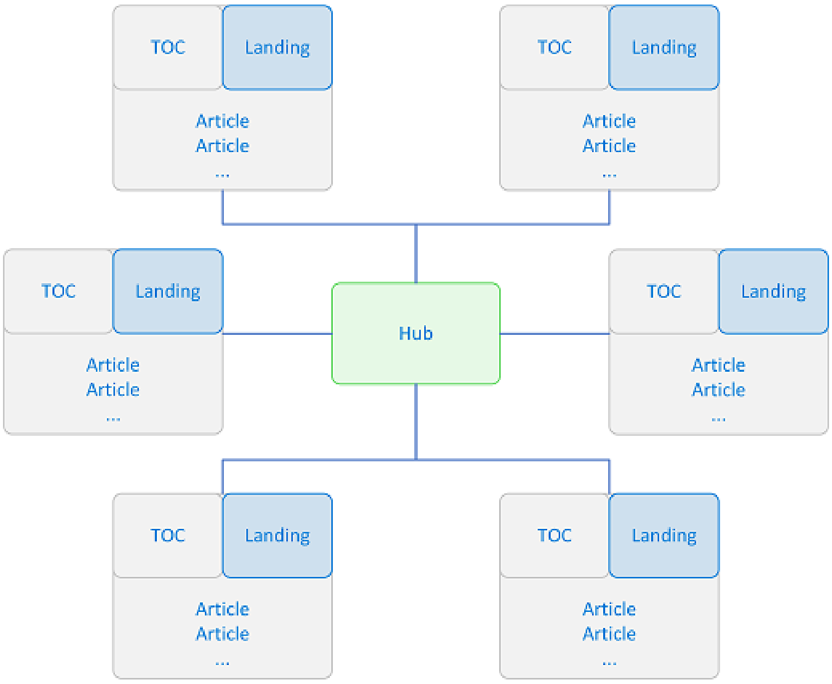
This topic is for Azure services and other products that follow the Content & Learning content model. It describes the high-level process to create or update a hub page.

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

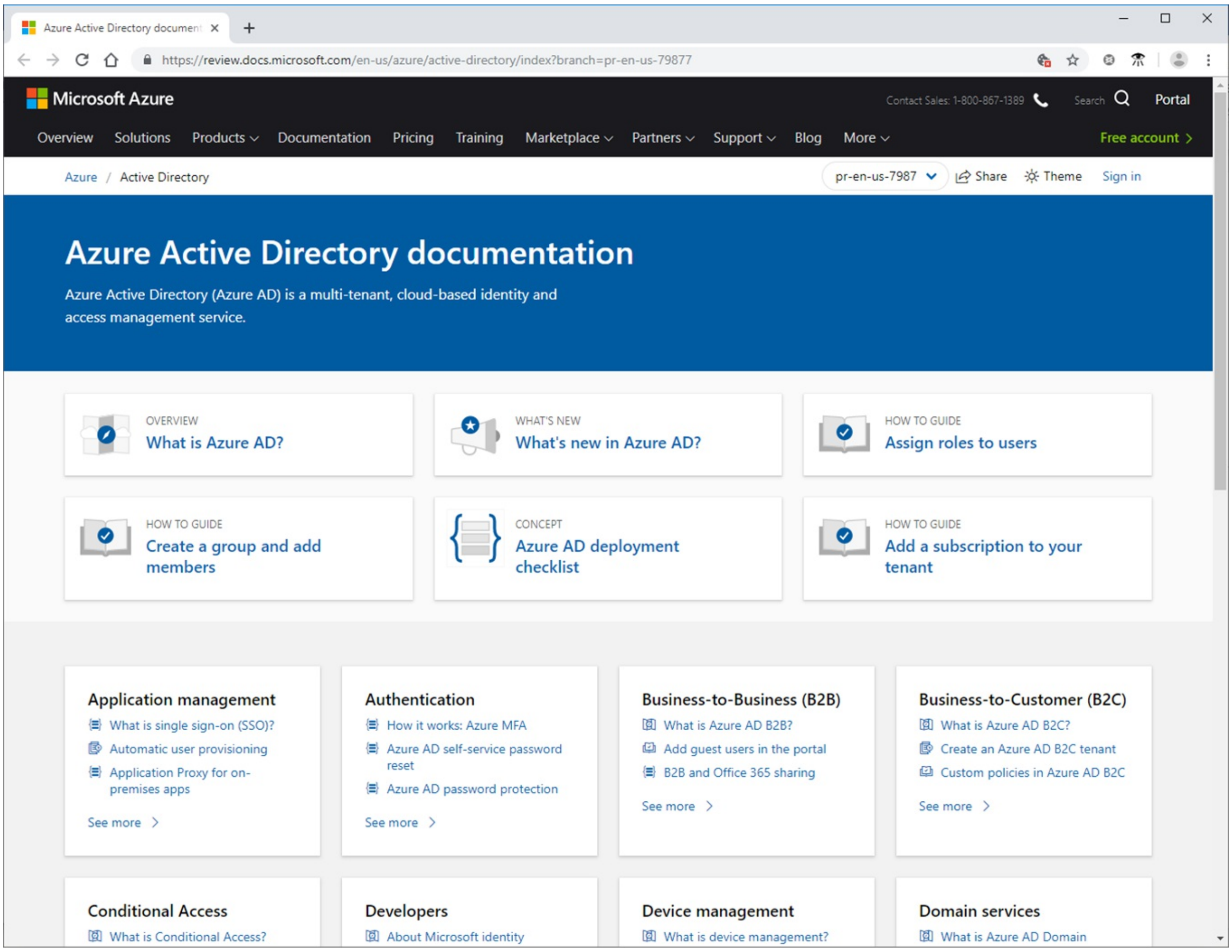
Design

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:

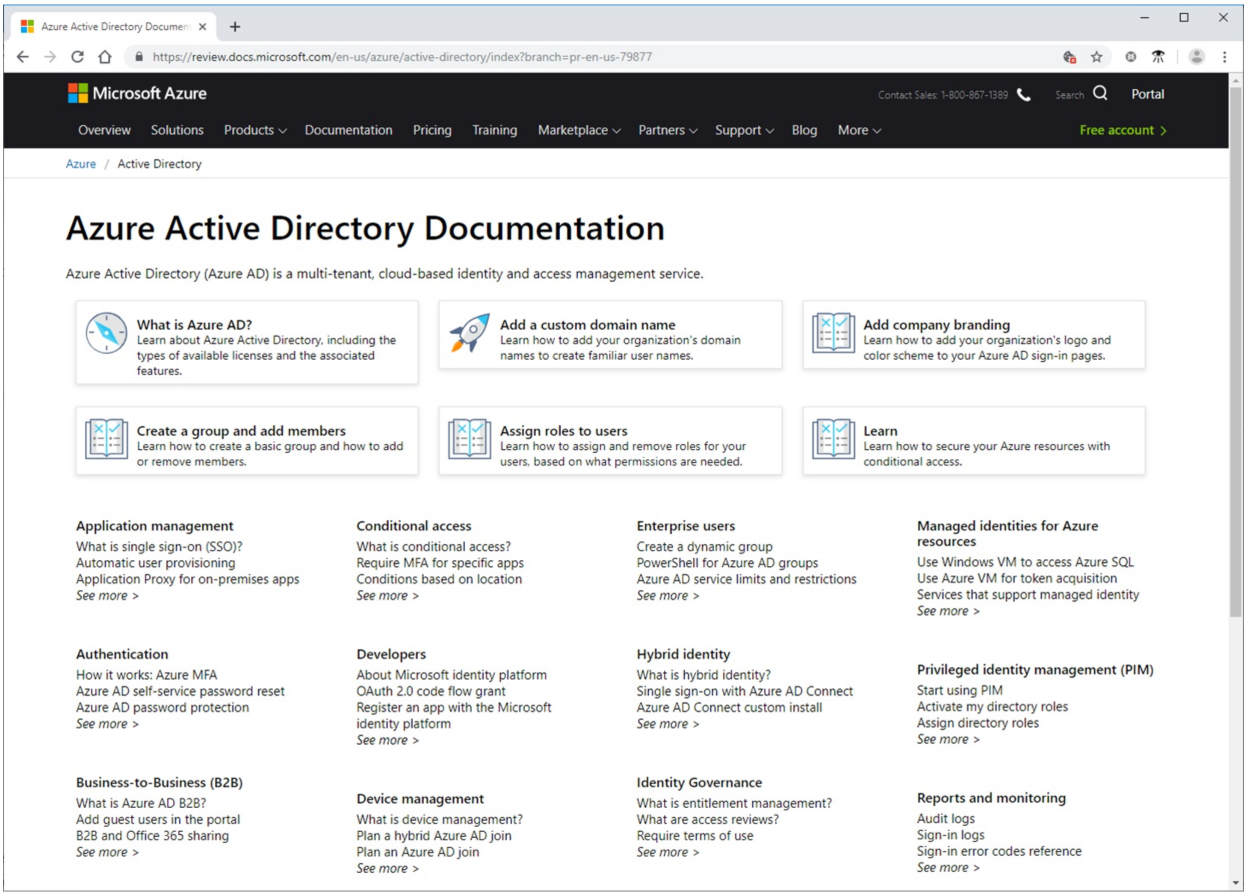
- Improve browsing experience for users to find content faster by creating a more dense presentation of info and revealing more content
- More focused on top customer tasks, products, or services

- 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:



Here's an example of the **previous** hub page design:



Example hub pages

HUB PAGE	SOURCE
Azure Active Directory	https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/active-directory/index.yml
Endpoint Configuration Manager	https://github.com/MicrosoftDocs/memdocs/blob/master/memdocs/index.yml
SQL	https://github.com/MicrosoftDocs/sql-docs/blob/live/docs/index.yml
Visual Studio	https://github.com/MicrosoftDocs/visualstudio-docs/blob/master/docs/index.yml

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 brand. < 60 chars
```

```yaml
  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.
  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: 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
```

```yaml
      # 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
```

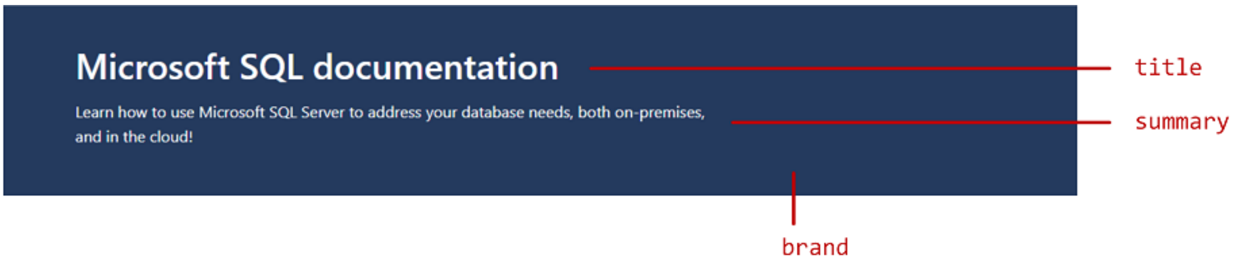
```
        - title: cardtitle2
          summary: cardsummary2
          url: file1.md OR https://docs.microsoft.com/file2
        # Card
        - title: cardtitle3
          summary: cardsummary3
          url: file1.md OR https://docs.microsoft.com/file3
  # footer (optional)
  footer: "footertext [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.

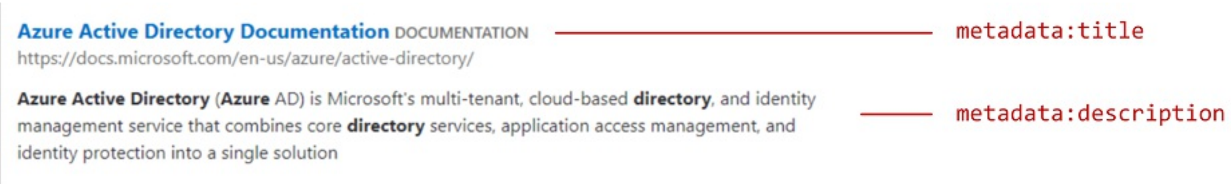


```
### YamlMime:Hub

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.

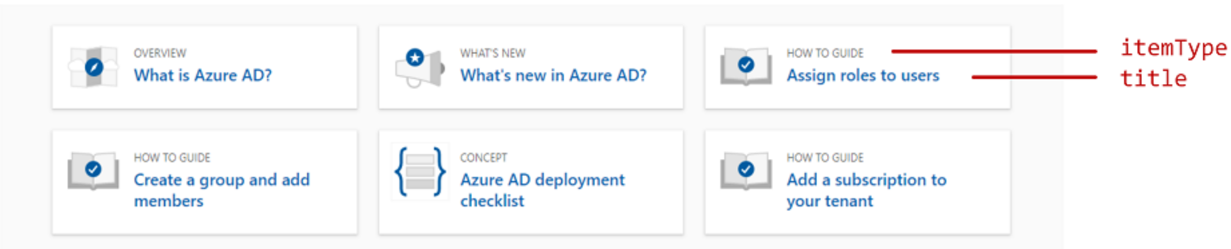



```
metadata:
  title: Azure Active Directory documentation
  description: Azure Active Directory (Azure AD) is Microsoft's multi-tenant, cloud-based directory, and
identity management service that combines core directory services, application access management, and
identity protection into a single 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

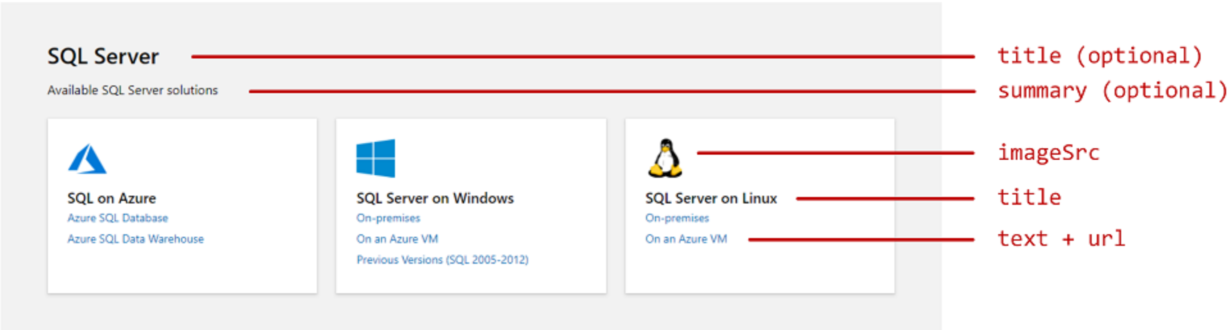


```
# highlightedContent section (optional)
# Maximum of 8 items
highlightedContent:
# itemType: architecture | concept | deploy | download | get-started | how-to-guide | learn | overview |
quickstart | reference | sample | tutorial | video | whats-new
  items:
    # Card
    - title: cardtitle1
      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

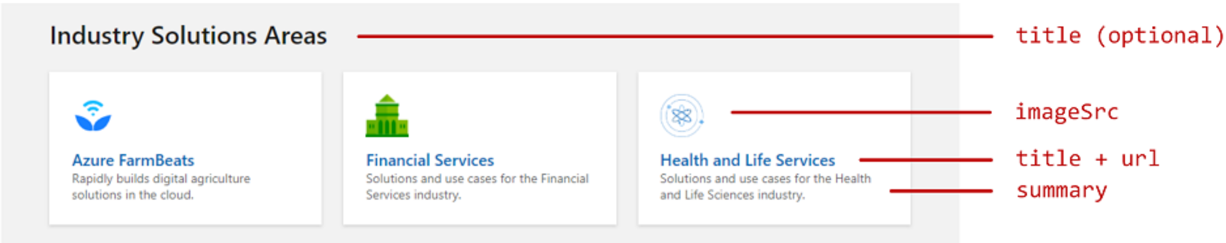
The `productDirectory` section is designed for product-related content and is optional. This section can have a title and a summary, and can contain 25 cards. Each card includes an image, a title, and links. The image is one that you specify and should be square in ratio with no whitespace.

ITEM	MIN	MAX
Number of cards	1	25
Number of links on a card	1	6



```yaml
# 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.

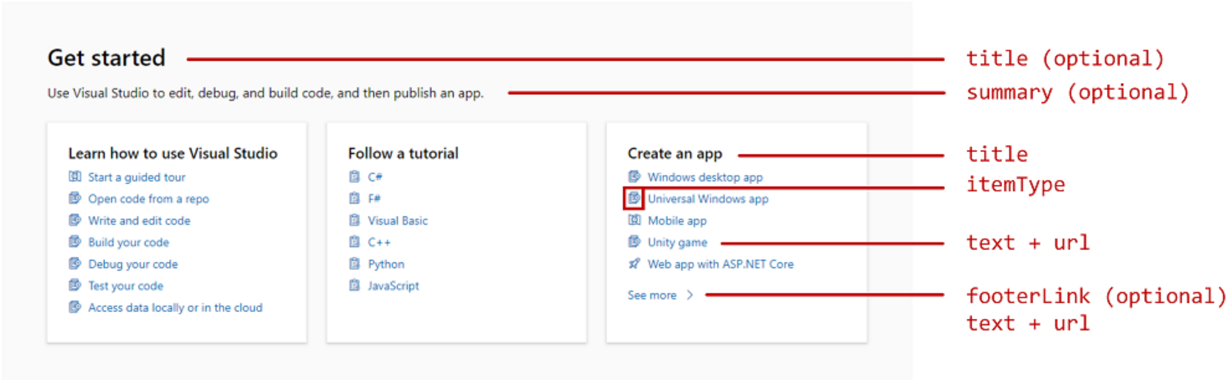



```yaml
# productDirectory section (optional)
productDirectory:
  title: sectiontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:
    # Card
    - title: cardtitle1
      # imageSrc should be square in ratio with no whitespace
      imageSrc: ./media/index/image1.svg 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

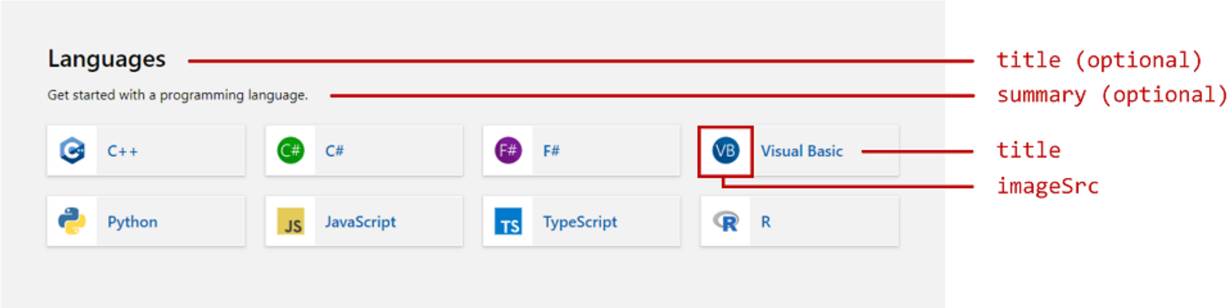


```yaml
# conceptualContent section (optional)
conceptualContent:
# Supports up to 3 sections
# itemType: architecture | concept | deploy | download | get-started | how-to-guide | learn | overview |
quickstart | reference | sample | tutorial | video | whats-new
  title: sectiontitle # < 60 chars (optional)
  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



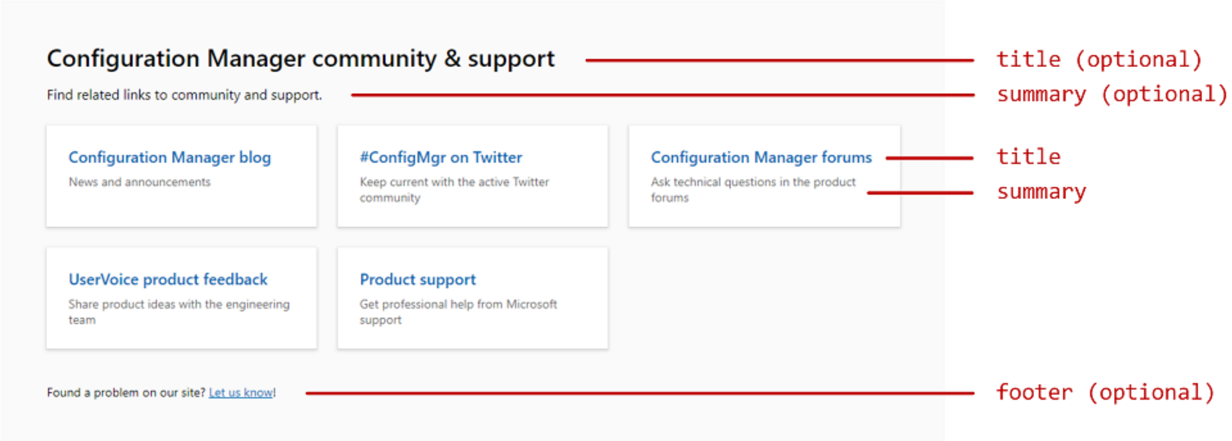

```
# tools section (optional)
tools:
  title: sectiontitle # < 60 chars (optional)
  summary: sectionsummary # < 160 chars (optional)
  items:
    # Card
    - title: cardtitle1
      # imageSrc should be square in ratio with no whitespace
      imageSrc: ./media/index/image1.svg OR https://docs.microsoft.com/media/logos/image1.svg
      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.

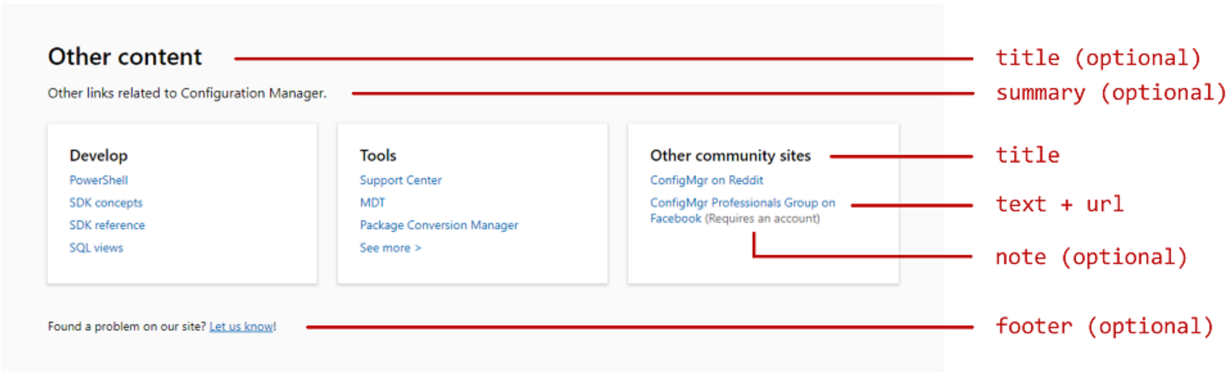
The following shows the `additionalContent` section for cards with the summary style. Each card includes a title, a summary (which allows for markdown), and a URL. The `url` key is only used if the `summary` key does not have a link. The `footer` section is the final optional subsection that is beneath 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




```
# 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.



```yaml
# 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 (`"`).

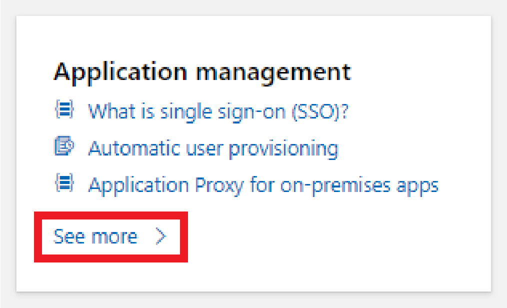
SPECIAL CHARACTER	EXAMPLE
Colon (:)	`- text: "Microsoft Ignite: Lock down access to Azure"`
Number sign (#) # is a comment in YAML	`- text: "See #azuredocs on Twitter"`

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.



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 uses 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.

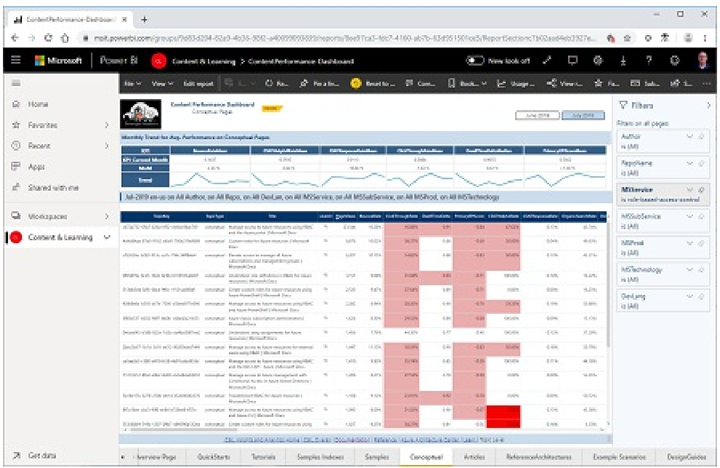
If you have questions about the best ways to organize content on your hub page or want to know more about the ways you can use user research to discover the most useful organization for your audience, please contact the DevRel Information Architecture team.

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 identity 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.

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 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 imageScr 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 itemType.
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

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

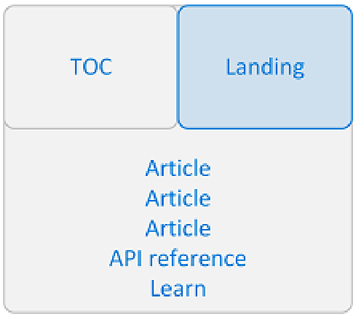
7/7/2021 • 8 minutes to read

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.

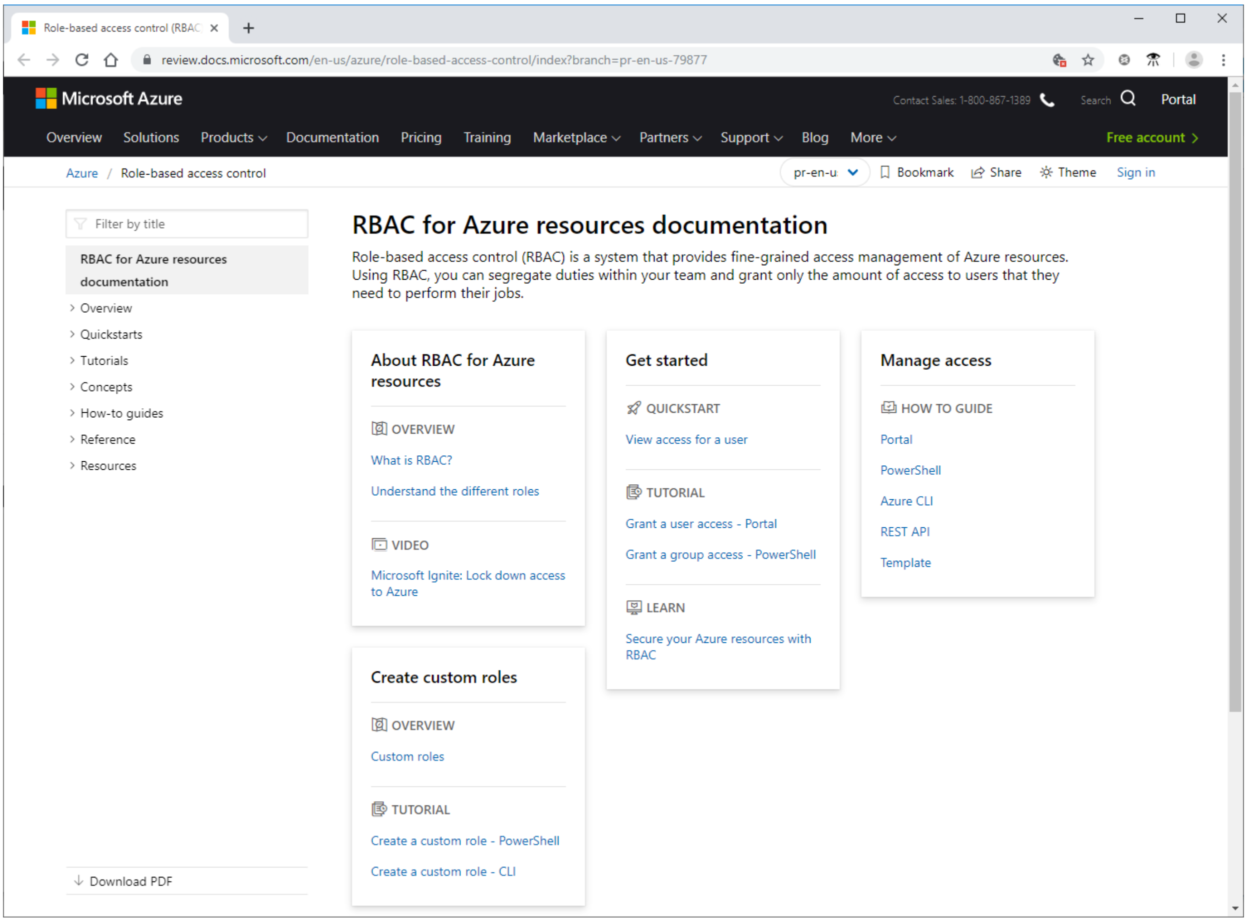


Design

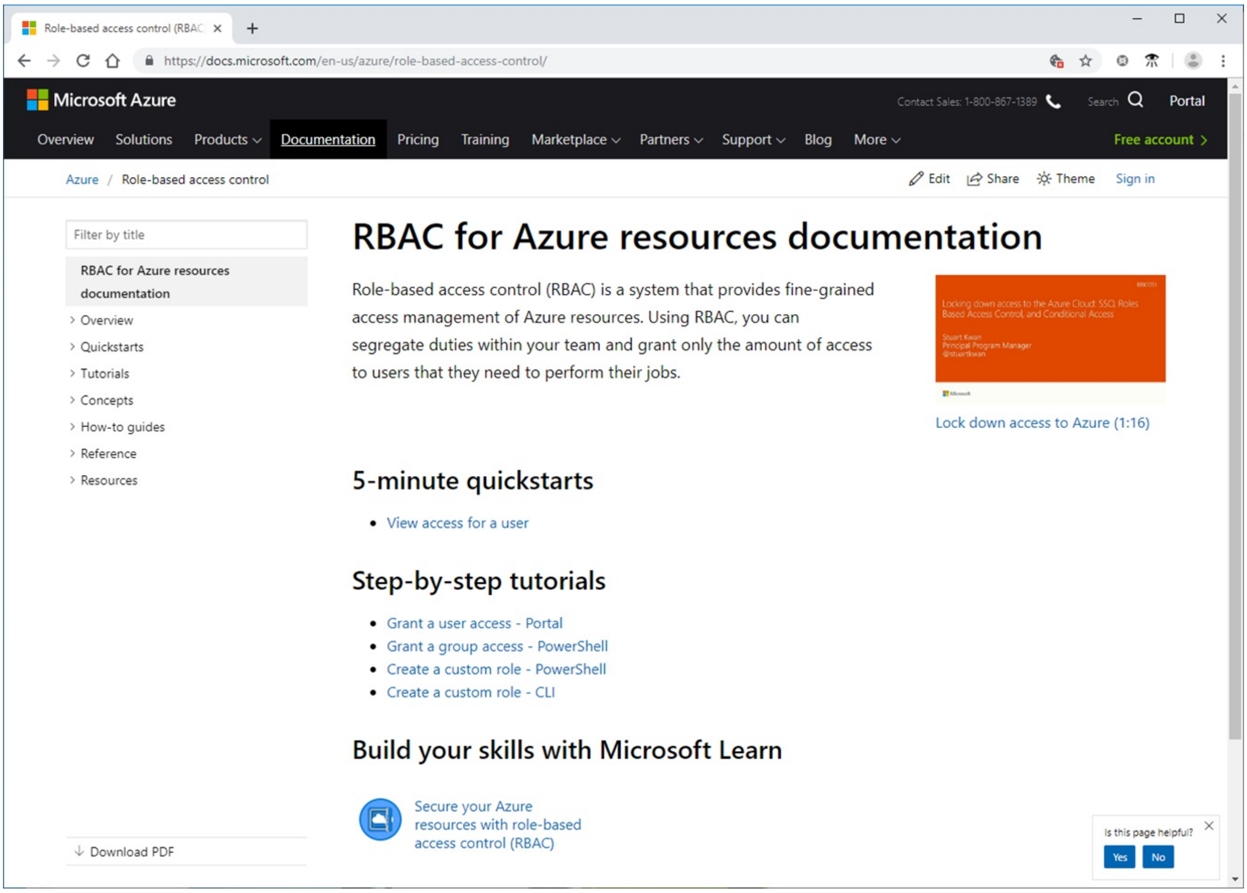
Starting in August 2019, a card-based landing page design was implemented that replaces the previous design. All teams in Content & Learning should use this new design going forward. This new design has the following benefits:

- 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:



Here's an example of the **previous** landing page design:

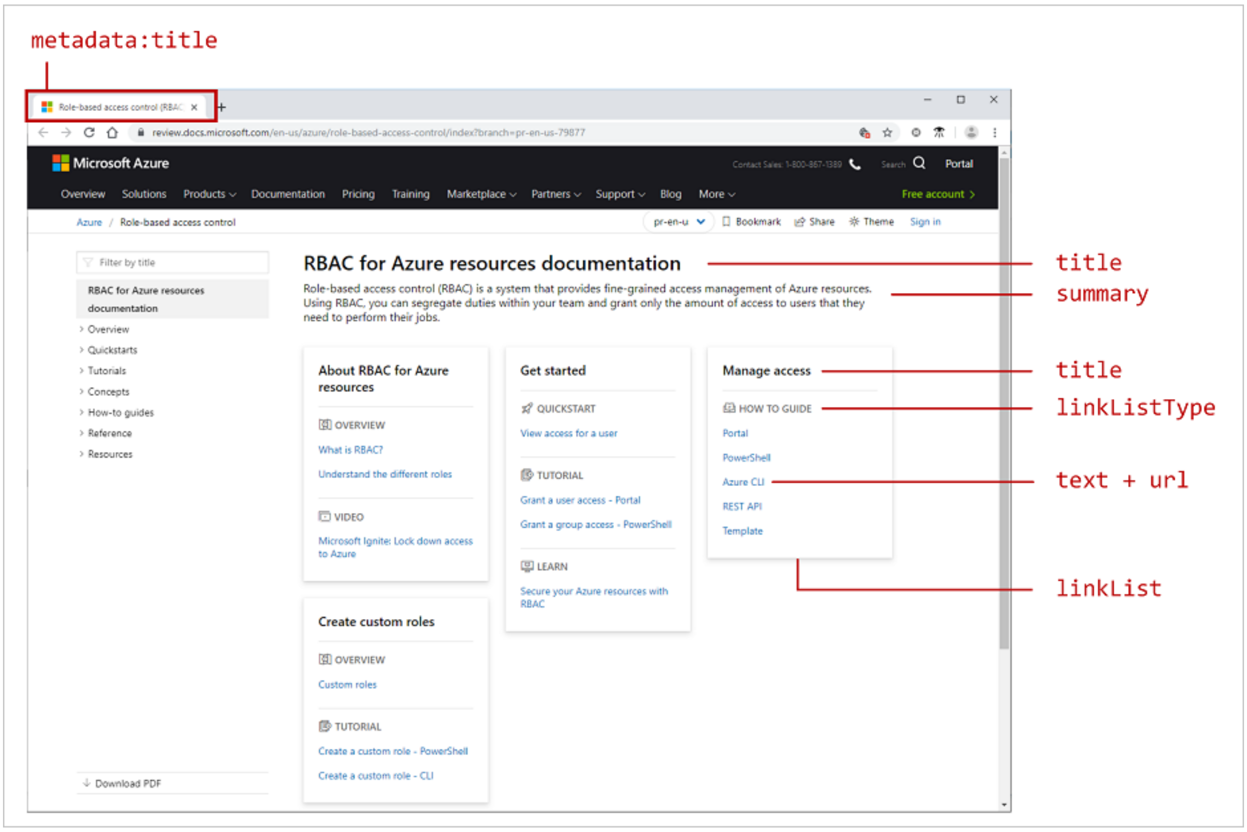


Example landing pages

LANDING PAGE	SOURCE
Azure Active Directory Identity Governance documentation	https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/active-directory/governance/index.yml
Azure Data Box documentation	https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/databox/index.yml
Azure AD Domain Services documentation	https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/active-directory-domain-services/index.yml
Azure IoT Edge documentation	https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/iot-edge/index.yml
Microsoft identity platform	https://github.com/MicrosoftDocs/azure-docs/blob/master/articles/active-directory/develop/index.yml

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.




```
### 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.
```

```yaml
  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

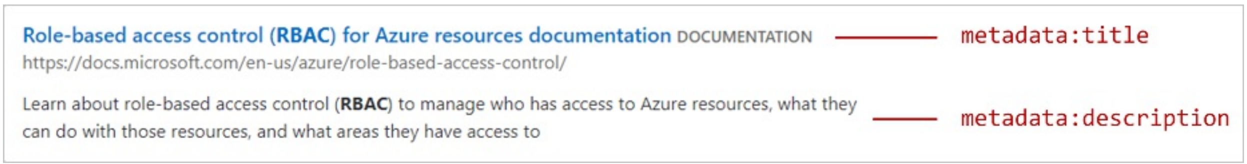
  # Card
  - title: cardtitle4
    linkLists:
      - linkListType: linkListType
        links:
          - text: linktext1
            url: file1.md
          - text: linktext2
            url: file2.md
```

```
    - 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
```



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 your exceed these limits, you will get a validation error when building.

ITEM	MAX
Number of cards	12
Number of `linkListType` in a card	6
Number of links in a `linkListType`	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 (:)	`- text: "Microsoft Ignite: Lock down access to Azure"`

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:

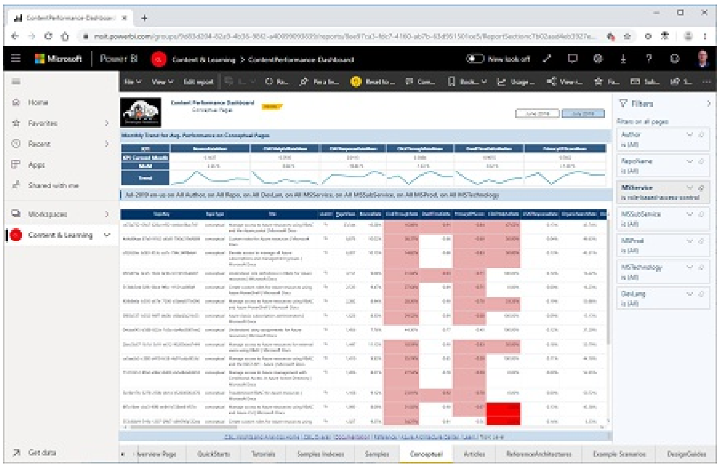
- Cards and links should be based on top customer tasks or top subjects.
- Don't just repeat the TOC.
- Order cards by relevancy.
- Card titles should start with a verb.
- When possible, use relative paths for links.
- `linkListType` should match the content type.
- Don't duplicate links.
- Link directly 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 identity 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 linkListType.

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.

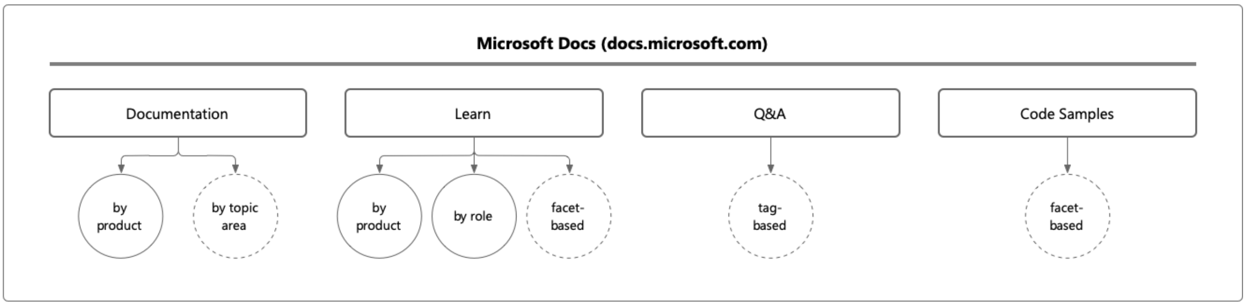
Header navigation governance

6/28/2021 • 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":

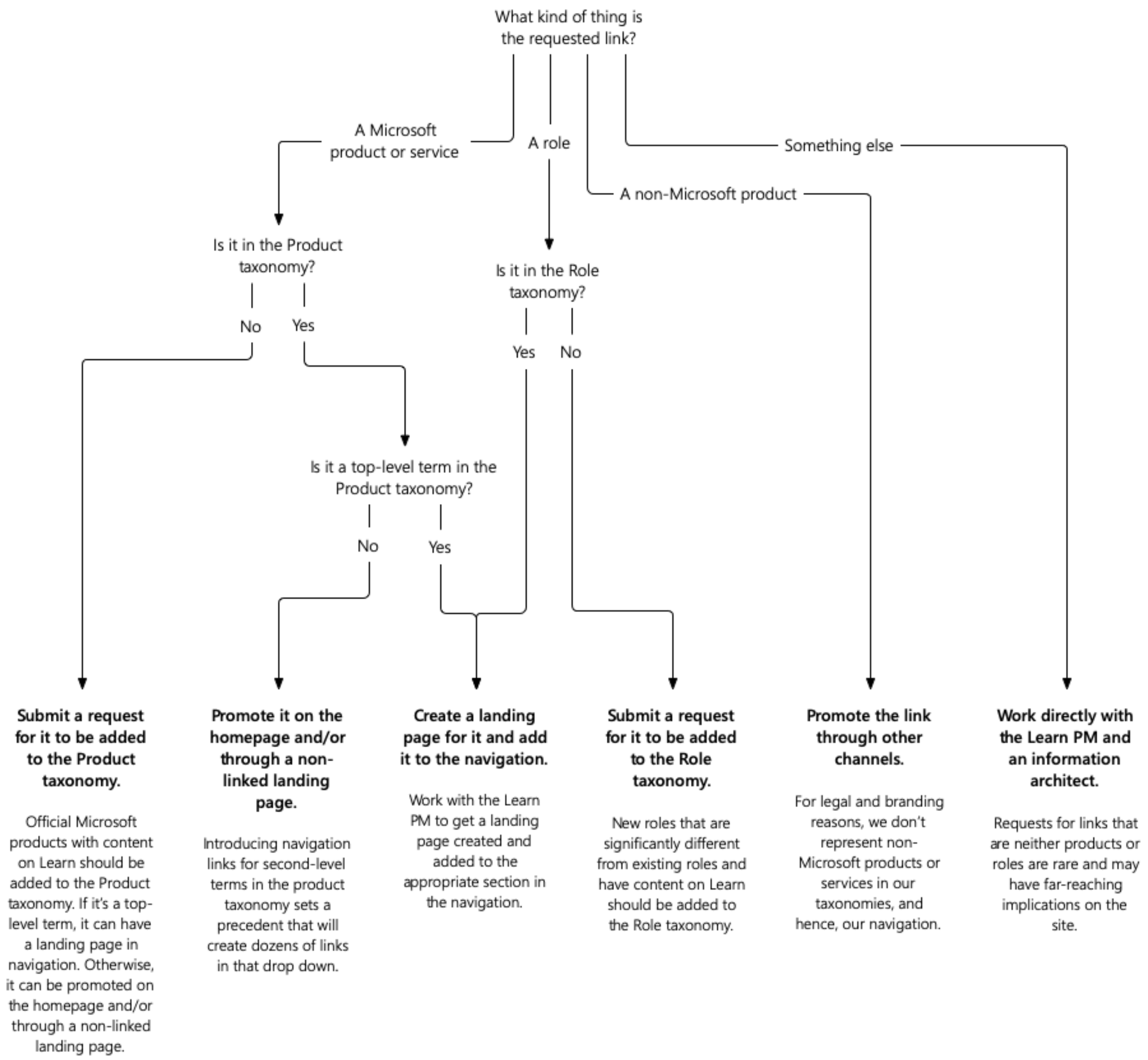
- Learning paths & modules (organized by Product and by Role)
- Learn TV (videos)
- Certifications & courses

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 DevRelIA@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

7/7/2021 • 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

6/29/2021 • 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

☑	TASKS
☐	Identify the lead PM, sponsor, and subject matter experts for the new content.
☐	Identify the expected timeline for the release.
☐	Identify if a repository for your need exists. If not, request the creation of a content repo.
☐	Work with your product team to understand customer challenges or user scenarios to help determine the content that you want to create. Review the writing principles and the acceptable content for Microsoft Docs.
☐	Consider whether you'll need reference content for any APIs, libraries, packages, or CLIs. Onboarding times can be long, so take that into consideration.
☐	Determine if the release is public preview or general availability. For a preview, see Identify preview content, and review how it will affect your table of contents and landing page.

Design

☑	TASKS

☑	TASKS
☐	*If a repo doesn't exist*: 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. For example, *virtual-machines* is the service slug for Azure Virtual Machines and is consistently used between the Docs site (https://docs.microsoft.com/azure/virtual-machines/) and the ACOM site (https://azure.microsoft.com/services/virtual-machines/).
	If a repo already exists: Work with the sponsor to coordinate your new content.
☐	Create a new release branch for your new content.
☐	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. For example, this folder matching the service slug is used for the Azure Virtual Machines articles: https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/virtual-machines.
☐	Create the new metadata value. Use the new ms.service value request form to add the new metadata to the list. Be sure to select **Yes** to include your new Docs metrics in the content engagement reports. The ms.service metadata property typically uses the same value as the service slug.
☐	Get the new .svg icon from the marketing representative for your release or product.
☐	Based on your content needs, list the articles that you need to create. Review the recommended article types for Docs.
☐	Based on your article list, define the TOC structure 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 TOC checklist spreadsheet to draft your TOC.
☐	Start thinking about landing pages and hub pages 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.
☐	Add your content set to the main breadcrumb file: \bread\toc.yml. Review the breadcrumb guidelines.

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

Write

☑	TASKS
☐	Ensure your articles align with the docs.microsoft.com writing and technical principles checklist.
☐	Understand the various content types before you create your content.
☐	Understand and incorporate SEO techniques in your content.
☐	Ensure your content has all the required Metadata.
☐	Run Acrolinx to ensure compliance with Microsoft standards.
☐	Create your TOC.
☐	Create your Landing page.
☐	Divy Sharma (DIVYARATNA) is responsible to include your new content. Confirm with Divy Sharma that everything is on schedule.

Review and publish

☑	TASKS
☐	When your content is ready, understand the pull-request (PR) submission recommendations and create a PR to receive feedback from SMEs.
☐	Check the build validation, and address any issues.
☐	Check the Acrolinx results, if available. See Acrolinx coverage.
☐	Invite reviewers to provide feedback on the new content.

☑	TASKS
☐	After you incorporate the reviews and feedback, sign off (**#sign-off**) 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.
☐	Schedule your release branch merge with the main branch.
☐	Request Go Live.
☐	Generate a forward link (FWLink) to your landing page so your product team can link to the content from their product's user interface.
☐	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 PLR sign-off for articles.
☐	After the content is published, do a visual pass of all the content, including any hub and landing pages.

Maintain

☑	TASKS
☐	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.
☐	Review content performance.
☐	Address any GitHub issues.
☐	Retire old content.

How content resources are assigned to product documentation

6/24/2021 • 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
Criteria	- Cloud first - Strategic or big bet - Clear development roadmap - Engineering team availability - High customer/install-base/revenue - Available content team resources	- Strategic or big bet - Clear development roadmap - Engineering team availability - Medium customer/install-base/revenue - Available content team resources	- Development roadmap unclear - Or low/no engineering team availability - Or low customer/install-base/revenue - Or lack of content team resources (Can you fund?)
Content Strategy	✓	✓	✓
Shared Content Creation	✓	✓	✗
Onboarding + Training	✓	✓	✓
IA/TOC/Hub + Landing pages	✓	✓	✓
Editorial Service	✓	✓	✓
Content Dashboards	✓	✓	✓
Quarterly Reports	✓	✓	✓
Monthly Sprint Reports	✓	✓	✗
Top Scenarios + Tasks	✓	✓	✗
Release Management	✓	✓	✗
Customer Research	✓	✓	✗
Active Monitoring and Improvements	✓	✓	✗

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

7/7/2021 • 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

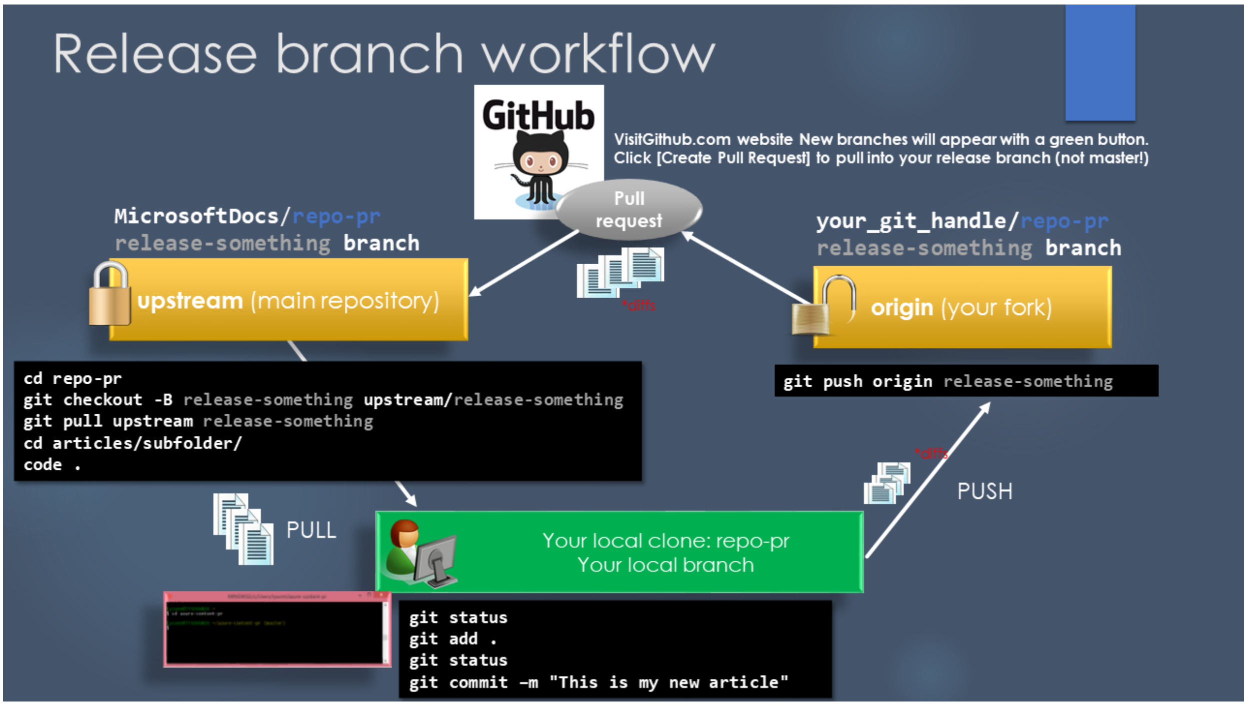
Understand release branches

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.

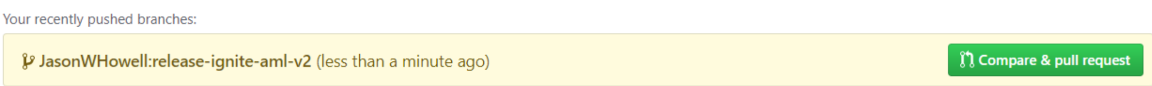
1. Once the changes are ready, issue a PR.

   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>
   ```

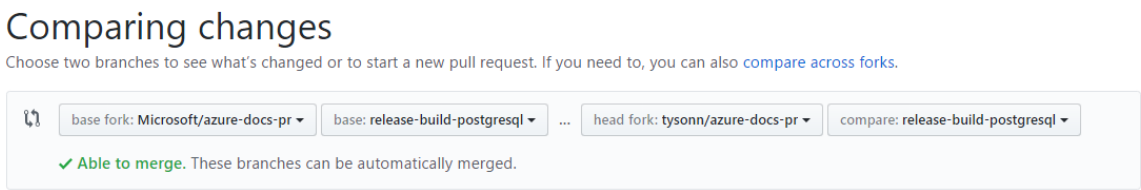
   GitHub represents recent new branches created in your fork with a yellow alert bar and green button to find them with ease.



> **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.



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

7/9/2021 • 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 revealed to only a select customer base. We don't 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 doesn't 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. The allowance includes content for *gated public previews* where the product, service, or feature is publicly revealed and available. The *gated* part of the preview means that 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.

Don't add *Preview* or *(preview)* to your TOC labels.

Body

Use the general naming guidance above to identify product, service, or feature previews. Do so on first mention in the body and in each major section. If context is clear, other mentions in each major section can omit *Preview* or *(preview)*.

Whenever possible, avoid extra or generic use of *preview* in your articles. You may need to discuss functionality or limitations directly related to the preview release. If so, 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 Markdown 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). 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 repo?

7/7/2021 • 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 synced 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 the article author reviews their contributions. For example, if you work in CSS and only contribute occasionally, working in the public repo 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. Various quality support features are present in the private repos:

   - Build-time validation runs on the updated article to check links, metadata values, and other basics.
   - Most private repos have staging enabled at the review.docs.microsoft.com site. Review all staged updates before 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. You might be working on a time-sensitive release like a feature leading up to a conference, embargoed content, a public preview launch, or a general availability launch. If that's the case, use the private repo and a private release branch. Doing so will protect that content until it's ready for release.

3. If you're updating an article where you're the listed author, use the private repo. The PRMerger system automatically closes pull requests in a public repo if you're 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'll 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 repo that you know exists, try signing in to GitHub first. If you're already signed in, that 404 error typically means you don't have access. To get access, your GitHub account needs to be set up with two-factor authentication, and joined to the right the organization (like 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 already made changes in the public repo, you can move those changes to the private repo. Use 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

A contributor made changes in the public repo in a branch called `patch-124`. The contributor can pull those changes to their local clone's master branch. The contributor's GitHub user name is GitHubBob, and their using `patch-124` branch:

```
git pull https://github.com/GitHubBob/azure-docs.git patch-124
```

Then, GitHubBob does `git add`, `git commit`, `git push origin`, and creates a pull request to the private `master` branch.

Editorial service for technical content

6/24/2021 • 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 listed in the style guidelines page, as well as the order to use the guides in.
- **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)

6/24/2021 • 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**: inks 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.

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 before applying for access. **For Power Platform repos**, please contact Buckley Guderian buckley.guderian@microsoft.com before applying for access. **For Azure repos**, please contact Justin Hall 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
- Repo URL: 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
- Repo URL: 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
- Repo URL: 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

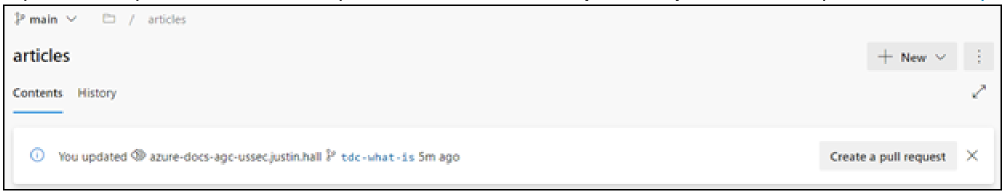
The workflow between GitHub and Azure DevOps is similar. If you're using VSCode with the Docs Authoring pack, you can still do so, however Acrolinx won't be available. For information on general steps you can adapt, see Make changes to an article.

NOTE

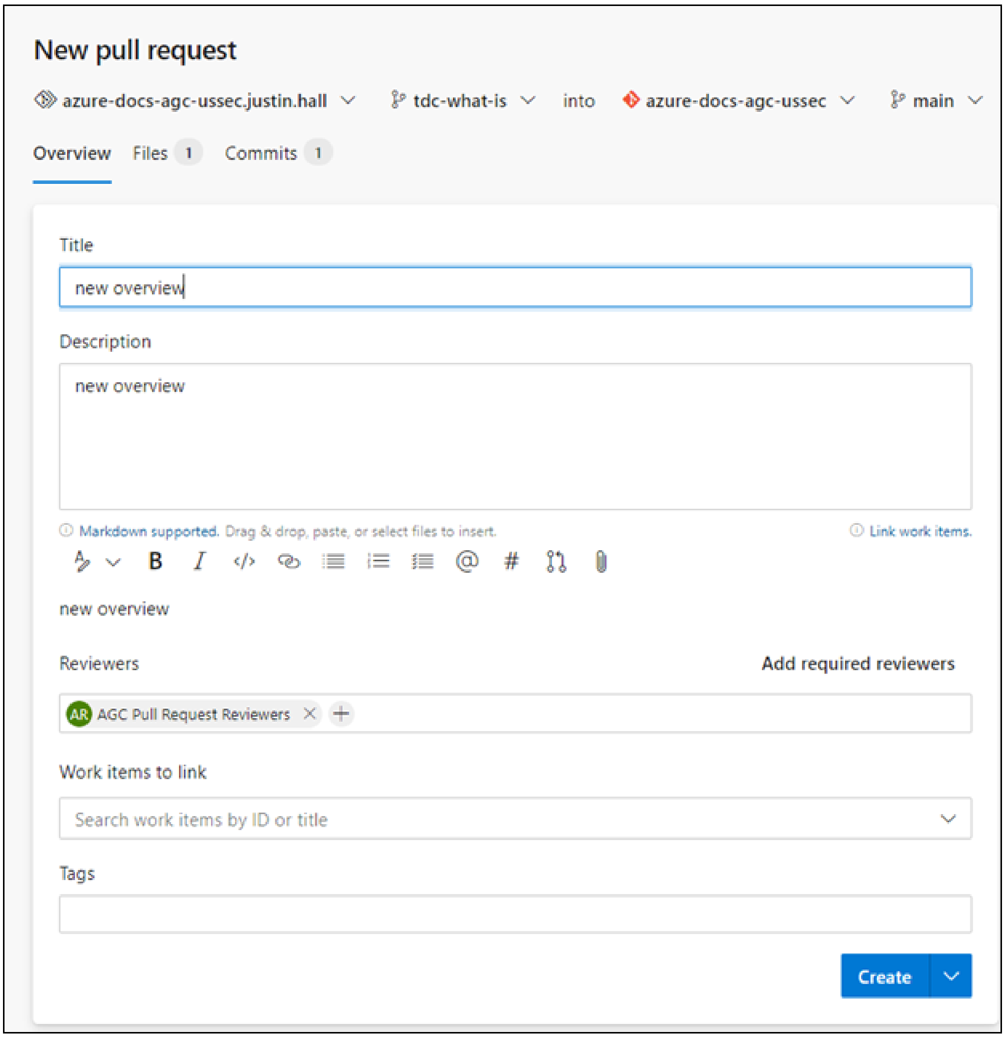
For **Azure repos**, please contact Joelle Faustin **before** contributing your first pull request. Joelle is coordinating region specific updates. As noted above, for Office Microsoft 365 repos, the contact is David Zazzo.

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
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**.



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

7/7/2021 • 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 with 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

6/28/2021 • 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 devrelIA@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

7/14/2021 • 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?
`author`	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.
`description`	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?
`ms.author`	The author's Microsoft alias, *without* "@microsoft.com".	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. This value must be *one* employee or *one* approved Microsoft group alias. Don't add more than one alias. To use a group alias, you must first have it approved and added to the ms.author allowlist. The `ms.author` value can't be a user-created friendly email name, like `Jonathan.Dough`. The value must be the shorter email name that the company assigned to the person. Vendor (v-dash) aliases aren't recommended for `ms.author` because frequent turnover results in invalid values. When the ownership of the article changes, you should update this alias value promptly in the article. In general, `ms.author` should refer to the same person as the `author` value, unless `ms.author` is a group alias, in which case you'll need to assign an employee with a GitHub account as the owner in the `author` field.
`ms.date`	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.
`ms.service` *or* `ms.prod`	The service or product. Use one or the other, never both.	Used for issue triage and reporting. Generally, use `ms.service` for cloud applications and use `ms.prod` for on-premises servers and applications. If you need help with this entry or have any questions, contact Docs Metadata Management. Depending on the product or service, `ms.prod` might take `ms.technology` as a child value and `ms.service` might take `ms.subservice`. For available value pairs, see the ms.prod and ms.service allowlists.

FIELD	VALUE	WHY?
`ms.topic`	The type of content.	See the quick reference section below for information about common topic types. See the ms.topic allowlist for the complete list of valid values. Usually one of the following values: `article`, `conceptual`, `contributor-guide`, `interactive-tutorial`, `overview`, `quickstart`, `reference`, `sample`, `tutorial`.
`title`	The page title.	It's the most important metadata for SEO. For more information, see SEO: Tips for writing titles.

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
`archived`	Use in archived files only. Archived content stays on the site. A banner announces it as archived documentation.
`article`	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. **Note**: In the past, `article` was used for "any general article". The new guidance is to use `conceptual` in that case.

MS.TOPIC VALUE	DESCRIPTION
`conceptual`	Any content linked in a ToC that *isn't* marked in metadata as a: • Quickstart • Tutorial • Sample • Reference • Overview • *any other article type in this table* We consider this article type as the **default** article type. If you're uncertain which type to choose, use this value.
`contributor-guide`	Any content in a content contributor guide.
`end-user-help`	Content aimed at end users that provides resolution on a specific issue.
`error-reference`	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.
`github-samples`	Used in Docs Samples autogenerated files only. Don't manually add to markdown content.
`how-to`	Any procedural content that isn't a Quickstart or Tutorial. Located under the **How-to guides** node of the ToC.
`hub-page`	Use in hub page files only.
`include`	Use in [!INCLUDE] files only. It's required so Docs Build, validation, and other tools to work correctly.
`interactive-tutorial`	Any interactive content, like TripleCrown, Hands-On Labs, or articles with Content & Learning-supported interactive learning elements.
`kb-support`	Use *only* 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 `troubleshooting` value.
`landing-page`	Use *only* in landing page files.
`language-reference`	Set this value globally in *docfx.json* to identify a docset as autogenerated unmanaged API reference (Java, Node.js, Python, Azure CLI) or as authored API ref articles (C++).
`managed-reference`	Use in *docfx.json* 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.

MS.TOPIC VALUE	DESCRIPTION
`overview`	Overview or "User Guide" articles. These articles live under an **Overview** node in a ToC, or are high-level conceptual articles in a "New User" or "Onboarding" guide.
`quickstart`	Articles placed under a **Quickstart** node in a ToC. These articles *must* follow the specific guidelines and template for a Quickstart document.
`reference`	Any authored (read: *not* autogenerated) content placed under a **Reference** node in a ToC. For autogenerated API reference, see `language-reference` and `managed-reference`. For the specific cases of Visual Studio and .NET error reference pages, see `error-reference`.
`retired`	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.
`sample`	Any authored (read: *not* autogenerated) content placed under a **Samples** or **Examples** node in a ToC.
`tutorial`	Articles placed under a **Tutorials** node in a ToC. These articles *must* follow the specific guidelines and template for a Tutorial document.
`troubleshooting`	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. Knowledge Base (KB) articles owned by the CSS support organization don't use this value. They use `kb-support` instead.

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?
`ms.collection`	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?
`ms.custom`	**For writer or team use only.** 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. Example: `ms.custom: "experiment1, content_reporting, all_uwp_docs, CI_Id=101022"` **Character limit: The maximum string value length is 125 characters**.	`ms.custom` is a custom field that writers can use to track special projects or a subset of content. The Developer Experience docs team uses `devx-track-*` to track article groupings across services and repositories. Contact Barbara Kess if you have any questions.
`ms.reviewer`	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.
`ms.subservice`	The more granular service to which the content belongs. Only use `ms.subservice` if you're also using `ms.service`	`ms.subservice` by itself isn't valid metadata. The author must associate it with a parent `ms.service` value. This attribute is a way to drill down further in reporting for a given `ms.service`. See the ms.service allowlist for valid pairs.
`ms.technology`	The technology to which the content belongs. Only use `ms.technology` if you're also using `ms.prod`.	`ms.technology` by itself isn't valid metadata. The author must associate it with a parent `ms.prod` value. This attribute is a way to drill down further in reporting for a given `ms.prod`. In **Exchange**, for example, you could have `ms.prod: exchange-server-itpro` that applies to all evergreen Exchange Server articles, and beneath that an `ms.technology: high-availability` 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 ms.prod allowlist for valid pairs.

FIELD	VALUE	WHY?
`ROBOTS`	`NOINDEX`, `UNFOLLOW`	Use ROBOTS in your metadata section to prevent the build and publishing process from showing content on search pages. When you want to use `ROBOTS` (and yes, it's all capitalized, even though other metadata tags aren't): - Add `ROBOTS: NOINDEX` to your metadata section. - `NOINDEX` causes the asset to not show up in search results. - Use `NOFOLLOW` 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. - Customers who have a direct link to assets that have `NOINDEX` and `NOFOLLOW` can still view the content, but they can't find that content using search. - As a best practice, don't use `ROBOTS` without a value like this `ROBOTS:`.
`contributors_to_exclude`	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 *docfx.json* file.
`ms.localizationpriority`	Indicate a need for human translation (as opposed to machine translation) by assigning a value of `high`. A value of `medium` 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 need human translation.
`social_image_url`	Specifies an image URL that's used as the source for the open graph image (`og:image`) metadata on the page. It should be an absolute URL on docs, including the locale (so starting with `https://docs.microsoft.com/en-us/`) 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.	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 *only* be output on the page if you've also specified a `description` value (listed in the required metadata section earlier in this article). **If this metadata isn't specified, and the author includes a description, the site will automatically output the Microsoft logo as the open graph image value.**
`no-loc`	A list of words in the article that the Localization team shouldn't translate (localized). For more information, see Identify content to NOT be localized.	

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. Instead, they are injected in the extraction tool config file and are 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 APIScan. If source content has APIScan metadata, the author *must* 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.
`is_archived`	`true`	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."
`is_retired`	`true`	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."
`ms.author`	`Archiveddocs`	Indicates that there's no active author because the content isn't maintained.
`ms.date`	The extraction tool will extract the Last updated date from MTPS as the `ms.date` value. Make sure content has an `ms.date` value and *isn't* from `updated_at` metadata.	The date when the author last reviewed the content and confirmed its accuracy and relevance.

FIELD	VALUE	WHY?
`ms.prod`	Upon creating the repo, the Engagement team will provide the string value into the extraction tool config file after discussing with the Product team.	
`ms.topic`	`archived` or `retired`	Indicates the status of the content.
`ms.translationtype`	`MT`	Shows a slightly different disclaimer to inform user content is machine translated.
`ROBOTS`	`NOINDEX`, `NOFOLLOW`	Content shouldn't be discoverable in Search so it's marked `NOINDEX`. Add `NOFOLLOW` only when you've archived an entire content set.
`title`	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?
`author`	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.
`ms.author`	The author's Microsoft alias.	Identifies the author by Microsoft alias. Used for reporting and other Microsoft-internal contact.
`ms.date`	A date in MM/DD/YYYY format.	Indicates the last time the file was substantially edited or guaranteed fresh.
`ms.service` (and `ms.subservice`, if used) OR `ms.prod` (and `ms.technology`, if used)	The service or product the included file applies to.	Used for issue triage and reporting.
`ms.topic`	*Always* `include`.	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

7/7/2021 • 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.

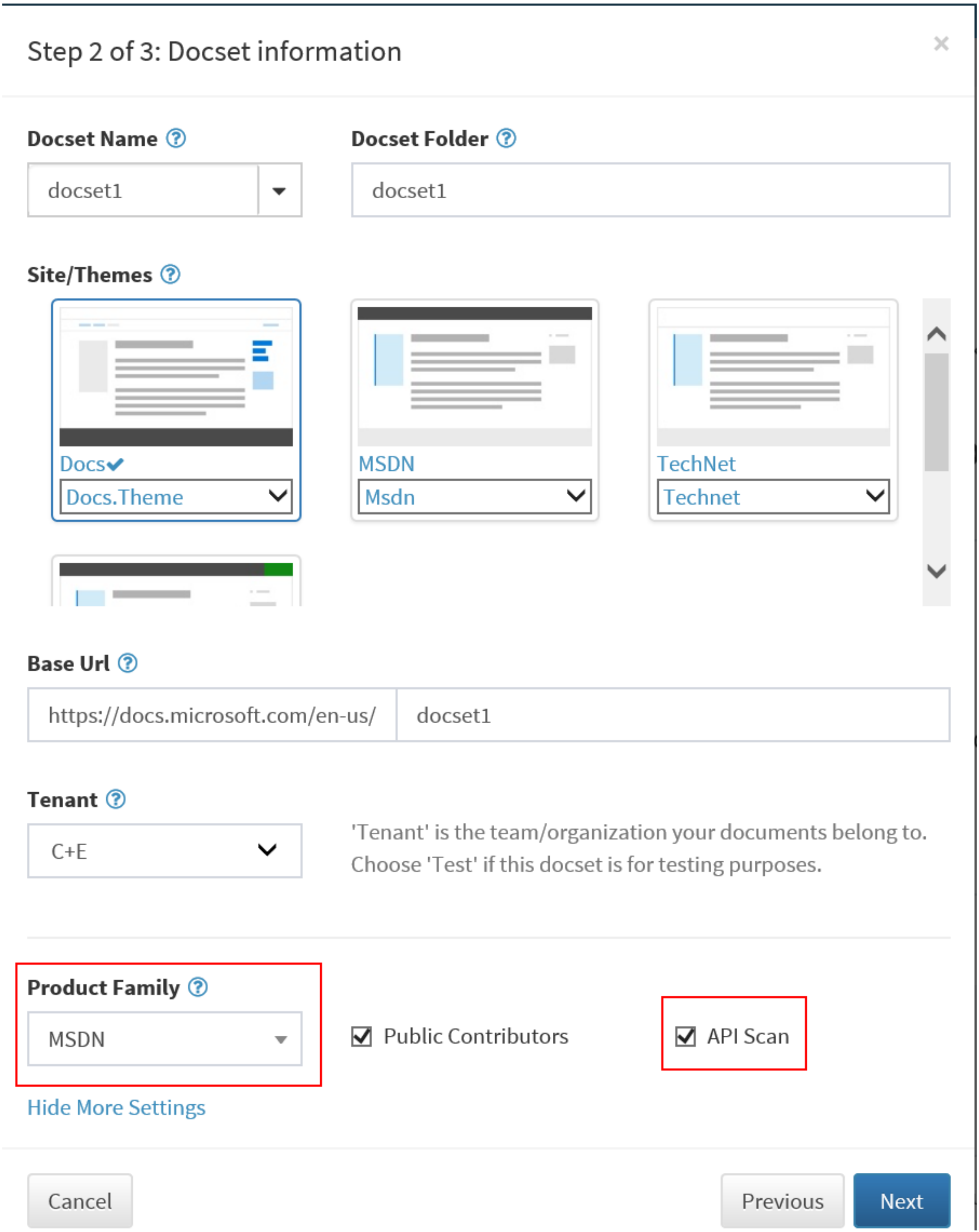
Configuring the docset for APIScan

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.

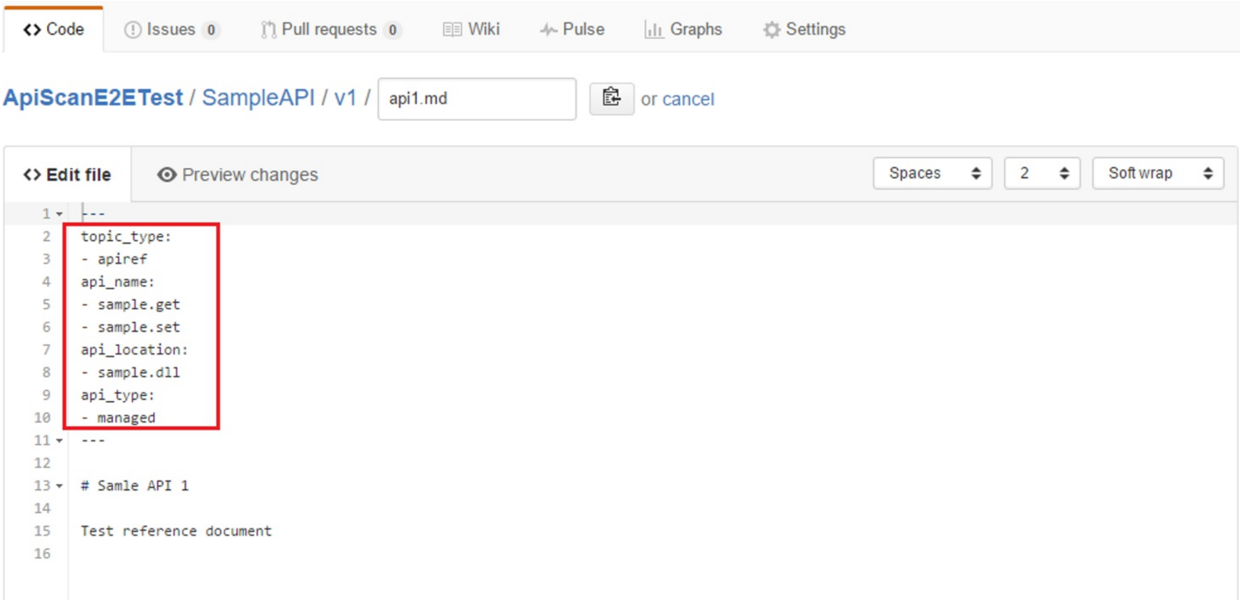


The mref build system applies the following metadata:

- **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.



- `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:

dooriya Update docfx.json

1 contributor

33 lines (32 sloc) | 499 Bytes

```
{
  "build": {
    "content": [
      {
        "files": [
          "**/*.md"
        ],
        "exclude": [
          "**/obj/**"
        ]
      }
    ],
    "resource": [
      {
        "files": [
          "**/*.png",
          "**/*.jpg"
        ],
        "exclude": [
          "**/obj/**"
        ]
      }
    ],
    "globalMetadata": {
      "product": ["SDK10", "Sample"]
    },
    "template": [
      "op.html"
    ],
    "dest": "SampleAPI/v1"
  }
}
```

How to request metadata changes

7/7/2021 • 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.

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 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.

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.

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.

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 DevRelIA@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 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 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.

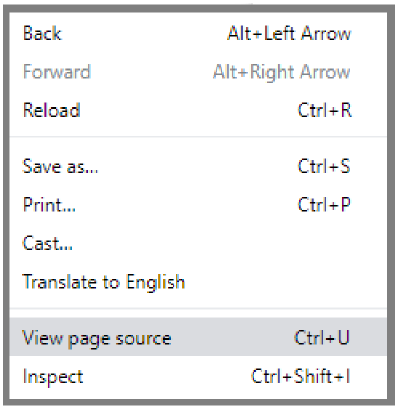
Find the author of an article via metadata

6/24/2021 • 2 minutes to read

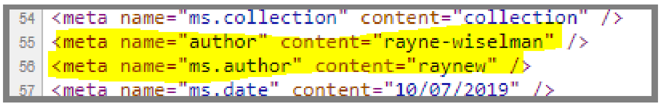
Internal Docs users at Microsoft sometimes want to find the author of an article to ask about the subject matter without suggesting edits to the article. Because author metadata can be set in multiple ways, the most reliable way to find the current author is to check the metadata on the published page.

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.



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

7/7/2021 • 81 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	ASP.NET

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
powerappsfl	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-ondemand-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-forms	Microsoft Forms
		office-kaizala	Kaizala
		office-lync-server	Lync Server
		office-onedrive	OneDrive
		office-onenote	OneNote

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		office-outlook	Outlook
		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	Microsoft 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

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		power-virtual-agents	Power Virtual Agents
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)

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
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.

SLUG	LABEL	DEFINITION
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.
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.

SLUG	LABEL	DEFINITION
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.
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.
support-engineer	Support Engineer	People that serve as frontline technical resources for Microsoft's customers and partners via phone, email, or web.
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
		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

PARENT SLUG	PARENT LABEL	CHILD SLUG	CHILD LABEL
		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
		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.

For more details, see the full taxonomy Detail View.

MS.AUTHOR DL
amlstudiodocs
apimpm

MS.AUTHOR DL
archiveddocs
azfuncdf
azurespheredocs
betafred
daxcpft
dotnetcontent
hisdocs
hvdev
msedgedevrel
msgraphdocsvteam
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 DevRelIA@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
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

SLUG	LABEL
ruby	Ruby
rust	Rust
spark-scala	Spark Scala
swift	Swift
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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		sharepoint-framework	SharePoint Framework
		tabs	Tabs
		vba	VBA
adaptive-cards	Adaptive Cards	(empty)	(empty)
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	ASP.NET	(empty)	(empty)
asp.net-4.0	ASP.NET 4.0	(empty)	(empty)
asp.net-4.5.1	ASP.NET 4.5.1	(empty)	(empty)
asp.net-5.0	ASP.NET 5.0	(empty)	(empty)
asp.net-mvc-3.0	ASP.NET MVC 3.0	(empty)	(empty)
asp.net-mvc-5.0	ASP.NET MVC 5.0	(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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		ms-bot-framework	Microsoft Bot Framework
		tabs	Tabs
calendar	Outlook	(empty)	(empty)
		microsoft-graph	Microsoft Graph
call	Non Product Specific	(empty)	(empty)
		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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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)
		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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
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
d365ce-op	Dynamics 365 Customer Engagement (on-premises)	(empty)	(empty)
data-access	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
data-connect	Microsoft Graph data connect	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		csharp-spec	C# Language Spec
		csharp-whats-new	C# what?s new
dotnet-data	.NET Data Access	(empty)	(empty)
		dotnet-ado	.NET ADO
		dotnet-ef	.NET Entity Framework
		dotnet-sql-client	.NET SQL Client
dotnet-data-prep	Microsoft Data Prep .NET SDK	(empty)	(empty)
dotnet-desktop	.NET Desktop	(empty)	(empty)
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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)
dynamics-rmspos	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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
		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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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)
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

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
kinect-dk	Azure Kinect	(empty)	(empty)
learning-aspnetcore	MSFT Learn: ASP.NET Core	(empty)	(empty)
learning-azure	MSFT Learn: Azure	(empty)	(empty)
learning-bonsai	Bonsai	(empty)	(empty)
		machine-teaching	Machine Teaching
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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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
		mdb	Defender for Business

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		mde	Defender for Endpoint
		mdep1	Defender for Endpoint Plan 1
		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
microsoft-365-business	Microsoft 365 Business	(empty)	(empty)
microsoft-365-enterprise	Microsoft 365 Enterprise	(empty)	(empty)
microsoft-365-lighthouse	Microsoft 365 Lighthouse	(empty)	(empty)
		admin	admin
		microsoft-graph	Microsoft Graph
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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		tabs	Tabs
		vba	VBA
multi-tenant-management	Multi-tenant management	(empty)	(empty)
		microsoft-graph	Microsoft Graph
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
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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

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	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
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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		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
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-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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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-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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		webhooks	Microsoft Graph webhooks
online-meeting	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
on-premises-data-gateway	On Premises Data Gateay	(empty)	(empty)
		microsoft-graph	Microsoft Graph
open-extensions	Non Product Specific	(empty)	(empty)
		microsoft-graph	Microsoft Graph
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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
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

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
powerpoint	PowerPoint	(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
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		powershell-gallery	Gallery
		powershell-jea	PowerShell JEA
		powershell-wmf	Windows Management Framework
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
		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

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
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
		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.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
search-server-2010	Search Server 2010	(empty)	(empty)
security	Security	(empty)	(empty)
		microsoft-graph	Microsoft Graph
seller-center	Seller Center	(empty)	(empty)
service-communications	Service health and communications	(empty)	(empty)
		microsoft-graph	Microsoft Graph
sharepoint	SharePoint	(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
		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
		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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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-busines-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
		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.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
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
		backup-restore	backup and restore
		big-data-cluster	Big Data Clusters
		clr	clr
		configuration	configuration
		connectivity	connectivity
		database-mirroring	database mirroring

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		migration-guide	Migration Guide
		native-client	native client
		performance	performance
		performance-monitor	performance monitor
		polybase	Polybase
		profiler	profiler

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		filestream	filestream
		high-availability	high availability
		in-memory-oltp	In-memory OLTP
		install	install
		integration-services	Integration Services (SSIS)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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
		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
		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
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
		windows	Windows
surface-hub	Surface hub	(empty)	(empty)
		windows	Windows
sway	Sway	(empty)	(empty)
		uri-scheme	URI scheme
system-center	System Center	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
		apps	Apps
		app-types	App types
		audio-video-and-camera	Audio, video, and camera
		communication	Communication
		data-and-files	Data and files
		deployment	Deployment

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		app-types	App types
		audio-video-and-camera	Audio, video, and camera
		communication	Communication
		data-and-files	Data and files
		deployment	Deployment
		design	Design

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		bf-bot-builder	Bot Builder
		connectors	Connectors
		content-pack	Content pack
		data-visualizations	Data visualizations
		fast-search	FAST search
		file-format	File format

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		shapesheet	Shapesheet
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		msbuild	MS Build
		office-development	Office Development
		sharepoint-development	SharePoint Development
		vs-ai-tools	AI Tools
		vs-azure	Azure Development Tools
		vs-container-tools	Container 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
		vs-python	Python
		vs-rtvs	R
		vs-unity-tools	Unity Tools
		vs-workflow-designer	Workflow Designer
		vs-xaml-tools	XAML Tools

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		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
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
		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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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	Accessbility Insights
windows-accessibility-tools	Windows Accessibility Tools	(empty)	(empty)
		accessibility-insights	Accessbility Insights
windows-azure-pack-for-windows-server	Windows Azure Pack	(empty)	(empty)
windows-client	Windows Client	(empty)	(empty)
		windows-client-active-directory	Active Directory
		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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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)
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)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		cfu	cfu
		dashboard	dashboard
		debugger	debugger
		devapps	devapps
		develop	develop
		devtest	devtest
		display	display
		driversecurity	driversecurity
		gettingstarted	gettingstarted
		gnss	gnss

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		gpio	gpio
		gpiobtn	gpiobtn
		hid	hid
		ieee	ieee
		ifs	ifs
		image	image
		install	install
		kernel	kernel
		mobilebroadband	mobilebroadband
		multifunction	multifunction
		netcx	netcx
		network	network
		nfc	nfc
		parports	parports
		partnerapps	partnerapps
		pci	pci
		pcmcia	pcmcia
		portable	portable
		pos	pos
		powermeter	powermeter
		print	print
		samples	samples
		sd	sd
		sensors	sensors
		serports	serports

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		sfu	sfu
		smartcard	smartcard
		spb	spb
		storage	storage
		stream	stream
		taef	taef
		troubleshooting	troubleshooting
		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)
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

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		remote-desktop-services	Remote Desktop Services
		security	Security
		storage	Storage
		storage-spaces	Storage Spaces
		troubleshoot	Troubleshoot

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		windows-server-rds	RDS
		windows-server-resources	Resources
		windows-server-sdn	SDN
		windows-server-security	Windows Security

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		techgroup-management-and-automation	Management/Automation
		techgroup-networking	Networking
		techgroup-security	Security
		techgroup-storage	Storage

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		techgroup-security	Security
		techgroup-storage	Storage
windows-server-2012-R2-and2012	z_Windows Server 2012 R2 and 2012	(empty)	(empty)

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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	ASP.NET
		boot-event-collector	Boot Event Collector
		cimwin32	CIMWIN32
		compute-cluster-pack	Compute Cluster Pack
		data-deduplication	Data Deduplication

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		internet-protocol-address-management	Internet Protocol Address Management
		iscsi-target	ISCSI Target

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		windows-server-backup	Windows Server Backup
		work-folders	Work Folders

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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	Networkign - DA
		networking-dhcp	Networking - DHCP
		networking-dns	Networking - DNS

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		storage-deduplication	Storage - Deduplication
		storage-dfsn	Storage - DFSN

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		storage-dfsr	Storage - DFSR
		storage-failover-clustering	Storage - Failover Clustering
		storage-file-systems	Storage - File Systems
		storage-fsrm	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)
windows-subsystem-for-linux	Windows Subsystem for Linux	(empty)	(empty)
		system-utilities	System Utilities

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
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
		sharepoint-framework	SharePoint Framework
		sharepoint-webhooks	SharePoint webhooks

MS.PROD SLUG	MS.PROD LABEL	MS.TECHNOLOGY SLUG	MS.TECHNOLOGY LABEL
		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
		xamarin-tools	Xamarin Tools
xboxlive	Xbox Live	(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
		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-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.

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
		app-proxy	App proxy
		authentication	Authentication

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		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
active-directory-b2c	Active Directory B2C	(empty)	(empty)
active-directory-ds	Active Directory Domain Services	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
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-windows	Windows
app-service-api	API Apps	(empty)	(empty)
app-service-mobile	Mobile Apps	(empty)	(empty)
app-service-web	Web Apps	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
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
attestation	Azure Attestation	(empty)	(empty)
		attestation	Azure Attestation
automation	Automation	(empty)	(empty)
		change-inventory-management	Change and Inventory
		dsc	Desired State Configuration (DSC)
		process-automation	Process Automation
		shared-capabilities	Shared capabilities
		update-management	Update Management
autonomous-systems	Autonomous Systems	(empty)	(empty)
avere-vfxt	Avere	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
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-kubernetes	Azure Arc for Kubernetes
		azure-arc-servers	Azure Arc for Servers
azure-arc-data	Azure Arc Data	(empty)	(empty)
		azure-arc-postgres-hyperscale	Azure Arc Postgres Hyperscale
		azure-arc-sql-managed-instance	Azure Arc SQL Managed Instance
		azure-arc-sql-server	Azure Arc SQL Server
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)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		databricks-sql-analytics	Azure Databricks SQL Analytics
azure-dev-spaces	Azure Dev Spaces	(empty)	(empty)
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

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		alerts	Alerts
		application-insights	Application Insights
		autoscale	Autoscale
		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

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		qdk	Quantum Development Kit
		qsharp-guide	Qsharp user guide
azure-redhat-openshift	Azure Red Hat OpenShift	(empty)	(empty)
		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
		video-analyzer-subservice	Video Analyzer subservice
azure-vmware	Azure VMWare Solution	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
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)
		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

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		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-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

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
bing-webmaster	Bing Webmaster	(empty)	(empty)
		bing-webmaster-api	Bing Webmaster API
biztalk-services	BizTalk Services	(empty)	(empty)
blueprints	Azure Blueprints	(empty)	(empty)
bonsai	Bonsai	(empty)	(empty)
		bonsai-sa	Bonsai Solution Accelerator
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

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		plan	plan
		ready	ready
		secure	secure
		strategy	strategy
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

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-speech	Bing Speech
		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)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		consumer	consumer
container-instances	Container Instances	(empty)	(empty)
		confidential-computing	Confidential Computing
container-registry	Container Registry	(empty)	(empty)
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)
		ahb	Azure Hybrid Benefits
		billing	Billing
		cloudyn	Cloudyn
		common	Common
		cost-management	Cost Management
		enterprise	Enterprise Agreement
		microsoft-customer-agreement	Microsoft Customer Agreement
		reservations	Reservations
crm-online	Dynamics 365 (CRM)	(empty)	(empty)
csp	Cloud Solution Provider	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
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
		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)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
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)
deployment-manager	Deployment Manager	(empty)	(empty)
devtest-lab	DevTest Labs	(empty)	(empty)
digital-twins	IoT Azure Digital Twins	(empty)	(empty)
dms	Data Migration Service	(empty)	(empty)
		connectivity	Connectivity
		migration-guide	Migration Guide
dns	DNS	(empty)	(empty)
documentdb	DocumentDB	(empty)	(empty)
dynamics365-accountant	Accountants	(empty)	(empty)
dynamics-365-ai	Dynamics 365 AI	(empty)	(empty)
dynamics365-business-central	Dynamics 365 Business Central	(empty)	(empty)
dynamics-365-commerce	Dynamics 365 Commerce	(empty)	(empty)
dynamics-365-connected-store	Connected Store	(empty)	(empty)
dynamics-365-cross-app	Dynamics 365 cross-applications	(empty)	(empty)
dynamics-365-customerservice	Dynamics 365 for Customer Service	(empty)	(empty)
		channel-integration-framework	Channel Integration Framework
		unified-service-desk	Unified Service Desk

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
dynamics-365-customervoice	Dynamics 365 Customer Voice	(empty)	(empty)
dynamics-365-field-service	Dynamics 365 Field Service	(empty)	(empty)
		common-scheduler	Common scheduler
		connected-field-service	Connected field service
		field-service-mobile	Field service mobile
		resource-scheduling-optimization	Resource scheduling optimization
dynamics-365-finance	Dynamics 365 Finance	(empty)	(empty)
dynamics365-financials	Dynamics 365 for Financials	(empty)	(empty)
dynamics-365-fraud-protection	Dynamics 365 Fraud Protection	(empty)	(empty)
dynamics-365-guides	Dynamics 365 Guides	(empty)	(empty)
dynamics-365-human-resources	Dynamics 365 Human Resources	(empty)	(empty)
dynamics-365-import-tool	Dynamics 365 Import Tool	(empty)	(empty)
dynamics-365-industry	Dynamics 365 Industry Solutions	(empty)	(empty)
		return-to-school	Return to School
		return-to-work	Return to Work
		vaccination-management	Vaccination Management
dynamics-365-intelligent-order-management	Dynamics 365 Intelligent Order Management	(empty)	(empty)
dynamics-365-linkedinconnector	Dynamics 365 Connector for LinkedIn	(empty)	(empty)
dynamics-365-marketing	Dynamics 365 for Marketing	(empty)	(empty)
dynamics-365-productvisualize	Dynamics 365 Product Visualize	(empty)	(empty)
dynamics-365-project-operations	Dynamics 365 Project Operations	(empty)	(empty)
dynamics-365-retail	Dynamics 365 for Retail	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
dynamics-365-sales	Dynamics 365 for Sales	(empty)	(empty)
dynamics-365-supply-chain	Dynamics 365 Supply Chain Management	(empty)	(empty)
dynamics-365-talent	Dynamics 365 for Talent	(empty)	(empty)
dynamics-ax-applications	Dynamics 365 for Operations Applications	(empty)	(empty)
dynamics-ax-platform	Dynamics 365 for Operations Platform	(empty)	(empty)
dynamics-lifecycle-services	Dynamics Lifecycle Services	(empty)	(empty)
dynamics-nav-2018	Dynamics NAV 2018	(empty)	(empty)
event-grid	Event Grid	(empty)	(empty)
		iot-edge	Event Grid on IoT Edge
		kubernetes	Event Grid on Kubernetes
event-hubs	Event Hubs	(empty)	(empty)
exchange-online	Exchange Online	(empty)	(empty)
exchange-powershell	Exchange PowerShell	(empty)	(empty)
expressroute	ExpressRoute	(empty)	(empty)
firewall	Azure Firewall	(empty)	(empty)
		networking	Networking
firewall-manager	Azure Firewall Manager	(empty)	(empty)
forms-pro	Forms Pro	(empty)	(empty)
fraud-protection	Fraud Protection	(empty)	(empty)
frontdoor	Azure Front Door Service	(empty)	(empty)
fxt-edge-filer	Azure FXT Edge Filer	(empty)	(empty)
gamification	Gamification	(empty)	(empty)
genomics	Microsoft Genomics	(empty)	(empty)
germany	Azure Germany	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
globalization	Globalization	(empty)	(empty)
governance	Azure Governance	(empty)	(empty)
groove-music	Groove Music	(empty)	(empty)
guest-configuration	Guest Configuration	(empty)	(empty)
guidance	Guidance	(empty)	(empty)
hdinsight	HDInsight	(empty)	(empty)
healthbot	Health Bot	(empty)	(empty)
healthcare-apis	Microsoft Healthcare APIs	(empty)	(empty)
		fhir	FHIR
		iomt	IoMT connector
hotel-ads	Hotel Ads API	(empty)	(empty)
hotel-ads-hint-message	Hotel Ads Hint Message	(empty)	(empty)
hotel-ads-hotel-feed	Hotel Feed	(empty)	(empty)
hotel-ads-pos-feed	Hotel Ads POS Feed	(empty)	(empty)
hotel-ads-query-control-message	Hotel Ads Query Control Message	(empty)	(empty)
hotel-ads-query-message	Hotel Ads Query Message	(empty)	(empty)
hotel-ads-transaction-message	Hotel Ads Transaction Message	(empty)	(empty)
hpc-cache	HPC Cache	(empty)	(empty)
identity	Azure Active Directory	(empty)	(empty)
		verifiable-credentials	Verifiable Credentials
industrial-iot	Industrial IoT	(empty)	(empty)
industry	Industry	(empty)	(empty)
		financial	Financial Cloud
		healthcare	Healthcare Cloud

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		manufacturing-cloud	Manufacturing Cloud
		nonprofit	Nonprofit Cloud
		retail	Retail Cloud
information-protection	Microsoft Information Protection	(empty)	(empty)
		aip	Azure Information Protection
		aiplabels	Azure Information Protection policy
		analytics	Analytics (central reporting)
		azurerms	Azure RMS
		connector	RMS connector
		doctrack	Document tracking
		fci	Windows Server FCI
		hyok	HYOK
		kms	Key management (BYOK & MS-managed)
		labelmigrate	Label Migration
		migration	AD RMS migration
		mip	Microsoft Information Protection
		mipsdk	Microsoft Information Protection SDK
		prereqs	Requirements
		scanner	Scanner
		v1client	Classic client
		v2client	Unified labeling client
intelligent-systems	Intelligent Systems	(empty)	(empty)
internet-analyzer	Azure Internet Analyzer	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
internet-peering	Internet Peering	(empty)	(empty)
invoicing	Invoicing	(empty)	(empty)
iot-accelerators	IoT Solution Accelerators	(empty)	(empty)
iot-central	IoT Central	(empty)	(empty)
		iot-central-retail	IoT Central Retail
iot-develop	Azure IoT Device and Application Development	(empty)	(empty)
iot-dps	IoT Device Provisioning	(empty)	(empty)
iot-edge	IoT Edge	(empty)	(empty)
iot-fundamentals	IoT Fundamentals	(empty)	(empty)
iot-hub	IoT Hub	(empty)	(empty)
iot-hub-device-update	Device Update for IoT Hub	(empty)	(empty)
iot-pnp	IoT PNP	(empty)	(empty)
iot-suite	IoT Suite	(empty)	(empty)
jenkins	Jenkins on Azure	(empty)	(empty)
kaizala	Kaizala admin	(empty)	(empty)
key-vault	Key Vault	(empty)	(empty)
		certificates	Certificates
		general	General
		keys	Keys
		managed-hsm	Managed HSM
		secrets	Secrets
kusto	Kusto Query Language	(empty)	(empty)
lab-services	Lab Services	(empty)	(empty)
		classroom-labs	Classroom Labs
		custom-labs	Custom Labs

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		trial-labs	Trial Labs
lighthouse	Lighthouse	(empty)	(empty)
linkedin	LinkedIn	(empty)	(empty)
linux-vms	Virtual Machines	(empty)	(empty)
load-balancer	Load Balancer	(empty)	(empty)
log-analytics	Log Analytics	(empty)	(empty)
logic-apps	Logic Apps	(empty)	(empty)
m365-md	Microsoft 365 Managed Desktop	(empty)	(empty)
machine-learning	Machine Learning	(empty)	(empty)
		core	Core
		data-prep	Data prep
		studio-classic	Studio Classic
		team-data-science-process	Team Data Science Process
managed-applications	Managed Applications	(empty)	(empty)
managed-hsm	Managed HSM	(empty)	(empty)
managed-instance-apache-cassandra	Cassandra Managed Instance	(empty)	(empty)
mariadb	Azure Database for MariaDB	(empty)	(empty)
		migration-guide	Migration Guide
marketplace	Marketplace	(empty)	(empty)
		partnercenter-enroll	PartnerCenterEnroll
		partnercenter-marketplace-publisher	PartnerCenterMarketplacePublisher
		partnercenter-offers	PartnerCenterOffers
marketplace-customer	Commercial marketplace customer	(empty)	(empty)
mdm	Dynamics Marketing	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
media-services	Media Services	(empty)	(empty)
		lva-edge	Live Video Analytics on IoT Edge
		media-player	Azure Media Player
		rest-v2	Rest V2
		rest-v3	Rest V3
		video-indexer	Azure Media Services Video Indexer
mem	Microsoft Endpoint Manager	(empty)	(empty)
		fundamentals	Fundamentals
microsoft-365-education	Microsoft 365 Education	(empty)	(empty)
microsoft-academic-services	Microsoft Academic Services	(empty)	(empty)
microsoft-defender-for-identity	Microsoft Defender for Identity	(empty)	(empty)
microsoft-intune	Microsoft Intune	(empty)	(empty)
		apps	Apps
		configuration	Configuration
		developer	Developer
		education	Education
		end-user	End user help
		enrollment	Enrollment
		fundamentals	Fundamentals
		protect	Protect
		remote-actions	Remote actions
mobile-center	Mobile Center	(empty)	(empty)
mobile-engagement	Mobile Engagement	(empty)	(empty)
mobile-services	Mobile Services	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
movere	Movere	(empty)	(empty)
		movere	Movere
mse	Social Engagement	(empty)	(empty)
mssearch	Microsoft Search	(empty)	(empty)
msteams	Microsoft Teams	(empty)	(empty)
multi-factor-authentication	Multi-Factor Authentication	(empty)	(empty)
multiple	Multiple	(empty)	(empty)
mysql	Azure Database for MySQL	(empty)	(empty)
		flexible-server	Flexible Server
		migration-guide	Migration Guide
mysql-database	MySQL	(empty)	(empty)
		single-server	Single Server
mysql-database-mc	MySQL-Mooncake	(empty)	(empty)
nat	NAT Gateway	(empty)	(empty)
network-function-manager	Network Function Manager	(empty)	(empty)
network-watcher	Network Watcher	(empty)	(empty)
notification-hubs	Notification Hubs	(empty)	(empty)
o365-administration	Office 365 Administration	(empty)	(empty)
o365-proplus-itpro	Office 365 ProPlus (IT Pro)	(empty)	(empty)
O365-seccomp	O365 Security and Compliance	(empty)	(empty)
o365-solutions	Office 365 Solutions	(empty)	(empty)
odata	OData	(empty)	(empty)
office-scripts	Office Scripts	(empty)	(empty)
one-drive	One Drive	(empty)	(empty)
open-datasets	Azure Open Datasets	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
orbital	Orbital	(empty)	(empty)
partner-dashboard	Partner Center	(empty)	(empty)
		partnercenter-action-center	PartnerCenterActionCenter
		partnercenter-benefits	PartnerCenterBenefits
		partnercenter-billing	PartnerCenterBilling
		partnercenter-consumer	PartnerCenterConsumer
		partnercenter-csp	PartnerCenterCSP
		partnercenter-customers	PartnerCenterCustomers
		partnercenter-enroll	PartnerCenterEnroll
		partnercenter-incentives	PartnerCenterIncentives
		partnercenter-insights	PartnerCenterInsights
		partnercenter-membership	PartnerCenterMembership
		partnercenter-mpn	PartnerCenterMPN
		partnercenter-payouts	PartnerCenterPayouts
		partnercenter-pricing	PartnerCenterPricing
		partnercenter-referrals	PartnerCenterReferrals
		partnercenter-sdk	PartnerCenter SDK
		partnercenter-support	PartnerCenterSupport
partner-services	Partner Services	(empty)	(empty)
peering-service	Peering Service	(empty)	(empty)
postgresql	Azure Database for PostgreSQL	(empty)	(empty)
		flexible-server	Flexible Server
		hyperscale-citus	Hyperscale Citus
		migration-guide	Migration Guide
		single-server	Single Server

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
postgresql-database	PostgreSQL	(empty)	(empty)
powerapps	Power Apps	(empty)	(empty)
		canvas-developer	Canvas apps developer
		canvas-maker	Canvas apps maker
		common	Power Apps common content
		dataverse-developer	Dataverse developer
		dataverse-maker	Dataverse maker
		end-user	Power Apps end user
		guidance	Power Apps guidance
		mda-developer	Model-driven apps developer
		mda-maker	Model-driven apps maker
		mobile	Power Apps mobile
		pcf	Power Apps component framework
		portals	Power Apps portals
		sample-apps	Power Apps sample apps
		teams	Power Apps-Teams integration
		troubleshoot	Power Apps troubleshooting
power-automate	Power Automate	(empty)	(empty)
		bpf	Business process flow
		cloud-flow	Cloud flow
		common	Power Automate common content
		desktop-flow	Desktop flow
		Developer	Power Automate developer

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		guidance	Power Automate guidance
		Legacy	Power Automate legacy content
		process-advisor	Process Advisor
		teams	Power Automate-Teams integration
powerbi	Power BI	(empty)	(empty)
		pbi-collaborate-share	Collaborate & Share
		pbi-cross-product	Cross-product integration
		pbi-dataflows	Dataflows in Power BI
		pbi-data-sources	Data sources for Power BI
		pbi-deployment	Deployment pipelines in Power BI
		pbi-enterprise	Enterprise for Power BI
		pbi-explore	Explore Power BI
		pbi-fundamentals	Fundamentals for Power BI
		pbi-natural-language	Natural Language and Q&A
		pbi-reports-dashboards	Reports and dashboards in Power BI
		pbi-security	Security in Power BI
		pbi-transform-model	Transform, shape, and model in Power BI
		pbi-visuals	Visualizations in Power BI
		powerbi	Power BI all
		powerbi-admin	Power BI administration
		powerbi-consumer	Power BI Consumer
		powerbi-custom-visuals	Power BI Custom Visuals
		powerbi-desktop	Power BI Desktop

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		powerbi-developer	Power BI developer
		powerbi-eim	Power BI Enterprise Information Management
		powerbi-gateways	Power BI gateways
		powerbi-mobile	Power BI mobile
		powerbi-premium	Power BI Premium
		powerbi-report-server	Power BI Report Server
		powerbi-service	Power BI service
		powerbi-template-apps	Power BI Template Apps
		report-builder	Power BI Report Builder
power-bi-embedded	Power BI Embedded	(empty)	(empty)
power-platform	Power Platform	(empty)	(empty)
		admin	Power Platform admin
		alm	Power Platform ALM
		guidance	Power Platform guidance
		power-fx	Power Fx
powerquery	Power Query	(empty)	(empty)
power-virtual-agents	Power Virtual Agents	(empty)	(empty)
private-link	Azure Private Link	(empty)	(empty)
private-multi-access-edge-compute-mec	Azure private MEC	(empty)	(empty)
processrobot	ProcessRobot	(empty)	(empty)
product-insights	Product Insights	(empty)	(empty)
project-madeira	Financials	(empty)	(empty)
project-online	Project Online	(empty)	(empty)
project-operations	Project Operations	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
project-web	Project for the web	(empty)	(empty)
purview	Azure Purview	(empty)	(empty)
		purview-data-catalog	Azure Purview Data Catalog
pytorch-enterprise	PyTorch Enterprise	(empty)	(empty)
remoteapp	RemoteApp	(empty)	(empty)
remote-connect-tool	Exchange Remote Connectivity Analyzer Tool	(empty)	(empty)
resource-graph	Azure Resource Graph	(empty)	(empty)
resource-health	Resource Health	(empty)	(empty)
resource-move	resource-move	(empty)	(empty)
resource-mover	Resource Mover	(empty)	(empty)
rights-management	Rights Management	(empty)	(empty)
role-based-access-control	Role-Based Access Control	(empty)	(empty)
		conditions	Conditions
route-server	Route Server	(empty)	(empty)
rtos	Microsoft Azure RTOS	(empty)	(empty)
scheduler	Scheduler	(empty)	(empty)
school-data-sync	School Data Sync	(empty)	(empty)
search	Azure Search	(empty)	(empty)
security	Security	(empty)	(empty)
		disk-encryption	Azure Disk Encryption
		security-develop	Security and Develop
		security-fundamentals	Security-Fundamentals
security-center	Security Center	(empty)	(empty)
server-management	Server Management	(empty)	(empty)
service-bus	Service Bus	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
service-bus-messaging	Service Bus Messaging	(empty)	(empty)
service-bus-relay	Service Bus Relay	(empty)	(empty)
service-fabric	Service Fabric	(empty)	(empty)
service-fabric-mesh	Service Fabric Mesh	(empty)	(empty)
service-health	Service Health	(empty)	(empty)
sharepoint-online	SharePoint Online	(empty)	(empty)
sharepoint-pnp	SharePoint PnP	(empty)	(empty)
sharepoint-powershell	SharePoint PowerShell	(empty)	(empty)
signalr	SignalR Service	(empty)	(empty)
site-recovery	Site Recovery	(empty)	(empty)
		site-recovery	Site Recovery
site-reliability-engineering	Site Reliability Engineering	(empty)	(empty)
skype-for-business-online	Skype for Business Online	(empty)	(empty)
skypeforbusiness-powershell	Skype Powershell	(empty)	(empty)
spmt-powershell	SharePoint Migration Tool PowerShell	(empty)	(empty)
spring-cloud	Spring Cloud	(empty)	(empty)
sql-database	SQL Database	(empty)	(empty)
		backup-restore	Backup and restore
		business-continuity	business-continuity
		connect	Connect
		data-movement	Data Movement
		deployment-configuration	Deployment Configuration
		development	Development
		elastic-jobs	Elastic Jobs

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		elastic-pools	Elastic Pools
		high-availability	High Availability
		machine-learning	Machine Learning Services
		managed-instance	Managed Instance
		migration	Migration
		migration-guide	Migration guide
		monitor	Monitor
		operations	Operations
		performance	Performance
		replication	Replication
		scale-out	Scale Out
		scenario	Scenario
		security	Security
		service	Service
		service-overview	Service Overview
		single-database	Single Database
		sql-data-sync	SQL Data Sync
sql-database-edge	SQL Database Edge	(empty)	(empty)
		backup-restore	Backup and restore
		business-continuity	Business continuity
		data-movement	Data movement
		development	Development
		features	Features
		high-availability	High Availability
		machine-learning	Machine Learning

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		migrate	Migration
		monitor	Monitor
		performance	Performance
		security	Security
		service	Service
sql-data-warehouse	SQL Data Warehouse	(empty)	(empty)
		consume	Consume
		design	Design
		development	Development
		implement	Implement
		integration	Integration
		load-data	Load Data
		manage	Manage
		performance	Performance
		query	Query
		security	Security
		supportability	Supportability
		workload-management	Workload management
sql-db-mi	SQL DB and MI	(empty)	(empty)
		backup-restore	Backup and restore
		business-continuity	Business continuity
		connect	Connect
		data-movement	Data movement
		deployment-configuration	Deployment Configuration
		development	Development

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		features	Features
		high-availability	High Availability
		migrate	Migration
		monitor	Monitor
		performance	Performance
		security	Security
		service	Service
		service-overview	Service Overview
sql-edge	SQL Edge	(empty)	(empty)
		machine-learning	Machine Learning
sql-managed-instance	SQL Managed Instance	(empty)	(empty)
		backup-restore	Backup Restore
		business-continuity	business-continuity
		connect	Connect
		data-movement	Data Movement
		deployment-configuration	Deployment Configuration
		development	Development
		high-availability	High Availability
		instance-pools	Instance Pools
		machine-learning	Machine Learning Services
		migration	Migration
		migration-guide	Migration guide
		monitor	Monitor
		operations	Operations
		performance	Performance

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		replication	Replication
		scale-out	Scale Out
		security	Security
		service	Service
		service-overview	Service Overview
sql-server-stretch-database	SQL Server Stretch Database	(empty)	(empty)
		security	Security
		stretchDB	Stretch DB
sql-vms	Virtual Machines	(empty)	(empty)
staffhub-powershell	Staffhub Powershell	(empty)	(empty)
static-web-apps	App Service Static Web Apps	(empty)	(empty)
storage	Storage	(empty)	(empty)
		blobs	Blob storage
		common	Common
		data-lake-storage-gen2	Data Lake Storage Gen2
		disks	Disk Storage
		files	Azure Files
		partner	Partner integration
		queues	Queue storage
		tables	Table storage
storsimple	StorSimple	(empty)	(empty)
stream	Stream	(empty)	(empty)
stream-analytics	Azure Stream Analytics	(empty)	(empty)
synapse-analytics	Azure Synapse Analytics	(empty)	(empty)
		business-intelligence	Business Intelligence

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		cicd	Continuous Integration/Continuous Deployment
		data-explorer	Data Explorer
		machine-learning	Machine Learning
		metadata	Metadata
		monitoring	Monitoring
		overview	Overview
		pipeline	Pipeline
		purview	Purview
		security	Security
		spark	Spark
		sql	SQL
		sql-dw	SQL DW
		synapse-link	Synapse Link
		troubleshooting	Troubleshooting
		workspace	Workspace
teams-powershell	Microsoft Teams PowerShell	(empty)	(empty)
terraform	Terraform	(empty)	(empty)
time-series-insights	IoT Time Series Insights	(empty)	(empty)
traffic-manager	Traffic Manager	(empty)	(empty)
turbos	Turbos	(empty)	(empty)
universal-print	Universal Print	(empty)	(empty)
		connector	UPConnector
		fundamentals	CoreContent
		print-devices	OemDevices

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
uxstypo	Typography and UX Solutions	(empty)	(empty)
virtual-desktop	Azure Virtual Desktop	(empty)	(empty)
virtual-machines	Virtual Machines	(empty)	(empty)
		automanage	Azure Automanage
		automatic-extension-upgrade	Automatic Extension Upgrade
		automatic-guest-patching	Automatic Guest Patching
		automatic-image-patching	Automatic Image Patching
		autoscale	Autoscale
		azure-hybrid-benefit	Azure Hybrid Benefit
		baremetal-infrastructure	BareMetal Infrastructure
		benchmark	Benchmark
		billing	Billing
		boot-diagnostics	Azure Boot Diagnostics
		classic-to-arm-migration	Classic to ARM Migration
		cloud-foundry	Cloud Foundry
		cloud-init	Cloud-init
		confidential-computing	Azure Confidential Computing
		dedicated-hosts	Dedicated Hosts
		disks	Disks
		extensions	Extensions
		flexible-scale-sets	Flexible scale sets
		generation-2-vm	Generation 2 Virtual Machines
		hotpatch	Hotpatch
		hpc	HPC

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		image-builder	Image Builder
		imaging	Imaging
		mainframe-rehosting	Mainframe Rehosting
		maintenance-control	Maintenance Control
		metadata-service	Metadata Service
		monitoring	Monitoring
		networking	Networking
		openshift	OpenShift
		oracle	Oracle
		proximity-placement-groups	Proximity Placement Groups
		redhat	Redhat
		reserved-instances	Reserved Instances
		scheduled-events	Scheduled Events
		security	Security
		shared-image-gallery	Shared Image Gallery
		sizes	Sizes
		spot	Azure Spot Virtual Machines
		trusted-launch	Trusted Launch
		vm-sizes-compute	Compute Optimized VM Sizes
		vm-sizes-fpga	FPGA Optimized VM Sizes
		vm-sizes-general	General Purpose VM Sizes
		vm-sizes-gpu	GPU Optimized VM Sizes
		vm-sizes-hpc	High Performance Compute VM Sizes

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		vm-sizes-memory	Memory Optimized VM Sizes
		vm-sizes-storage	Storage Optimized VM Sizes
virtual-machine-scale-sets	Virtual Machine Scale Sets	(empty)	(empty)
		automatic-guest-patching	Automatic VM Guest Patching
		automatic-instance-repairs	Automatic Instance Repairs
		automatic-os-upgrade	Automatic-Os Upgrade
		autoscale	Autoscale
		availability	Availability
		azure-hybrid-benefit	Azure Hybrid Benefit
		dedicated-hosts	Dedicated Hosts
		disks	Disks
		extensions	Extensions
		hotpatch	Hotpatch
		hpc	HPC
		imaging	Imaging
		instance-protection	Instance Protection
		maintenance-control	Maintenance Control
		management	Management
		monitoring	Monitoring
		networking	Networking
		oracle	Oracle
		proximity-placement-groups	Proximity Placement Groups
		redhat	RedHat
		scale-in-policy	Scale-In Policy

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		security	Security
		shared-image-gallery	Shared Image Gallery
		sizes	Sizes
		spot	Spot
		terminate-notification	Terminate Notification
virtual-machines-linux	Linux Virtual Machines	(empty)	(empty)
		disks	Disk storage
		extensions	Extensions
		imaging	Imaging
		monitoring	Monitoring
		networking	Networking
		recovery	Recovery
		security	Security
		sizes	Sizes
		workloads	Workloads
virtual-machines-sap	Virtual machines SAP	(empty)	(empty)
		baremetal-sap	SAP HANA on BareMetal
virtual-machines-sql	SQL Virtual Machines	(empty)	(empty)
		backup	SQL VM Backup
		deployment	SQL VM Deployment
		hadr	SQL VM HADR
		management	SQL VM Management
		migration	SQL VM Migration
		migration-guide	SQL VM Migration Guide
		performance	SQL VM Performance

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		resource-manager	SQL VM Resource Manager
		rhel	Red Hat Enterprise Linux
		security	SQL VM Security
		service-management	SQL VM Service Management
		service-overview	Service Overview
		sql-vm	SQL VMs
		suse	SUSE Enterprise Linux
		ubuntu	Ubuntu
virtual-machines-windows	Windows Virtual Machines	(empty)	(empty)
		disks	Disk storage
		extensions	Extensions
		imaging	Imaging
		monitoring	Monitoring
		networking	Networking
		recovery	Recovery
		security	Security
		sizes	Sizes
		workloads	Workloads
virtual-network	Virtual Network	(empty)	(empty)
		ip-services	IP Services
		nat	Virtual Network NAT
virtual-wan	Virtual WAN	(empty)	(empty)
vmware-virtustream	Azure VMware Solution by Virtustream	(empty)	(empty)
vnf-manager	VNF- Manager	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
vpn-gateway	VPN Gateway	(empty)	(empty)
vs-appcenter	App Center	(empty)	(empty)
		app-center-analytics	Analytics
		app-center-auth	Auth
		app-center-build	Build
		app-center-data	Data
		app-center-diagnostics	Diagnostics
		app-center-distribute	Distribute
		app-center-general	General
		app-center-push	Push
		app-center-test	Test
		app-center-transition	Transition
vs-appcenter-sdk	App Center SDK	(empty)	(empty)
		app-center-analytics	Analytics
		app-center-auth	Auth
		app-center-build	Build
		app-center-data	Data
		app-center-diagnostics	Diagnostics
		app-center-distribute	Distribute
		app-center-general	General
		app-center-push	Push
		app-center-test	Test
		app-center-transition	Transition
web-app-container	Web App for Containers	(empty)	(empty)
web-application-firewall	Web Application Firewall	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
websites	Websites	(empty)	(empty)
web-sites	Websites	(empty)	(empty)
whiteboard-powershell	Whiteboard PowerShell	(empty)	(empty)
windows-vms	Virtual Machines	(empty)	(empty)
yammer	Yammer	(empty)	(empty)

ms.topic

The `ms.topic` value indicates the type of content.

To request a new ms.topic value, email docsmetamanager@microsoft.com.

For more details, see the full taxonomy Detail View.

SLUG
archived
article
best-practice
callback
checklist
class
conceptual
contributor-guide
design-pattern
end-user-help
enum
enumeration
error-reference
example-scenario
function
github-sample

SLUG
guide
how-to
hub-page
include
interactive-tutorial
interface
ioctl
kb-support
landing-page
language-reference
macro
managed-reference
method
overview
portal
quickstart
reference
reference-architecture
retired
sample
struct
structure
troubleshooting
tutorial
ui-reference

Docs Markdown reference

7/8/2021 • 18 minutes to read

TIP

You may refer external Docs contributors to this public contributor guide page for similar information. Keep the information in sync between these two pages as things change.

This article provides an alphabetical reference for writing Markdown for docs.microsoft.com (Docs).

Markdown is a lightweight markup language with plain text formatting syntax. Docs supports CommonMark compliant Markdown parsed through the Markdig parsing engine. Docs also supports custom Markdown extensions that provide richer content on the Docs site.

You can use any text editor to write Markdown, but we recommend Visual Studio Code with the Docs Authoring Pack. The Docs Authoring Pack provides editing tools and preview functionality that lets you see what your articles will look like when rendered on Docs.

Alerts (Note, Tip, Important, Caution, Warning)

Alerts are a Markdown extension to create block quotes that render on Docs with colors and icons that indicate the significance of the content. The following alert types are supported:

```
> [!NOTE]
> Information the user should notice even if skimming.

> [!TIP]
> Optional information to help a user be more successful.

> [!IMPORTANT]
> Essential information required for user success.

> [!CAUTION]
> Negative potential consequences of an action.

> [!WARNING]
> Dangerous certain consequences of an action.
```

These alerts look like this on Docs:

NOTE

Information the user should notice even if skimming.

TIP

Optional information to help a user be more successful.

> **IMPORTANT**
> Essential information required for user success.

Caution

Negative potential consequences of an action.

> **WARNING**
> Dangerous certain consequences of an action.

Angle brackets

If you use angle brackets in text in your file (for example, to denote a placeholder) you need to manually encode the angle brackets. Otherwise, Markdown thinks that they're intended to be an HTML tag.

For example, encode `<script name>` as `<script name>` or `\<script name>`.

Angle brackets don't have to be escaped in text formatted as inline code or in code blocks.

Apostrophes and quotation marks

If you copy from Word into a Markdown editor, the text might contain "smart" (curly) apostrophes or quotation marks. These need to be encoded or changed to basic apostrophes or quotation marks. Otherwise, you end up with things like this when the file is published: Itâ€™s

Here are the encodings for the "smart" versions of these punctuation marks:

- Left (opening) quotation mark: `“`
- Right (closing) quotation mark: `”`
- Right (closing) single quotation mark or apostrophe: `’`
- Left (opening) single quotation mark (rarely used): `‘`

> **TIP**
> To avoid "smart" characters in your Markdown files, rely on the Docs Authoring Pack's smart quote replacement feature. For more information, see smart quote replacement.

Blockquotes

Blockquotes are created using the `>` character:

```
> This is a blockquote. It is usually rendered indented and with a different background color.
```

The preceding example renders as follows:

> This is a blockquote. It is usually rendered indented and with a different background color.

Bold and italic text

To format text as **bold**, enclose it in two asterisks:

```
This text is **bold**.
```

To format text as *italic*, enclose it in a single asterisk:

```
This text is *italic*.
```

To format text as both ***bold and italic***, enclose it in three asterisks:

```
This text is both ***bold and italic***.
```

For guidance on when to use bold and italic text, see text formatting guidelines.

Code snippets

Docs Markdown supports the placement of code snippets both inline in a sentence and as a separate "fenced" block between sentences. For more information, see How to add code to docs.

Columns

The **columns** Markdown extension gives Docs authors the ability to add column-based content layouts that are more flexible and powerful than basic Markdown tables, which are only suited for true tabular data. You can add up to four columns, and use the optional `span` attribute to merge two or more columns.

The syntax for columns is as follows:

```
:::row:::
   :::column span="":::
      Content...
   :::column-end:::
   :::column span="":::
      More content...
   :::column-end:::
:::row-end:::
```

Columns should only contain basic Markdown, including images. Headings, tables, tabs, and other complex structures shouldn't be included. A row can't have any content outside of column.

For example, the following Markdown creates one column that spans two column widths, and one standard (no `span`) column:

```
:::row:::
   :::column span="2":::
      **This is a 2-span column with lots of text.**

      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vestibulum mollis nunc
      ornare commodo. Nullam ac metus imperdiet, rutrum justo vel, vulputate leo. Donec
      rutrum non eros eget consectetur.
   :::column-end:::
   :::column span="":::
      **This is a single-span column with an image in it.**

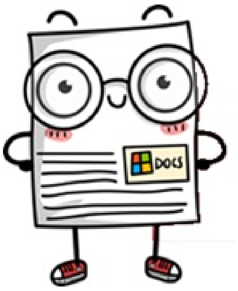
      ![Doc.U.Ment.](media/markdown-reference/document.png)
   :::column-end:::
:::row-end:::
```

This renders as follows:

This is a 2-span column with lots of text.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vestibulum mollis nunc ornare commodo. Nullam ac metus imperdiet, rutrum justo vel, vulputate leo. Donec rutrum non eros eget consectetur.

This is a single-span column with an image in it.



Comments

Docs supports HTML comments if you must comment out sections of your article:

```
<!--- Here's my comment --->
```

> **WARNING**
>
> Do not put private or sensitive information in HTML comments. Docs carries HTML comments through to the published HTML that goes public to customers. While HTML comments are invisible to the reader's eye, they are exposed in the HTML underneath.

Headings

Docs supports six levels of Markdown headings:

```
# This is a first level heading (H1)

## This is a second level heading (H2)

...

###### This is a sixth level heading (H6)
```

- There must be a space between the last `#` and heading text.
- Each Markdown file must have one and only one H1 heading.
- The H1 heading must be the first content in the file after the YML metadata block.
- H2 headings automatically appear in the right-hand navigating menu of the published file. Lower-level headings don't appear, so use H2s strategically to help readers navigate your content.
- HTML headings, such as `<h1>`, aren't recommended, and in some cases will cause build warnings.
- You can link to individual headings in a file via bookmarks.

HTML

Although Markdown supports inline HTML, HTML isn't recommended for publishing to Docs, and except for a limited list of values will cause build errors or warnings.

For more information, see HTML allowlist in the Docs Admin Guide.

Images

The following file types are supported by default for images:

- .jpg
- .png

To support additional image types, such as .gif, you must add them as resources in docfx.json:

```
"resource": [
  {
    "files" : [
      "**/*.png",
      "**/*.jpg,
      "**/*.gif"
    ],
```

Standard conceptual images (default Markdown)

The basic Markdown syntax to embed an image is:

```
![<alt text>.](<folderPath>)

Example:
![alt text for image.](../images/Introduction.png)
```

Where `<alt text>` is a brief description of the image and `<folder path>` is a relative path to the image. Alternate text is required for screen readers for the visually impaired. It's also useful if there's a site bug where the image can't render.

Underscores in alt text aren't rendered properly unless you escape them by prefixing them with a backslash (`\_`). However, don't copy file names for use as alt text. For example, instead of this:

```
![ADextension_2FA_Configure_Step4.](./media/bogusfilename/ADextension_2FA_Configure_Step4.PNG)
```

Write this:

```
![Active Directory extension for two-factor authentication, step 4: Configure.]
(./media/bogusfilename/ADextension_2FA_Configure_Step4.PNG)
```

Standard conceptual images (Docs Markdown)

The Docs custom `:::image:::` extension supports standard images, complex images, and icons.

For standard images, the older Markdown syntax will still work, but the new extension is recommended because it supports more powerful functionality, such as specifying a localization scope that's different from the parent topic. Other advanced functionality, such as selecting from the shared image gallery instead of specifying a local image, will be available in the future. The new syntax is as follows:

```
:::image type="content" source="<folderPath>" alt-text="<alt text>":::
```

If `type="content"` (the default), both `source` and `alt-text` are required.

Complex images with long descriptions

You can also use this extension to add an image with a long description that is read by screen readers but not rendered visually on the published page. Long descriptions are an accessibility requirement for complex images, such as graphs. The syntax is the following:

```
:::image type="complex" source="<folderPath>" alt-text="<alt text>":::
   <long description here>
:::image-end:::
```

If `type="complex"`, `source`, `alt-text`, a long description, and the `:::image-end:::` tag are all required.

Automatic borders

The `:::image:::` extension also supports the `border` property, which automatically adds a 1-pixel gray border around your image. This is the equivalent of the `mx-imgBorder` style described in Gray borders. The `border` property is `true` by default for `content` and `complex` images, so you'll get the border automatically unless you explicitly add the property with a value of `false`. The `border` property is `false` by default for `icon` images.

The `border` property is the recommended way to add a border - please don't create your own borders manually.

Creating an expandable image

The optional `lightbox` property allows you to create an expanded image, as described in Create an expandable screenshot (lightbox). The value of `lightbox` is the path to the expanded image.

Specifying loc-scope

Sometimes the localization scope for an image is different from that of the article or module that contains it. This can cause a bad global experience: for example, if a screenshot of a product is accidentally localized into a language the product isn't available in. To prevent this, you can specify the `loc-scope` attribute. The `loc-scope` attribute can be specified on any image of types `content` and `complex`, and is *required* for screenshots that show a product with a different localization scope than the article or module that contains it.

For more information, see How to specify the localization scope for an image.

Icons

The image extension supports icons, which are decorative images and should not have alt text. The syntax for icons is:

```
:::image type="icon" source="<folderPath>":::
```

If `type="icon"`, `source` should be specified but `alt-text` shouldn't be.

The `border` property is `false` by default for icons. If your decorative image requires the standard image border, explicitly add `border="true"` to the `:::image:::` tag.

The `lightbox` property works the same for `icon` images as for standard `content` images.

Other image file types

You can add support for other image types by adding them as resources to the docfx.json file for your docset. For example, add .gif to enable animated .gif files.

For more information about creating and using images, see Create a screenshot, Create an expandable screenshot, Create conceptual art, and Create an animated GIF.

Included Markdown files

Where markdown files need to be repeated in multiple articles, you can use an include file. The includes feature instructs Docs to replace the reference with the contents of the include file at build time. You can use includes in the following ways:

- Inline: Reuse a common text snippet inline with within a sentence.
- Block: Reuse an entire Markdown file as a block, nested within a section of an article.

An inline or block include file is a Markdown (.md) file. It can contain any valid Markdown. Include files are typically located in a common *includes* subdirectory, in the root of the repository. When the article is published, the included file is seamlessly integrated into it.

Includes syntax

Block include is on its own line:

```
[!INCLUDE [<title>](<filepath>)]
```

Inline include is within a line:

```
Text before [!INCLUDE [<title>](<filepath>)] and after.
```

Where `<title>` is the name of the file and `<filepath>` is the relative path to the file. `INCLUDE` must be capitalized and there must be a space before the `<title>`.

Here are requirements and considerations for include files:

- Use block includes for significant amounts of content--a paragraph or two, a shared procedure, or a shared section. Do not use them for anything smaller than a sentence.
- Includes won't be rendered in the GitHub rendered view of your article, because they rely on Docs extensions. They'll be rendered only after publication.
- Write all the text in an include file in complete sentences or phrases that don't depend on preceding or following text in the article that references the include. Ignoring this guidance creates an untranslatable string in the article.
- Don't embed include files within other include files.
- `/Includes` folders are excluded from build. Therefore, images stored in `/includes` folders and referenced in included files won't be displayed in published content. Store images in a `/media` folder outside the `/includes` folder.
- As with regular articles, don't share media between include files. Use a separate file with a unique name for each include and article. Store the media file in the media folder that's associated with the include.
- Don't use an include as the only content of an article. Includes are meant to be supplemental to the content in the rest of the article.

For more information, see Include reusable content in articles.

Links

For information on syntax for links, see Links.

Lists (Numbered, Bulleted, Checklist)

Numbered list

To create a numbered list, you can use all 1s. The numbers are rendered in ascending order as a sequential list when published. For increased source readability, you can increment your lists manually.

Don't use letters in lists, including nested lists. They don't render correctly when published to Docs. Nested lists using numbers will render as lowercase letters when published. For example:

```
1. This is
1. a parent numbered list
   1. and this is
   1. a nested numbered list
1. (fin)
```

This renders as follows:

1. This is
2. a parent numbered list
   a. and this is
   b. a nested numbered list
3. (fin)

Bulleted list

To create a bulleted list, use `-` or `*` followed by a space at the beginning of each line:

```
- This is
- a parent bulleted list
  - and this is
  - a nested bulleted list
- All done!
```

This renders as follows:

- This is
- a parent bulleted list
  - and this is
  - a nested bulleted list
- All done!

Whichever syntax you use, `-` or `*`, use it consistently within an article.

Checklist

Checklists are available for use on Docs via a custom Markdown extension:

```
> [!div class="checklist"]
> * List item 1
> * List item 2
> * List item 3
```

This example renders on Docs like this:

- List item 1
- List item 2
- List item 3

Use checklists at the beginning or end of an article to summarize "What will you learn" or "What have you learned" content. Do not add random checklists throughout your articles.

Next step action

You can use a custom extension to add a next step action button to Docs pages.

The syntax is as follows:

```
> [!div class="nextstepaction"]
> [button text](link to topic)
```

For example:

```
> [!div class="nextstepaction"]
> [Learn about adding code to articles](code-in-docs.md)
```

This renders as follows:

Learn about adding code to articles

You can use any supported link in a next step action, including a Markdown link to another web page. In most cases, the next action link will be a relative link to another file in the same docset.

Non-localized strings

You can use the custom `no-loc` Markdown extension to identify strings of content that you would like the localization process to ignore.

All strings called out will be case-sensitive; that is, the string must match exactly to be ignored for localization.

To mark an individual string as non-localizable, use this syntax:

```
:::no-loc text="String":::
```

For example, in the following, only the single instance of `Document` will be ignored during the localization process:

```
# Heading 1 of the Document

Markdown content within the :::no-loc text="Document":::.  The are multiple instances of Document, document,
and documents.
```

NOTE

Use `\` to escape special characters:

```
Lorem :::no-loc text="Find a \"Quotation\""::: Ipsum.
```

You can also use metadata in the YAML header to mark all instances of a string within the current Markdown file as non-localizable:

```
author: cillroy
no-loc: [Global, Strings, to be, Ignored]
```

> **NOTE**
>
> The no-loc metadata is not supported as global metadata in *docfx.json* file. The localization pipeline doesn't read the *docfx.json* file, so the no-loc metadata must be added into each individual source file.

In the following example, both in the metadata `title` and the Markdown header the word `Document` will be ignored during the localization process.

In the metadata `description` and the Markdown main content the word `document` is localized, because it does not start with a capital `D`.

```
---
title: Title of the Document
author: author-name
description: Description for the document
no-loc: [Title, Document]
---
# Heading 1 of the Document

Markdown content within the document.
```

Selectors

Selectors are UI elements that let the user switch between multiple flavors of the same article. They are used in some doc sets to address differences in implementation across technologies or platforms. Selectors are typically most applicable to our mobile platform content for developers.

Because the same selector Markdown goes in each article file that uses the selector, we recommend placing the selector for your article in an include file. Then you can reference that include file in all your article files that use the same selector.

There are two types of selectors: a single selector and a multi-selector.

Single selector

```
> [!div class="op_single_selector"]
> - [Universal Windows](../articles/notification-hubs-windows-store-dotnet-get-started/)
> - [Windows Phone](../articles/notification-hubs-windows-phone-get-started/)
> - [iOS](../articles/notification-hubs-ios-get-started/)
> - [Android](../articles/notification-hubs-android-get-started/)
> - [Kindle](../articles/notification-hubs-kindle-get-started/)
> - [Baidu](../articles/notification-hubs-baidu-get-started/)
> - [Xamarin.iOS](../articles/partner-xamarin-notification-hubs-ios-get-started/)
> - [Xamarin.Android](../articles/partner-xamarin-notification-hubs-android-get-started/)
```

... will be rendered like this:

Multi-selector

```
> [!div class="op_multi_selector" title1="Platform" title2="Backend"]
> - [(iOS | .NET)](./mobile-services-dotnet-backend-ios-get-started-push.md)
> - [(iOS | JavaScript)](./mobile-services-javascript-backend-ios-get-started-push.md)
> - [(Windows universal C# | .NET)](./mobile-services-dotnet-backend-windows-universal-dotnet-get-started-push.md)
> - [(Windows universal C# | Javascript)](./mobile-services-javascript-backend-windows-universal-dotnet-get-started-push.md)
> - [(Windows Phone | .NET)](./mobile-services-dotnet-backend-windows-phone-get-started-push.md)
> - [(Windows Phone | Javascript)](./mobile-services-javascript-backend-windows-phone-get-started-push.md)
> - [(Android | .NET)](./mobile-services-dotnet-backend-android-get-started-push.md)
> - [(Android | Javascript)](./mobile-services-javascript-backend-android-get-started-push.md)
> - [(Xamarin iOS | Javascript)](./partner-xamarin-mobile-services-ios-get-started-push.md)
> - [(Xamarin Android | Javascript)](./partner-xamarin-mobile-services-android-get-started-push.md)
```

... will be rendered like this:

Subscript and superscript

You should only use subscript or superscript when necessary for technical accuracy, such as when writing about mathematical formulas. Don't use them for non-standard styles, such as footnotes.

For both subscript and superscript, use HTML:

```
Hello <sub>This is subscript!</sub>
```

This renders as follows:

Hello $_{\text{This is subscript!}}$

```
Goodbye <sup>This is superscript!</sup>
```

This renders as follows:

Goodbye $^{\text{This is superscript!}}$

Tabbed conceptual

Tabbed conceptual is a Markdown extension for Docs that allows us to present different versions of content, such as procedural steps to accomplish the same task on different platforms, in a tabbed format. See Tabbed Conceptual in the Docs Admin Guide for more information.

Tables

The simplest way to create a table in Markdown is to use pipes and lines. To create a standard table with a header, follow the first line with dashed line:

```
|This is   |a simple   |table header|
|----------|-----------|------------|
|table     |data       |here        |
|it doesn't|actually   |have to line up nicely!|
```

This renders as follows:

THIS IS	A SIMPLE	TABLE HEADER
table	data	here
it doesn't	actually	have to line up nicely!

You can align the columns by using colons:

```
| Fun                  | With                 | Tables          |
| :------------------- | -------------------: |:---------------:|
| left-aligned column  | right-aligned column | centered column |
| $100                 | $100                 | $100            |
| $10                  | $10                  | $10             |
| $1                   | $1                   | $1              |
```

The Markdown above renders as follows:

FUN	WITH	TABLES
left-aligned column	right-aligned column	centered column
$100	$100	$100
$10	$10	$10
$1	$1	$1

TIP

The Docs Authoring Extension for VS Code makes it easy to add basic Markdown tables!

You can also use an online table generator.

Line breaks within words in any table cell

Long words in a Markdown table might make the table expand to the right navigation and become unreadable. You can solve that by allowing Docs rendering to automatically insert line breaks within words when needed. Just wrap up the table with the custom class `[!div class="mx-tdBreakAll"]`.

Here is a Markdown sample of a table with three rows that will be wrapped by a `div` with the class name `mx-tdBreakAll`.

```
> [!div class="mx-tdBreakAll"]
> |Name|Syntax|Mandatory for silent installation?|Description|
> |-------------|----------|---------|---------|
> |Quiet|/quiet|Yes|Runs the installer, displaying no UI and no prompts.|
> |NoRestart|/norestart|No|Suppresses any attempts to restart. By default, the UI will prompt before restart.|
> |Help|/help|No|Provides help and quick reference. Displays the correct use of the setup command, including a list of all options and behaviors.|
```

It will be rendered like this:

NAME	SYNTAX	MANDATORY FOR SILENT INSTALLATION?	DESCRIPTION
Quiet	/quiet	Yes	Runs the installer, displaying no UI and no prompts.
NoRestart	/norestart	No	Suppresses any attempts to restart. By default, the UI will prompt before restart.
Help	/help	No	Provides help and quick reference. Displays the correct use of the setup command, including a list of all options and behaviors.

Line breaks within words in second column table cells

You might want line breaks to be automatically inserted within words only in the second column of a table. To limit the breaks to the second column, apply the class `mx-tdCol2BreakAll` by using the `div` wrapper syntax as shown earlier.

Inconsistent column widths between tables

You may notice that the column widths of the tables in your articles look odd or inconsistent. This behavior occurs because the length of text within the cells determines the layout of the table. Unfortunately, there's no way to control how the tables render. This is a limitation of Markdown. Even though it would look nicer to have the width of table columns be consistent, this would have some disadvantages too:

- Interlacing HTML code with Markdown makes topics more complicated and discourages community contributions.
- A table that you make look good for a specific screen size may end up looking completely unreadable at different screen sizes as it preempts responsive rendering.

Data matrix tables

A data matrix table has both a header and a weighted first column, creating a matrix with an empty cell in the top left. Docs has custom Markdown for data matrix tables:

```
|                  |Header 1 |Header 2|
|------------------|---------|--------|
|**First column A**|Cell 1A  |Cell 2A |
|**First column B**|Cell 1B  |Cell 2B |
```

The example renders as:

	HEADER 1	HEADER 2
First column A	Cell 1A	Cell 2A
First column B	Cell 1B	Cell 2B

Every entry in the first column must be styled as bold (`**bold**`); otherwise the tables won't be accessible for screen readers or valid for Docs.

> **TIP**
>
> The Docs Authoring Pack for VS Code includes a function to convert a regular Markdown table into a data matrix table. Just select the table, right-click, and select **Convert to data matrix table**.

HTML Tables

HTML tables aren't recommended for Docs. They aren't human readable in the source - which is a key principle of Markdown.

Videos

For more information, see Embedding video in your content.

How to include code in docs

7/8/2021 • 18 minutes to read

TIP

You may refer external docs contributors to this public contributor guide page for similar information. Keep the information in sync between these two pages as things change.

There are several ways to include code in an article published on docs.microsoft.com:

- Individual elements (words) within a line.

  Here's an example of `code` style.

  Use code format when referring to named parameters and variables in a nearby code block in your text. Code format may also be used for properties, methods, classes, and language keywords. For more information, see Code elements later in this article.

- Code blocks in the article Markdown file.

  ````
  ```csharp
  public static void Log(string message)
  {
      _logger.LogInformation(message);
  }
  ```
  ````

  Use inline code blocks when it's impractical to display code by reference to a code file. For more information, see Code blocks later in this article.

- Code blocks by reference to a code file in the local repository.

  ```
  :::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-
  24,26":::
  ```

  For more information, see In-repo snippet references later in this article.

- Code blocks by reference to a code file in another repository.

  ```
  :::code language="csharp" source="~/samples-durable-functions/samples/csx/shared/Location.csx"
  highlight="2,5":::
  ```

  For more information, see Out-of-repo snippet references later in this article.

- Code blocks that let the user run code in the browser.

  ```
  :::code source="PowerShell.ps1" interactive="cloudshell-powershell":::
  ```

  For more information, see Interactive code snippets later in this article.

Besides explaining the Markdown for each of these ways to include code, the article provides some general guidance for all code blocks.

Code elements

A "code element" is a programming language keyword, class name, property name, and so forth. It's not always obvious what qualifies as code. For example, NuGet package names should be treated as code. When in doubt, see Text formatting guidelines.

Inline code style

To include a code element in article text, surround it with backticks (`) to indicate code style. Inline code style shouldn't use the triple-backtick format.

MARKDOWN	RENDERED
By default, the Entity Framework interprets a property that's named \`Id\` or \`ClassnameID\` as the primary key.	By default, the Entity Framework interprets a property that's named `Id` or `ClassnameID` as the primary key.

When an article is localized (translated into other languages), text styled as code is left untranslated. If you want to prevent localization without using code style, see Non-localized strings.

Bold style

Some older style guides specify bold for inline code. Bold is an option when code style is so obtrusive as to compromise readability. For example, a Markdown table with mostly code elements might look too busy with code styling everywhere. If you choose to use bold style, use non-localized strings syntax to make sure that code is not localized.

Links

A link to reference documentation may be more helpful than code format in some contexts. If you use a link, you can apply code format to the link text, but there's no difference in the rendered appearance compared to link text that isn't styled as code.

If you use a link and refer to the same element later in the same context, make the subsequent instances code format rather than links.

Placeholders

If you want the user to replace a section of displayed code with their own values, use placeholder text marked off by angle brackets. For example:

> `az group delete -n <ResourceGroupName>`

You may note that the brackets must be removed when substituting real values. The Microsoft Writing Style Guide calls for italics, which you may format within angle bracketed inline code:

> `<ResourceGroupName>` is the resource group where...

Curly braces { } are discouraged for use as syntactical placeholders. They may be confused with the same notation used in replaceable text, format strings, string interpolation, text templates, and similar programming constructs.

Placeholder names can be separated by hyphens ("kebab case"), with underscores, or not separated at all (Pascal case). Kebab case may generate syntax errors and underscores may conflict with underlining. All-caps may conflict with named constants in many languages, though it may also draw attention to the placeholder name.

> `<Resource-Group-Name>` or `<ResourceGroupName>`

Code blocks

The syntax for including code in a doc depends on where the code is located:

- In the article Markdown file
- In a code file in the same repository
- In a code file in a different repository

Following are guidelines that apply to all types of code blocks:

- Screenshots
- Automate code validation.
- Highlight key lines of code.
- Avoid horizontal scroll bars.

Screenshots

All of the methods listed in the preceding section result in usable code blocks:

- You can copy and paste from them.
- They're indexed by search engines.
- They're accessible to screen readers.

These are just a few of the reasons why IDE screenshots aren't recommended as a method of including code in an article. Use IDE screenshots for code only if you're showing something about the IDE itself, like IntelliSense. Don't use screenshots just to show colorization and highlighting.

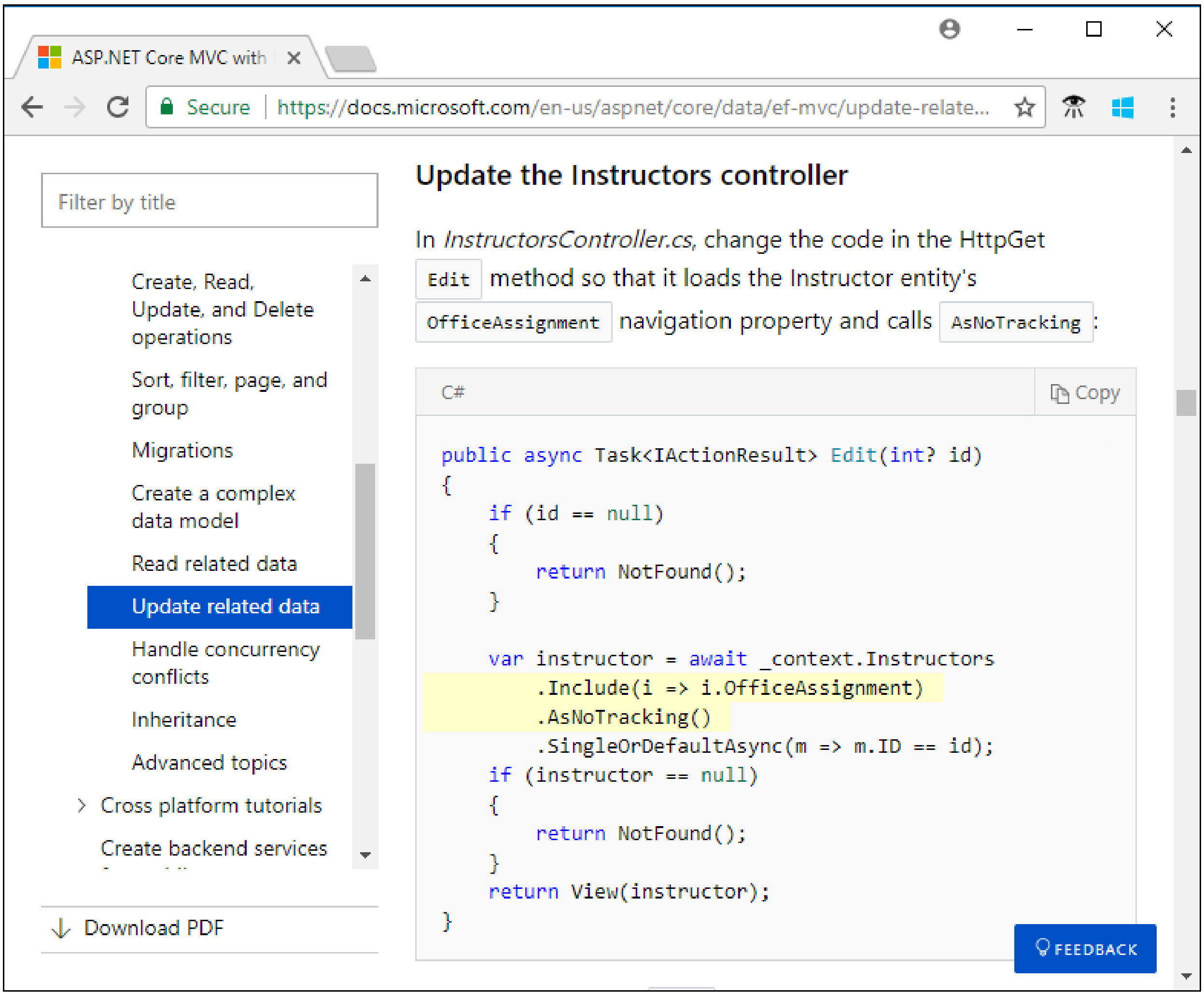
Code validation

Some repositories have implemented processes that automatically compile all sample code to check for errors. The .NET repository does this. For more information, see Learn how to contribute to the .NET docs repositories.

If you are including code blocks from another repository, work with the owners on a maintenance strategy for the code so that your included code does not break or go out of date as new versions of the libraries the code uses are shipped.

Highlighting

Snippets typically include more code than necessary in order to provide context. It helps readability when you highlight the key lines that you're focusing on in the snippet, as in this example:



You can't highlight code when you include it in the article Markdown file. It works only for code snippets included by reference to a code file.

Horizontal scroll bars

Break up long lines to avoid horizontal scroll bars. Scroll bars on code blocks make code hard to read. They're especially problematic on longer code blocks, where it may be impossible to see the scroll bar and the line you want to read at the same time.

A good practice for minimizing horizontal scroll bars on code blocks is to break up code lines longer than 85 characters. But keep in mind that the presence or absence of a scroll bar isn't the only criterion of readability. If breaking a line before 85 hurts readability or copy-paste convenience, feel free to go over 85.

Inline code blocks

Use inline code blocks only when it's impractical to display code by reference to a code file. Inline code is more difficult to test and keep up to date compared to a code file that is part of a complete project. And inline code may omit context that could help the developer to understand and use the code. These considerations apply mainly to programming languages. Inline code blocks can also be used for outputs and inputs (such as JSON), query languages (such as SQL), and scripting languages (such as PowerShell).

There are two ways to indicate a section of text in an article file is a code block: by *fencing* it in triple-backticks (```) or triple tildes (~~~), or by indenting it. Fencing is preferred because it lets you specify the language. Avoid using indentation because it's too easy to get wrong and it may be hard for another writer to understand your intent when they need to edit your article.

Language indicators are placed immediately after the opening triple-backticks or triple-tildes, as in the following example:

Markdown:

````
```json
{
    "aggregator": {
        "batchSize": 1000,
        "flushTimeout": "00:00:30"
    }
}
```
````

Rendered:

```json
{
    "aggregator": {
        "batchSize": 1000,
        "flushTimeout": "00:00:30"
    }
}
```

TIP

GitHub Flavored Markdown supports delimiting code blocks with tildes (~) as well as with backticks (`). The symbol used to open and close the code block must be consistent within the same code block.

For information about the values you can use as language indicators, see Supported languages.

If you use a language or environment word after the triple-backticks (```) that isn't supported, that word appears in the code section title bar on the rendered page. Whenever possible, use a language or environment indicator in your inline code blocks.

NOTE

If you copy and paste code from a Word document, make sure it has no "curly quotes," which aren't valid in code. If it does, change them back to normal quotes (`'` and `"`). Alternatively, rely on the Docs Authoring Pack, smart quotes replacement feature.

In-repo snippet references

The preferred way to include code snippets for programming languages in docs is by reference to a code file. This method gives you the ability to highlight lines of code and makes the wider context of the snippet available on GitHub for developers to use. You can include code by using the triple colon format (:::) either manually or in Visual Studio Code with the help of the docs.microsoft.com Authoring Pack.

1. In Visual Studio Code, click **Alt + M** or **Option + M** and select Snippet.
2. Once Snippet is selected, you will be prompted for Full Search, Scoped Search, or Cross-Repository Reference. To search locally, select Full Search.
3. Enter a search term to find the file. Once you've found the file, select the file.
4. Next, select an option to determine which line(s) of code should be included in the snippet. The options are: **ID**, **Range**, and **None**.
5. Based on your selection from Step 4, provide a value if necessary.

Display entire code file:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs":::
```

Display part of a code file by specifying line numbers:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26":::
```

Display part of a code file by specifying a snippet name:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" id="snippet_Create":::
```

The following sections explain these examples:

- Use a relative path to the code file
- Include only selected line numbers
- Refer to a named snippet
- Highlight selected lines

For more information, see Snippet syntax reference later in this article.

Path to code file

Example:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26":::
```

The example is from the ASP.NET docs repo, aspnetcore/data/ef-mvc/crud.md article file. The code file is referenced by a relative path to aspnetcore/data/ef-mvc/intro/samples/cu/Controllers/StudentsController.cs in the same repository.

The preceding example uses the folder structure of the ASP.NET Core docs repository. If your repo doesn't already have guidelines for where to locate code files that you access through snippet references, use the folder structure recommended for the .NET docs:

- In the same folder as the article that shows the snippets, create a *snippets* folder.
- In the *snippets* folder, create a folder that has the same name as the article file, but without the *.md* extension. If snippet files are shared among two or more articles in the same folder, name the folder *shared* instead of using an article file name.
- If your doc set shows multiple languages for the same snippets, create a folder for each language. Name the folders using the language code used to identify snippets. For example, language-specific folders could be named *csharp* and *vb*.
- Put the code file that has snippets in the folder named for a language. If language-named folders aren't used, put the code file in the folder that is named after the article or *shared*.

For an example, see Example folder structure.

Selected line numbers

Example:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26":::
```

This example displays only lines 2-24 and 26 of the *StudentController.cs* code file.

Prefer named snippets over hard-coded line numbers, as explained in the next section.

Named snippet

Example:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" id="Create":::
```

Use only letters and underscores for the name.

The example displays the `Create` section of the code file. The code file for this example has snippet tags in comments in the C# code:

```
// code excluded from the snippet
// <Create>
// code included in the snippet
// </Create>
// code excluded from the snippet
```

Named code snippets can be nested, as shown in the following example:

```
// <Method>
public static SomeMethod()
{
    // <Line>
    // Single line of code.
    // </Line>
}
// </Method>
```

When the `snippet_Method` code snippet is rendered, the `snippet_Line` tags aren't included in the rendered output.

Whenever you can, refer to a named section rather than specifying line numbers. Line number references are brittle because code files inevitably change in ways that make line numbers change. You don't necessarily get notified of such changes. Your article eventually starts showing the wrong lines and you have no clue anything has changed.

Highlighting selected lines

Example:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26"
highlight="2,5":::
```

The example highlights lines 2 and 5, counting from the start of the displayed snippet. Line numbers to highlight don't count from the start of the code file. In other words, lines 3 and 6 of the code file are highlighted.

Out-of-repo snippet references

If the code file you want to reference is in a different repository, set up the code repository as a *dependent repository*. When you do that, you specify a name for it. That name then acts like a folder name for purposes of code references.

Dependent repositories metadata

The data structure to set up dependent repositories for cross-repo references (CRRs) is a list that contains path, repository, and branch information. The following `dependent_repositories` list property is required in the *.openpublishing.publish.config.json* file.

```
{
    "dependent_repositories": [
        {
            "path_to_root": "<relative path to repository root>",
            "url": "<referenced repository url>",
            "branch": "<branch name of referenced repository>",
            "branch_mapping": {
                "<source repository branch>": "<referenced repository branch>",
                "<source repository branch>": "<referenced repository branch>"
            }
        },
        {
            "path_to_root": "token",
            "url": "https://github.com/Microsoft/token",
            "branch": "master",
            "branch_mapping": {
                "master": "master",
                "develop": "test"
            }
        }
    ]
}
```

METADATA	MEANING	PARAMETER REQUIRED
dependent_repositories	The CRR relationship list name	Yes
path_to_root	A relative folder path to repository root. The folder doesn't have to have been created yet.	Yes
url	URL of the reference repository that the current repository is dependent on.	Yes
branch	The branch of the reference repository that you want to use when doing builds.	Yes
branch_mapping	**Advanced feature**: A branch map of current repository and reference repository. Key is current repository's branch and value is reference repository's branch. This property allows you to use different reference branches when you build source from different branches of the current repository. **If the source build branch isn't specified in this property, the `branch` value will be used.**	No

For sample JSON that you can add to the *.openpublishing.publish.config.json* file and reference in CRR snippets, see Cross Repository Reference.

Snippet reference syntax

You use the same triple-colon snippet syntax for CRRs as for in-repo snippets. Only the path to the snippet is different.

A path to a CRR snippet must either start with a tilde (`~`) if it's relative, or else it must path out of the repo with

`..` syntax. The tilde signifies the root folder. For example, in the SQL repository, `~` represents "docs". To point to the repository root, use `~/..`. `..` denotes the parent directory one level back, or 'up,' relative to the current directory. This sequence can be used multiple times, for example `../../` denotes a directory 2 levels back. If a relative path requires several levels of backing out, consider the complexity of the path and the user experience; you may decide to use a different, friendlier link syntax.

Note that relative file paths cannot be directly copied and pasted because the folder they point to is relative to your working directory. If your environment differs from the source code environment, the path will be invalid.

The following snippet reference uses a relative file path:

`~/../xamarin-forms-samples/WebServices/TodoREST/TodoAPI/TodoAPI/Startup.cs`

> **NOTE**
>
> The name you assign to the dependent repository is relative to the root of the main repository, but the tilde (~) refers to the root of the doc set. The doc set root is determined by `build_source_folder` in *.openpublishing.publish.config.json*. The path to the snippet in the preceding example is correct for the AspNetCore.Docs repo because `build_source_folder` in that repository's *.openpublishing.publish.config.json* file is set to the *aspnetcore* folder.
>
> ```
> "build_source_folder": "aspnetcore"
> ```
>
> Suppose the AspNetCore.Docs repo had the following setting in its *.openpublishing.publish.config.json* file:
>
> ```
> "build_source_folder": "."
> ```
>
> In that case, the path would start with `~/xamarin-forms-samples` instead of `~/../xamarin-forms-samples`.

> **IMPORTANT**
>
> Updating an external code snippet won't automatically trigger a content build. You need to trigger the build by either changing something in the doc repo or manually starting a build.

Snippets in a Jupyter notebook

You can reference a cell in a Jupyter notebook as a code snippet. In order to reference the cell:

1. Add cell metadata to the notebook for the cells you wish to reference.
2. Set up access to the repository.
3. Use Jupyter notebook snippet syntax in your markdown file.

Add metadata to notebook

1. Name the cell by adding cell metadata in the Jupyter notebook.
   - In Jupyter, you can edit cell metadata by first enabling the cell toolbar: **View > Cell Toolbar > Edit Metadata**.
   - Once the cell toolbar is enabled, select **Edit Metadata** on the cell you wish to name.
   - Or you can edit metadata directly in the notebook's JSON structure.
2. In the cell metadata, add a "name" attribute:

   ```
   "metadata": {"name": "<name>"},
   ```

   For example:

```
"metadata": {"name": "workspace"},
```

> **TIP**
>
> You can add any other metadata you'd like to help you track where the cell is being used. For example:
>
> ```
>     "metadata": {
>       "name": "workspace",
>       "msdoc": "how-to-track-experiments.md"
>     },
> ```

Set up repository access

If the notebook file you want to reference is in a different repository, set up the code repository as a dependent repository.

Jupyter notebook snippet syntax reference

Once your notebook contains the required metadata, reference it in your markdown file. Use the `<cell-name-value>` you added to the notebook, and the `<path>` you set up as your dependent repository.

```
[!notebook-<language>[] (<path>/<notebook-name.ipynb>?name=<cell-name-value>)]
```

For example:

```
[!notebook-python[] (~/MachineLearningNotebooks/train-on-local.ipynb?name=workspace)]
```

> **IMPORTANT**
>
> This syntax is a block Markdown extension. It must be used on its own line.

Use any of the supported languages for the `<language>` identifier.

Interactive code snippets

Inline interactive code blocks

For the following languages, code snippets can be made executable in the browser window:

- Azure Cloud Shell
- Azure PowerShell Cloud Shell
- C# REPL

When interactive mode is enabled, the rendered code boxes have a **Try It** or **Run** button. For example:

````
    ```azurepowershell-interactive
    New-AzResourceGroup -Name myResourceGroup -Location westeurope
    ```
````

renders as follows:

```
New-AzResourceGroup -Name myResourceGroup -Location westeurope
```

And

````
```csharp-interactive
var aFriend = "Maria";
Console.WriteLine($"Hello {aFriend}");
```
````

renders as:

```
var aFriend = "Maria";
Console.WriteLine($"Hello {aFriend}");
```

To turn on this feature for a particular code block, use a special language identifier. The available options are:

- `azurepowershell-interactive` - Enables the Azure PowerShell Cloud Shell, as in the preceding example
- `azurecli-interactive` - Enables the Azure Cloud Shell
- `csharp-interactive` - Enables Try .NET

For the Azure Cloud Shell and PowerShell Cloud Shell, users can run commands against only their own Azure account.

Code snippets included by reference

You can enable interactive mode for code snippets included by reference. To turn on this feature for a particular code block, use the `interactive` attribute. The available attribute values are:

- `cloudshell-powershell` - Enables the Azure PowerShell Cloud Shell, as in the preceding example
- `cloudshell-bash` - Enables the Azure Cloud Shell
- `try-dotnet` - Enables Try .NET
- `try-dotnet-class` - Enables Try .NET with class scaffolding
- `try-dotnet-method` - Enables Try .NET with method scaffolding

Here are some examples:

```
:::code source="PowerShell.ps1" interactive="cloudshell-powershell":::
```

```
:::code source="Bash.sh" interactive="cloudshell-bash":::
```

For the Azure Cloud Shell and PowerShell Cloud Shell, users can only run commands against their own Azure account.

Snippet syntax reference

Syntax:

```
:::code language="<language>" source="<path>" <attribute>="<attribute-value>":::
```

> **IMPORTANT**
>
> This syntax is a block Markdown extension. It must be used on its own line.

- `<language>` (*optional*)

    - Language of the code snippet. For more information, see the Supported languages section later in this article.
- `<path>` (*mandatory*)
    - Relative path in the file system that indicates the code snippet file to reference.
- `<attribute>` and `<attribute-value>` (*optional*)
    - Used together to specify how the code should be retrieved from the file and how it should be displayed:
        - `range`: `1,3-5` A range of lines. This example includes lines 1, 3, 4, and 5.
        - `id`: `snippet_Create` The ID of the snippet that needs to be inserted from the code file. This value cannot coexist with range.
        - `highlight`: `2-4,6` Range and/or numbers of lines that need to be highlighted in the generated code snippet. The numbering is relative to the lines displayed (as specified by range or id), not the file.
        - `interactive`: `cloudshell-powershell`, `cloudshell-bash`, `try-dotnet`, `try-dotnet-class`, `try-dotnet-method` String value determines what kinds of interactivity are enabled.
        - For details about tag name representation in code snippet source files by language, see the DocFX guidelines.

To enable Try .NET with class scaffolding:

```
:::code language="csharp" source="ClassLevel.cs" interactive="try-dotnet-class":::
```

This will render a code block with a **Try It** button. When the **Try It** button is clicked, the interactive window only displays code inside the class.

```
public static void Main()
{
    // Specify the data source.
    int[] scores = new int[] { 97, 92, 81, 60 };        // Define the query expression.

    IEnumerable<int> scoreQuery =
        from score in scores
        where score > 80
        select score;

    // Execute the query.
    foreach (int i in scoreQuery)
    {
        Console.Write(i + " ");
    }
}
```

To enable Try .NET with method scaffolding:

```
:::code language="csharp" source="MethodLevel.cs" interactive="try-dotnet-method":::
```

This will render a code block with a **Try It** button, that when selected, activates an interactive window that only displays code inside the method.

```
// Create a new dictionary of strings, with string keys.
//
Dictionary<string, string> openWith =
    new Dictionary<string, string>();

// Add some elements to the dictionary. There are no
// duplicate keys, but some of the values are duplicates.
openWith.Add("txt", "notepad.exe");
openWith.Add("bmp", "paint.exe");
openWith.Add("dib", "paint.exe");
openWith.Add("rtf", "wordpad.exe");

// The Add method throws an exception if the new key is
// already in the dictionary.
try
{
    openWith.Add("txt", "winword.exe");
}
catch (ArgumentException)
{
    Console.WriteLine("An element with Key = \"txt\" already exists.");
}
```

To enable Try .NET with full code experience:

```
:::code language="csharp" source="Example.cs" interactive="try-dotnet":::
```

This will render a code block with a **Try It** button, that when selected, activates an interactive window that displays full code including the using statement.

```
using System;

namespace Calculator
{
    class Program
    {
        static void Main(string[] args)
        {
            // Declare variables and then initialize to zero.
            int num1 = 1; int num2 = 2;

            Console.WriteLine("num1 + num2 = " + (num1 + num2));
            Console.WriteLine("num1 - num2 = " + (num1 - num2));
            Console.WriteLine("num1 * num2 = " + (num1 * num2));
            Console.WriteLine("num1 / num2 = " + (num1 / num2));
        }
    }
}
```

Supported languages

The Docs Authoring Pack includes a feature to provide statement completion and validation of the available language identifiers for code fence blocks. Available languages are pulled from the Docs devlang taxonomy. Each devlang slug corresponds to a standard label and accurate colorization for the language. Only languages in the list render correctly on Docs.

To request a new language, complete the steps in How to request metadata changes.

TIP

The Docs Authoring Pack Dev Lang Completion feature uses the first valid alias when multiple aliases are available.

Next steps

For information about text formatting for content types other than code, see Text formatting guidelines.

Add links to articles

7/15/2021 • 18 minutes to read

This article describes how to use hyperlinks from pages hosted at Docs. Links are easy to add into Markdown with a few varying conventions. Links point users to content in the same page, other neighboring pages, or to external sites and URLs.

By order of preference, links hosted on Docs should be Relative if they are in the same repository and docset. If they are in a different docset, even if in the same repository, they should be Site Relative. Links to content hosted on Docs shouldn't use a complete URL, otherwise known as Fully Qualified Domain Names (FQDN). Using a complete URL from Docs to other content on Docs will cause that link to be non-functional in air-gapped cloud (AGC) environments.

You may refer external docs contributors to this public contributor guide page for similar information. Help keep the information in sync between these two pages as things change.

> **TIP**
> The Docs Authoring Pack extension for Visual Studio Code can help you insert links to other files and headings without the tedium of figuring out paths. Several link options are available:
>
> - **Link to file in repo** inserts a relative link to a file in the same docset.
> - **Link to web page** inserts a fully-qualified link to a page outside of docs.microsoft.com.
> - **Link to Docs page by URL** converts a Docs URL to a relative link (if the page is in the same docset) or a site-relative link (if the page is in a different docset). This is required so Docs links can be validated in build and aren't broken in isolated environments.
> - **Link to heading** inserts a link to a subheading, such as an H2, in the current article or another article in the docset.
> - **Link to Xref** inserts a cross reference link.

Link text

Link text is the text that's displayed for a clickable link. Descriptive link text makes content easier to consume and improves accessibility and SEO. Instead of "click here", use the title of the page you're linking to or other descriptive text.

Why link text matters

Accessibility:

- Users interact with links in many ways, including the use of adaptive technology (for example, screen readers and speech-recognition technology). When using a screen reader, users can generate a list of all the links on a page and navigate to them. In this case, link text like "click here" is useless.
- Displaying the actual URL isn't good because screen readers will read the whole URL aloud. If a link is long or contains many numbers or symbols, it can be unpleasant to listen to the whole link being read character by character.
- Users using speech-recognition technology can say the link text they want to navigate to, so it's important for link text to be concise, easy to say, and unique.
- On mobile devices, it takes more time to determine what a link is linking to than on a computer. So it's nice to know what you're clicking or selecting and where it's taking you.

SEO:

Good link text can improve your page's search ranking. Link text helps search engines understand how other people see your page and what it's about.

How to write good link text

When writing link text, consider:

- How will this sound read aloud?
- Do I understand the purpose of the link out of context?

Link text should:

- Contain text that gives a description of the link.
- Be meaningful when read out of context.
- Be unique.
- Be easy to say.
- Be concise.
- Use at least one full word (two is typically better).

Link text shouldn't:

- Use the word "link".
- Display the actual URL.
- Be redundant.
- Be all capitalized.
- Be long sentences or paragraphs.
- Be overly general, for example "click here", "more", "info", "read more".

Link text examples

BAD LINK TEXT	BETTER LINK TEXT
To learn more about how to make chocolate chip cookies `[click here](https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/)` .	Learn more about how to make `[chocolate chip cookies](https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/)` .
To learn more about how to make chocolate chip cookies: `[https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/](https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/)`	Learn more about how to make `[chocolate chip cookies](https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/)` .
`[The best chocolate chip cookies have one cup of white sugar and one cup of brown, a whole stick of butter and of course semisweet chocolate chips. Make sure to bake them at 350 degrees F and remember to take them out when the edges are golden brown. Share them with your coworkers, friends and family or just eat all of them yourself.](https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/)`	The `[best chocolate chip cookies](https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/)` have one cup of white sugar and one cup of brown, a whole stick of butter, and of course semisweet chocolate chips. Make sure to bake them at 350 degrees F and remember to take them out when the edges are golden brown. Share them with your coworkers, friends, and family, or just eat all of them yourself.
`[CLICK HERE FOR THE BEST CHOCOLATE CHIP COOKIES](https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/)`	Here's a recipe for `[the best chocolate chip cookies](https://www.allrecipes.com/recipe/10813/best-chocolate-chip-cookies/)` .

Overview of link types

- A **file-relative path** is a path to a file *relative to* the current file. If the reference is within the same directory, the relative path can be as simple as the name of the file.
- A **root-relative path** is a path to a file that begins at the root directory, and then paths out to the file. Contrast this path to a file-relative path, which begins with the current file as the point of origin, not the root directory.
- A **fully qualified domain name (FQDN)** includes the top-level domain, subdomain, and any hostnames used with reference to the file.
- A **uniform resource locator (URL)** is a reference to a specific resource. It is like an address for a file that is located somewhere on a network (the internet). Unlike an FQDN, a URL also includes the transmission protocol, such as *http://* (the hypertext protocol, for example).

Link priorities and examples

LINK TYPE	PRIORITY	EXAMPLE	LAY EXAMPLE
File relative path	1st	`another-file.md`	
Root relative path	2nd	`~/different-directory/another-file.md`	
Site relative URL	3rd	`/windows/uwp/get-started/get-set-up`	
FQDN / Full URL	Avoid	`https://doc.microsoft.com/help/contribute/links-how-to`	
Cross repo reference	1st (if supported)	See Cross repo reference	

Link validation

Build validation in pull requests (PR) gives just-in-time Suggestion, Warning, Error, or Validate messages. These messages can allow or prevent PRs from being merged into many of the Docs repos.

The table below outlines the levels of validation support depending on the link type.

LINK TYPE	PR	BOOKMARK	AGC
FQDN / Full URL	N	N	N
Site relative URL	N	N	Y
Root relative path	Y	Partial	Y
File relative path	Y	Partial	Y
Reference	Partial	Partial	Y
HTML URL	N	N	Partial
Cross reference	Y	--	Y

Links to articles in the same docset

To link to an article in the same docset, use a *relative path* link. A relative path is the path to the target file relative to the current file. Use the following syntax to build a relative path:

- **file.md** or **./file.md** specifies a file that's in the same directory (folder) as the current file
- **ide/file.md** or **./ide/file.md** specifies a file that's in a child directory (subfolder) named **ide**
- **../file.md** specifies a file that's in the parent directory of the current directory
- **../../file.md** specifies a file that's two directories above the current directory
- **../ide/file.md** specifies a file that's in a directory named **ide** that's a peer directory of the current directory

For example:

```
[link text](../../folder/filename.md)
```

Instead of constructing a relative path from the current file to the target file, you can start the file path at the root of the docset. This type of link starts with **~**. For example, if the docset's root folder is **docs**, the current file is **docs/ide/current-file.md**, and the target file is **docs/test/unit-test/load-tests.md**, the link is as follows:

```
[load tests](~/test/unit-test/load-tests.md)
```

> **TIP**
> Starting links with ~ produces invalid links when navigating source repositories on GitHub.

Include the **.md** file extension in all types of relative link. If you don't, the link may still work on the docs.microsoft.com site, but it won't work when you view the file on GitHub. Also, a build warning is generated at build time.

Advantages

Relative-path links (that is, links that don't start with **/** and that do end with **.md**) are validated at build time. Build validation produces a helpful warning if there's a typo in the link or if the file doesn't exist. For this reason, it's best to use a relative path for links to topics in the same docset. File-relative links are also click-navigable in the Visual Studio Code editor. However, root-relative links that start with ~ are not clickable in the editor.

Links to articles in other repos or docsets

Links to articles in other Docs repos or docsets are known as *site-relative links*.

A repo can have multiple docsets. (You can view the different docsets in a repo on OPS's Docsets tab for a repo.) You can't use a file-relative link to a file that's in the same repo but a different docset.

To create a site-relative link to an article in another Docs repo or docset, use the part of the target URL that comes after the locale code. This makes the link functional on docs.microsoft.com, in AGC environments like JEDI, and in offline books (if that docset is published as an offline book). For example, to link to the article at **https://docs.microsoft.com/windows/uwp/get-started/get-set-up**, the link syntax is **/windows/uwp/get-started/get-set-up** as shown here:

```
For more information, see [Get set up](/windows/uwp/get-started/get-set-up).
```

Don't append the file extension *.md* on a link to an article in another repo or docset.

Site-relative links aren't validated at build time like file-relative links. If you have a typo in your link, no warning is generated. Site-relative links are also not click-navigable on the staging server `review.docs.microsoft.com`. To verify that the link works, replace the branch name in the link's generated URL with `master`. You can also check the broken links report for your docset on ops.microsoft.com after you publish your changes.

Links to external sites

Use a complete URL to link to an external site, including non-docs.microsoft.com Microsoft sites.

```
[Microsoft](https://www.microsoft.com)
```

All links must be secure (`https` instead of `http`) whenever the target supports it.

Locale codes in external links

Don't include locale codes such as `en-us` in your links to MSDN, TechNet, microsoft.com, and Azure.com articles. Hard-coded locale codes prevent localized content from displaying, which is a bad customer experience for users in other locales. When you copy a URL from a browser, delete the locale code from it when you create your link.

For example:

```
[OneDrive](https://developer.microsoft.com/onedrive)
```

And not:

```
[OneDrive](https://developer.microsoft.com/en-us/onedrive)
```

In some cases, the locale code is necessary for the link to work. For example, links to videos on Microsoft Virtual Academy require a locale code. Always test your link without the locale code to be sure it functions.

Third-party site link guidance

Minimize links that send users to a non-Microsoft site. If you must link to a third-party site, use the following guidance:

- **Accountability**: Link to third-party content when it's the third party's information to share. For example, it's not Microsoft's place to tell people how to use Android developer tools—that's Google's story to tell. You can explain how to use Android developer tools *with* Azure, but Google should tell the story of how to use their tools.
- **PM signoff**: Have PMs sign off on third-party content. By linking to it, it says something about Microsoft's trust in the content, and our obligation if people follow the instructions.
- **Freshness reviews**: Make sure that the third-party info is current, correct, and relevant.
- **Offsite**: Make users aware that they're going to another site. If the context doesn't make that clear, add a qualifying phrase. For example: "Prerequisites include the Android Developer Tools, which you can download on the Android Studio site".
- **Next steps**: It's fine to add a link to, say, an MVP blog in a **Next steps** section. Just make sure that users understand they're leaving the site.
- **Legal**: Microsoft is covered legally under **Links to Third-Party Sites** in the **Terms of Use** footer on every ms.com page.

Links to bookmarks

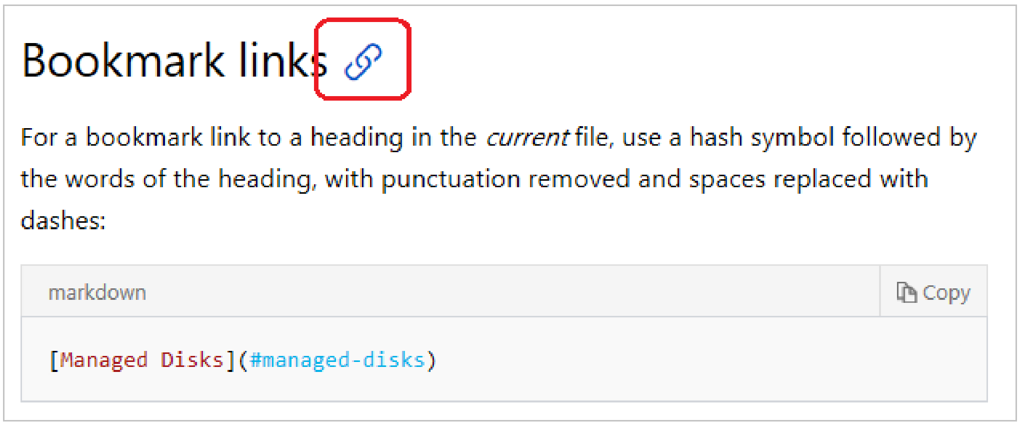
For a bookmark link to a heading in the *current* file, use a hash symbol followed by the lowercase words of the heading. Remove punctuation from the heading and replace spaces with hyphens:

```
[Managed Disks](#managed-disks)
```

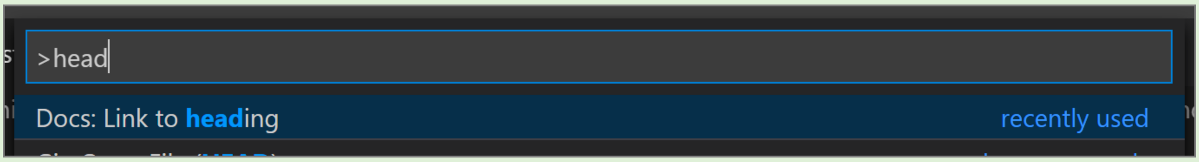
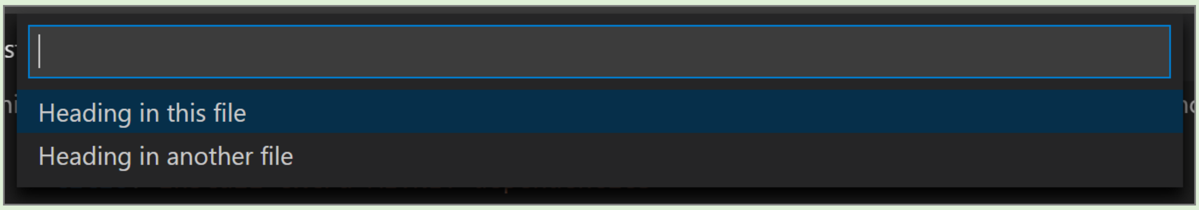
To link to a bookmark heading in another article, use the file-relative or site-relative link plus a hash symbol, followed by the words of the heading. Remove punctuation from the heading and replace spaces with hyphens:

```
[Managed Disks](../../linux/overview.md#managed-disks)
```

You can also copy the bookmark link from the URL. To find the URL, hover your mouse over the heading line on docs.microsoft.com. You should see a link icon appear:



Select the link icon and then copy the bookmark anchor text from the URL (that is, the part after the hash).

> **TIP**
>
> The Docs Authoring Pack makes it easy to add links to headings (and other types of link, for that matter). Press `F1` in Visual Studio Code to open the command palette, then type in **heading**. Select the `Docs: Link to heading` option.
>
> 
>
>
> Then, select `Heading in this file` or `Heading in another file`.
>
> 
>
>
> After you select a file, you're presented with a list of available headings to link to. Select a heading and you're done.

Explicit anchor links

Adding explicit anchor links using the `<a>` HTML tag aren't required or recommended, except in hub and landing pages. Instead, use the autogenerated bookmarks as described in bookmark links. For hub and landing pages, declare anchors as follows:

```
## <a id="anchortext">Header text</a>
```

or

```
## <a name="anchortext">Header text</a>
```

And the following to link to the anchor:

```
To go to a section on the same page:
[text](#anchortext)

To go to a section on another page.
[text](filename.md#anchortext)
```

> **NOTE**
> Anchor text must always be lowercase and not contain spaces.

Links to specific versions

To hard code a link so it always goes to a specific version of an article, add the `view` parameter with its value set to the moniker (product and version) you want to link to, and then add `&preserve-view=true` so Docs knows you've set the version explicitly and it shouldn't be overridden. For example, the following URL will always resolve to the SQL Server 2019 (v15) version of the article:

```
https://docs.microsoft.com/sql/t-sql/functions/cast-and-convert-transact-sql?view=sql-server-ver15&preserve-view=true
```

If you set the view parameter without adding `&preserve-view=true`, Docs won't know that the view was set explicitly, and the version may be overridden based on the default version and the user's selections.

XRef (cross reference) links

XRef links are the recommended way to link to APIs, because they're validated at build time. Before using XRef links in your article, make sure that XRef-style links are enabled for your docset.

To link to autogenerated API reference pages in the current docset or other docsets, use XRef links with the unique ID (UID) of the type or member. UIDs are case-sensitive.

Check if the API you want to link to is on docs.microsoft.com by typing all or some of its full name in the .NET API browser or Windows UWP search box. If you don't see any results displayed, the type isn't yet on docs.microsoft.com.

You can use one of the following syntaxes:

- Auto-links

  ```
  <xref:UID>
  <xref:UID?displayProperty=[chosen display property]>
  ```

  Example: `<xref:System.String>`

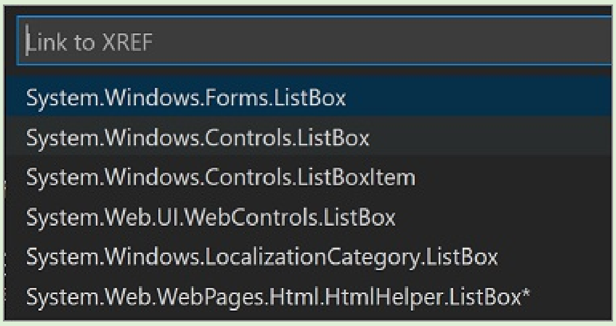
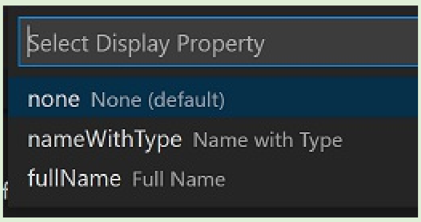
For more information about the `displayProperty` parameter, see Display properties.

- Markdown-style links

  Use Markdown-style xref links when you want to customize the link text that's displayed beyond the qualified or unqualified API name.

  ```
  [link text](xref:UID)
  ```

  Example: `[String class](xref:System.String)`

> **TIP**
>
> You can use the Docs Markdown extension for VS Code (part of the Docs Authoring Pack) to insert .NET XRef links (that is, those surfaced in the .NET API Browser) into Markdown files.
>
> 1. Press **F1** in Visual Studio Code to open the command palette, then type in **xref**. Select the **Docs: Link to Xref** option.
>
> 
>
>
> 2. Type in part or all of the API's name, for example, **listbox**, and then press **Enter**. A list of matching APIs is shown.
>
> 
>
>
> 3. Select the desired API or type in some text to further filter the list.
> 4. Select a display property.
>
> 
>

Display properties

By default, link text for auto-links shows only the unqualified API name. Add the optional `displayProperty` query parameter to partially or fully qualify the displayed API name.

DISPLAY PROPERTY	DESCRIPTION	EXAMPLE
`fullName`	Displays the fully qualified API name.	System.String.Trim()
`nameWithType`	Displays **namespace.type** for types and **type.member** for type members.	String.Trim()

XRef build warnings and incremental builds

An incremental build only builds files that have changed or been affected by a change. If you see a build warning about an invalid XREF link, but the link is valid, this could be because the build was incremental. The file causing the warning didn't change, so it wasn't built and past warnings were replayed. The warning will disappear when the file changes or if you trigger a full build. This is a drawback to incremental builds, because DocFX can't detect a data update inside the XREF service.

Determine the UID

The UID is usually the fully qualified class or member name. There are at least two ways to determine the UID:

- Right-click on the Docs page for a type or member, select **View source**, and then copy the **content** value for **ms.assetid**:

```html
<meta name="page_type" content="dotnet" />
<meta name="page_kind" content="class" />
<meta name="description" content="Represents text as a sequence of UTF-16 code units. " />
<meta name="toc_rel" content="_splitted/System/toc.json" />
<meta name="source_url" content="" />
<meta name="ms.assetid" content="System.String" />
<meta name="pdf_url_template" content="https://docs.microsoft.com/pdfstore/en-us/VS.dotnet-api-docs/{branchName}{pdfName}" />
<meta name="search.mshattr.devlang" content="csharp vb cpp" />
```

- Use the autocomplete site by appending some or all of the name of the type to the URL. For example, entering `https://xref.docs.microsoft.com/autocomplete?text=Writeline` in the address bar of your browser displays all the types and methods that contain **Writeline** in their name, along with their UID.

Verify the UID

To test if you have the correct UID, replace **System.String** in the following URL with your UID, and then paste it into the address bar of a browser:

https://xref.docs.microsoft.com/query?uid=System.String

> **TIP**
>
> The UID in the URL is case-sensitive, and if you're checking a method overload UID, don't include spaces between the parameter types.

If you see something like the following, you have the correct UID:


```
[{"uid":"System.String","name":"String","fullName":"System.String","href":"https://docs.microsoft.com/dotnet/api/system.string","tags":",/dotnet,netframework-4.5.1,netframework-4.5.2,netframework-4.5,...xamarinmac-3.0,public,","vendor":null,"hash":null,"commentId":"T:System.String","nameWithType":"System.String"},
{"uid":"System.String","name":"String","fullName":"System.String","href":"https://docs.microsoft.com/dotnet/api/system.string","tags":",/dotnet,netframework-4.5.1,netframework-4.5.2,netframework-4.5,netframework-4.6,netframework-4.6.1,...,xamarinmac-3.0,public,","vendor":null,"hash":null,"commentId":"T:System.String","nameWithType":"System.String"}]
```


If you just see [] displayed on the page, you have the wrong UID.

HTML encoding

Special characters in the UID need to be HTML encoded as follows:

CHARACTER	HTML ENCODING
`	%60
#	%23

CHARACTER	HTML ENCODING
*	%2A

See a full list of HTML character codes.

Encoding examples:

- **System.Threading.Tasks.Task`1** encodes as `System.Threading.Tasks.Task%601` (see the section on generic types)
- **System.Exception.#ctor** encodes as `System.Exception.%23ctor`
- **System.Object.Equals*** encodes as `System.Object.Equals%2A`

Generic types

Generic types are those types such as **System.Collections.Generic.List<T>**. If you browse to this type in the .NET API browser and look at its URL, you see that **<T>** is written as **-1** in the URL, which actually represents **`1**:

https://docs.microsoft.com/dotnet/api/system.collections.generic.list-1

To link to a generic type such as **List<T>**, encode the ` backtick character as **%60**, as shown in the following example:

```
<xref:System.Collections.Generic.List%601>
```

Methods

To link to a method, you can either link to the general method page by adding a * after the method name, or to a specific overload. For example, use the general page when you want to link to the Object.Equals method without specific parameter types. The asterisk character * is encoded as **%2A**. For example:

<xref:System.Object.Equals%2A?displayProperty=nameWithType> links to Object.Equals

To link to a specific overload, add parenthesis after the method name and include the full type name of each parameter. Don't put a space character between the type names or the link won't work. For example:

<xref:System.Object.Equals(System.Object,System.Object)?displayProperty=nameWithType> links to Object.Equals(Object, Object)

Reference-style links

You can use reference-style links to make your source content easier to read. Reference-style links replace inline link syntax with simplified syntax that allows you to move the long URLs to the end of the article. Here's Daring Fireball's example:

Inline text:

```
I get 10 times more traffic from [Google][1] than from [Yahoo][2] or [MSN][3].
I start my morning with a cup of coffee and [The New York Times][NY Times].
```

Link references at the end of the article:

```
<!--Reference links in article-->

[1]: http://google.com/
[2]: http://search.yahoo.com/
[3]: http://search.msn.com/
[ny times]: http://www.nytimes.com/
```

Make sure that you include the space between the colon and the link. If you forget to include the space, links to other technical articles are broken.

Links inside HTML markup

If your Markdown file includes HTML markup, you can't use Markdown syntax for links inside that HTML markup. Instead, use HTML anchor tags.

Correct:

```
<a href="https://www.microsoft.com">link text</a>
```

Incorrect:

```
[link text](https://www.microsoft.com)
```

Links to Azure PowerShell reference content

The Azure PowerShell reference content has been through several changes since November 2016. Use the following guidelines for linking to this content:

Structure of the URL

- For cmdlets:

  ```
  /powershell/module/<module-name>/<cmdlet-name>[?view=<moniker-name>]
  ```

- For conceptual topics:

  ```
  /powershell/azure/<topic-file-name>[?view=<moniker-name>]
  /powershell/azure/<service-name>/<topic-file-name>[?view=<moniker-name>]
  ```

The <moniker-name> portion is optional. If it's omitted, you'll be directed to the latest version of the content. The <service-name> portion is one of the examples shown in the following base URLs:

- Azure Active Directory (AzureAD) PowerShell content: https://docs.microsoft.com/powershell/azure/active-directory
- Azure Service Fabric PowerShell: https://docs.microsoft.com/powershell/azure/service-fabric
- Azure Information Protection PowerShell: https://docs.microsoft.com/powershell/azure/aip/overview
- Azure Elastic DB Jobs PowerShell: https://docs.microsoft.com/powershell/azure/elasticdbjobs

When you use these URLs, you're redirected to the latest version of the content. This way, you don't have to specify a version moniker, and you don't have to update the link when the version changes.

To create the correct link, find the page that you want to link to in your browser, and copy the URL. Then, remove "https://docs.microsoft.com" and the locale info.

When you're linking from a table of contents, you must use the full URL without the locale information.

Example links:

```
[Get-AzureRmResourceGroup](/powershell/module/azurerm.resources/get-azurermresourcegroup)
[Get-AzureRmResourceGroup](/powershell/module/azurerm.resources/get-azurermresourcegroup?view=azurermps-4.1.0)
[New-AzureVM](/powershell/module/azure/new-azurevm?view=azuresmps-4.0.0)
[New-AzureRmVM](/powershell/module/azurerm.compute/new-azurermvm)
[Install Azure PowerShell](/powershell/azure/install-azurerm-ps)
```

Backlinks from user interfaces and product portals

To link to Docs pages from web portals and other in-product user interfaces, there are special considerations such as using FWLinks and campaign codes for tracking. For more information, see Link to articles from the user interface.

FWLinks and aka.ms links

FWLinks and aka.ms links (known collectively as ShortLinks) both shorten and redirect URLs. You can use these links in web portals and other in-product user interfaces, in some situations. For more information, see Add the right type of link.

Caution

Don't use FWLinks and aka.ms links in docs.microsoft.com content. Automation periodically converts links like this to full URLs to improve SEO and be verifiable by platform tools such as the OPS Broken Links report. Use FWLinks only as a last resort to point to content outside of docs.microsoft.com (for example, when you link to an external page that doesn't yet have a URL, or when you link to an external page and you know its URL will change). If you must use an FWLink, include the "/p/" parameter to make it a permanent redirect, as in the following example: `[https://go.microsoft.com/fwlink/p/?LinkId=389595](https://go.microsoft.com/fwlink/p/?LinkId=389595)`. Using this parameter doesn't affect your ability to later update the FWLink.

Link to content from the user interface

7/14/2021 • 13 minutes to read

This article describes the process of creating context-sensitive help (CSH) links for a user interface (UI). For other types of contextual help, such as tooltips and help bubbles, see the Cloud Style Guide section on UX/UI content design.

If you need help with your content integration strategy, contact Sarah Luck or Miriam Lottner.

Overview

Some users may need help with a concept or procedure while using a product or service. To address this need for just-in-time assistance, add a CSH link to the UI. The link targets core content that typically opens in a new browser tab. This type of integrated help is designed to minimize on-page explanation while providing a resource that helps users move forward with a task.

The type of link you choose depends on where you're using the link:

- For most products and services, you'll use either an FWLink or a hard-coded full link to `https://docs.microsoft.com/<path/to/your/content>`. For more information about which option to choose, see Add the right type of link later in this article.
- For Microsoft 365, use the MARVELCSH system for all context-sensitive help links. For more information, see the published onboarding guidance.

To implement a CSH linking strategy for your product, service, or feature, follow these steps:

1. Understand the context in which the link appears.
2. Choose the best content as your help source.
3. Determine the best link text to use, considering accessibility requirements.
4. Add the right type of link and reference it in code.
5. Use campaign IDs for tracking.
6. Monitor traffic and satisfaction to ensure the links are working well.

Understand context

Links that target content are intended to provide help for the current user context. We can't yet customize the content experience for the user context based on factors such as tenant lifecycle, subscription type, or last action. Today, context is limited to the location in the UI and our intuition of the user's mindset and intent.

When in the UI, users are in task-completion mode. Our goal is to keep them on task and in the UI, so don't interrupt that flow with unnecessary clicks out of the workspace. Think of help resources in the UI as a safety net – it's great it's there to catch the users who stumble, but even better if they don't need help at all. If a user clicks on a help link, it's likely they're confused, unsure, or blocked by not understanding next steps or how features work together.

Linked help content should immediately deliver the information the user needs to complete the task. Use the smallest, most unobtrusive type of help for the current context. Most of the time, you don't need to link to content. You can use placeholder text or a help bubble to explain correct formatting or a tooltip to introduce an unfamiliar term. Well-written strings coupled with intuitive workflows go a long way toward reducing the need for additional help.

Example links in context

The following screenshots illustrate some examples of link placement in the UI.

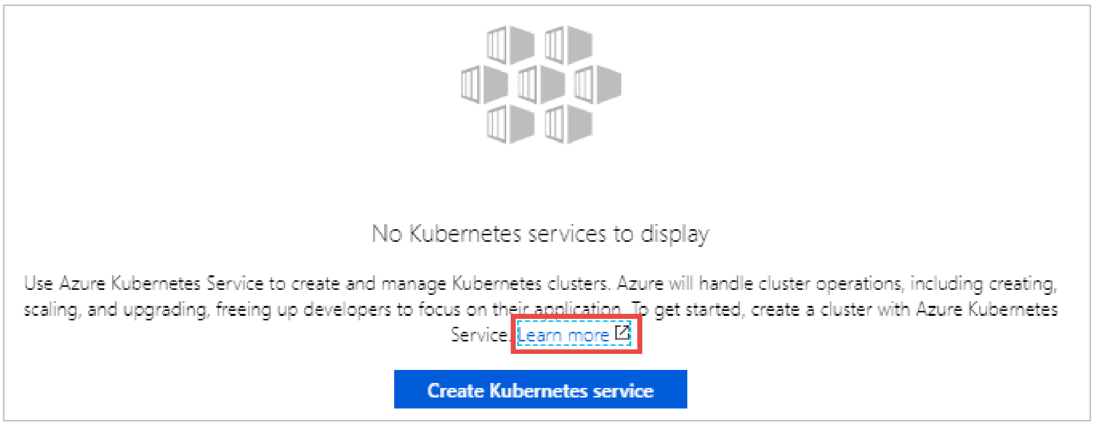


Figure 1: Learn more link in Azure service empty state

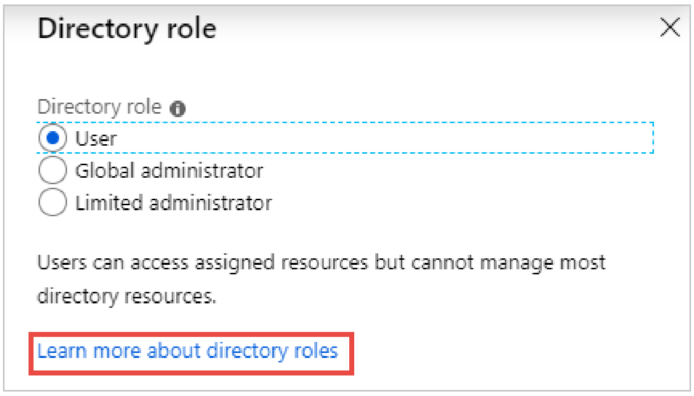


Figure 2: Learn more link in Azure form footer

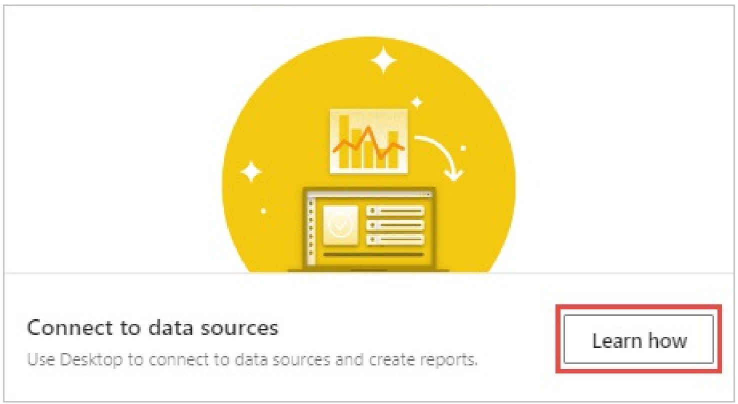


Figure 3: Learn how link in Power BI getting started

Choose the best content

Links embedded in the UI should be specific, targeted, and task-oriented. When possible, choose content on docs.microsoft.com, the authoritative site for product-related technical content.

When you choose a link target, consider the location in the UI and the user intent. Why did you want the user to click there and what information do they need right now? If you're linking to an article in documentation, it's often best to link to a specific section to go right to what the user needs to know. Linking to a specific section minimizes scrolling and unnecessary overview and instruction. Follow these general guidelines:

DO	DON'T
Work with writers, designers, and engineering teams to design the contextual help strategy and select link targets.	Duplicate content or create one-off processes.
Link to articles, article sections, Learn modules, and other content on docs.microsoft.com	Link to destinations that aren't maintained to follow the product lifecycle.
Link to azure.com if connecting to Azure pricing information.	Link to azure.com for product overviews or technical information.

Determine the best link text

Links in the UI must follow accessibility guidelines and be optimized for screen readers. There are three elements to link text:

- The visible label on the page
- A tooltip that appears on hover
- ARIA text, from a suite of web standards ("Web Accessibility Initiative - Accessible Rich Internet Applications") to make the Web accessible to people with disabilities

Screen readers always narrate the tooltip and append it to the other labels. If no ARIA text is present, the visible label is narrated. When ARIA text is present, screen readers read that text instead of the visible label.

Visible label

Use a short phrase as link text that provides some indication of the link destination. For a sighted user, the phrase "learn more" when appended to a string is fine. The string provides the necessary context. However, a vision-impaired user relies on a screen reader to identify links on the page. If there's more than one link on the page that's labeled "learn more", it can be confusing. It's a best practice to include a phrase that says what the user will learn more about. Follow these guidelines for visible link text:

- Keep it short, but give a clue as to where the link will take you.
- Use "Learn more" as stand-alone text when targeting generic content, like an overview.
- Use "Learn more about *something*" when targeting intent-focused content, like a tutorial or a Learn module.
- Only use the "learn more" phrase when the link targets technical content.
- If there's more than one "learn more" link on the page, use link-specific text.

For more information, see "Link text" in the Microsoft Cloud Style Guide.

Tooltip

Tooltips DO NOT repeat the visible label. They provide helpful information for all users. Include a tooltip with link text to let readers know how the link will behave. The tooltip is shown on hover to sighted users and is appended to either the visible label or the ARIA text for screen reader narration. The tooltip can read "Content opens in a new window" or "Content opens in a new tab".

For more information, see "Tooltip guidelines" in the Microsoft Cloud Style Guide.

ARIA text

ARIA text doesn't display in the UI, but it's an important part of meeting accessibility standards. ARIA text is read by screen readers instead of the visible label. Think of ARIA text as "screen reader text." When ARIA text is present, a screen reader will narrate the ARIA text + the tooltip. To author ARIA text, follow these guidelines:

- Avoid language that assumes you can see, including directional terms and phrases such as "show" or "take a look".

- Avoid language that assumes you can use your hands.
- Use end punctuation to cause the screen reader to pause briefly and to avoid narrated text that runs together.
- Use text instead of symbols, which may be narrated differently by different screen readers.
- Don't just repeat the visible label. If your visible label says "learn more", the context may be understood by a sighted user because of what surrounds the text. Don't make that assumption for ARIA text – always include a full expression, such as "Learn more about securing resources with role-based access control."

Add the right type of link

We've always recommended FWLinks for in-product linking because they were designed for this purpose. Hard-coding full URLs (`https://docs.microsoft.com/<path/to/your/content>`) can result in issues that require a product or service update to correct. When you use an FWLink, you can change the link target easily and quickly, with new destinations live within minutes. Making changes to hard-coded URLs is definitely quicker in services than on-premises products, but it can still take time and typically requires a developer. For more information, see FWLinks and aka.ms links.

The introduction of air-gapped clouds (AGC) like the one used for Joint Enterprise Defense Infrastructure (JEDI) presents a problem because there is currently no redirection service available in the AGC to handle FWLinks or aka.ms links. There is a version of Docs in the AGC, but the FWLinks or aka.ms links don't resolve to the targets. In other words, these links return a 404.

There is now a plan to address this issue, but no ETA at this time (May 2021). In addition to the issue with FWLinks and aka.ms links, full URLs work only if the service handles them correctly. See the following table for details, and contact Sarah Luck or Michael Blythe if you or your engineering team needs more information.

PRODUCT OR SERVICE	CONSIDERATIONS / RECOMMENDATIONS
Azure portal	Use full URLs. The Azure portal tokenizes the domain name and replaces it with the AGC domain in those environments.
On-premises products	Use FWLinks
Services that aren't in the AGC	Use FWLinks
Services that are in the AGC	Given the plan for a redirection service in AGC, consider continuing to use FWLinks. If you want links to work today, the service needs to do two things: • Use full URLs • Rewrite the domain of all docs.microsoft.com links to the appropriate air-gapped domain. The Azure portal already does this, and we're checking with other services. Carefully evaluate whether the effort to update current FWLink and aka.ms links makes sense given the plan for AGC.
Microsoft 365	Use the MARVELCSH system

Use campaign IDs for tracking

A campaign ID is a query string that is appended to a URL to track sessions from a particular traffic source. Campaign ID query strings use this syntax: ?wt.mc_id=*your_unique_campaignID_here*, and they can be added to all CSH links - full URLs, FWLinks, and aka.ms links.

When applied to CSH links, a campaign ID is used to filter data for reporting, to track user actions for the session, and includes information about where the link is located. **We strongly recommend using campaign IDs to get detailed reporting about referral traffic from products and services.** To assign a campaign ID to your link target, first generate the ID using the In-Product Tag Generator. For more information about campaign IDs, contact James M. Kim. To plan out large-scale use of campaign IDs, contact Sarah Luck or Miriam Lottner.

Monitor and maintain

After your CSH links are live, track performance to make sure you're achieving the results you want. Make sure content is maintained or updated based on where the link appears and what you think the customer needs.

Track performance

Use content reports to track performance. To isolate traffic, SAT, and verbatim to CSH views, filter the referrer to your product or service, like `portal.azure.com`. You can also look for a campaign ID in the destination URL. These reports can help you figure out whether you've selected the best content for this link. A high number of clicks on a CSH link as a percentage of users to the UI could indicate a problem with the user experience. You can also export traffic data for selected FWLinks and aka.ms links in the link manager tool.

For more information about contextual help metrics and the actions that data drives, see this presentation about CSH.

Maintain link health

Broken links, especially in user experiences, aren't just annoying, they look unprofessional. A resource that was supposed to help the user has become a source of frustration. As a link owner, monitor the occurrence of 404s and make corrections as quickly as possible.

To avoid broken links, update associated FWLinks before retiring content and when changing content titles or section labels. If you're not sure whether content is the target of a UI link, search the FWlink and aka.ms link system for the target URL to identify the associated links to update. If you're using a full URL to the content, update the URL in the product or service and add a redirect in docs.microsoft.com, as you would for any retired content.

FWLinks and aka.ms links

Both FWLinks and aka.ms links shorten and redirect URLs, but they were designed for different purposes. Aka.ms links provide an easy-to-remember URL for marketing and social-media campaigns. FWLinks were designed for product-to-web linking to avoid hard-coding the destination URL within a product's UI. FWlinks provide additional functionality for localization, but we typically don't use this functionality anymore for links to docs.microsoft.com. Either type of link will work from the UI, but we'll focus on FWLinks in this article.

Before you get started, read Add the right type of link to understand the implications of using FWlinks in some services. Also consider how you'll manage the set of FWLinks for your product or service. We've seen many instances of out-of-date and duplicate links; work with your partners to make sure your approach to linking and link ownership is maintainable. For more information about link maintenance, contact Kelly Pittman.

Create an FWLink

An FWLink has the following format: `https://go.microsoft.com/p/fwlink/?linkid=` `<number ID>` (where the system generates the numerical ID when you create the FWLink). Follow the instructions below to create an FWLink. **Always include the "/p/" parameter to make the redirect permanent**. Using this parameter doesn't affect your ability to later update the FWLink.

1. Go to the AKA Link Manager tool.
2. Set **Host** to `go.microsoft.com`, and then select **New Link**.

3. Complete the link details as shown in the tables below.
4. Select **Create**. If you get an error about a duplicate link, add details to the optional fields to make the link unique.
5. The new link is displayed in the summary of links owned by you. Use the **copy to clipboard** icon to copy the link.

> **TIP**
> If you need to create or edit many links, download the Excel bulk tool add-in from the AKA Link Manager tool.

Link details

Asterisk denotes required field.

FWLINK FIELD	VALUE	NOTE
Target URL*	The URL that should open when the link is clicked.	Make sure that the target doesn't include **en-us** (or any other language locale). You can link to a bookmark to target the relevant content section.
CLCID (Country Locale ID)	Leave as default	
PRD (Product)*	Select the appropriate product, like Microsoft Azure (12177), Dynamics 365 (10965), or Office 365 (12413)	
PVER (Product Version)*	Leave as default	
PLCID (Product Locale ID)	Leave as default	
AR (Area)*	CSH	Describes the property of where the FWLink will be used (for example, *Settings app*).
SBA (Sub Area)*	General type of link (for example, Learn more)	Describes the topic or page where the FWLink will be used (for example, *Bluetooth settings* or *Turn on Bluetooth*)

Link owners

FWLINK FIELD	VALUE	NOTE
Security Group Alias (as a Link Owner)	Use a common security group if you have one, like azureportallinks, wdglinks, or ddfwlink. If you don't know of a relevant security group, check with other writers in your area or the product team.	The Link Manager tool lets you view only links owned by you or by security groups in which you have membership. If your team's group is the link owner, you limit who can view and maintain the link.
Co-Owner Aliases	User aliases separated by semicolon.	The feature PM and the documentation owner are recommended. Owners are automatically notified via email whenever an FWLink is updated.

FWLINK FIELD	VALUE	NOTE
Make this link Active	Enabled	
Allow Parameters	Optional	If you turn on this feature, you can forward query string parameters from your FWLink to your target URL.

Optional fields

Field usage is not validated. Use as many of the optional fields as you want to leave an historical trail for the link and to provide options for reporting.

FWLINK FIELD	VALUE	NOTE
Category	CSH-Static	
Description	Optional	Recommended to include some text like "link from portal VM create to procedural doc."
SBP (Sub Product)	Name of your product or service	
OS (Operating System)	Leave as default	
OLCID (OS Locale ID)	Blank	
OVER (OS Version)	Blank	
Mobile URL	Optional	Use only if the mobile URL is different from the target URL.
Optional Field 1	Documentation	
Optional Fields 2-5	Optional	You can use these fields to provide additional information about the link, such as the link text or whether it's part of an experiment.

Related resources

For more information about integrating content with user interfaces, see these resources:

- Microsoft Cloud Style Guide
- Microsoft 365 Context Sensitive Help

If you need help with your content integration strategy, contact Sarah Luck or Miriam Lottner.

Guidelines for using bullets and numbered lists in content

5/10/2021 • 2 minutes to read

For basic elements of grammar, like numbered and bulleted lists, we follow the Microsoft Writing Style Guide and Microsoft Cloud Style Guide.

In addition to that guidance, there are a limited number of exceptions or clarifications for numbered lists.

When should I use numbered lists?

You should use numbered lists for:

- Portal-based procedures
- Procedures that rely heavily on screenshots

The following is an example of using numbered lists for portal-based procedures or procedures that rely heavily on screenshots.

```
1. View your repo's branches by selecting **Repos** > **Branches** while viewing your repo on the web.

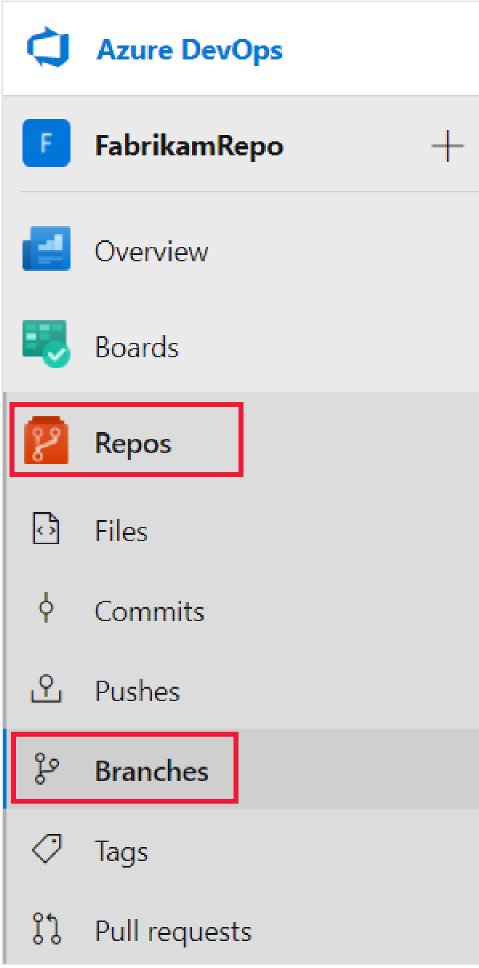
   :::image type="content" source="media/contribute-how-to-mvc-bullet-list/repos-branches.png" alt-text="Screenshot that shows how to view your branches.":::

1. Select the **New branch** button in the upper right corner of the page.

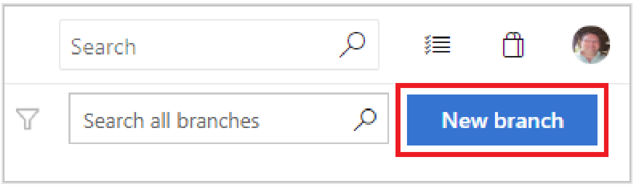
   :::image type="content" source="media/contribute-how-to-mvc-bullet-list/create-branch.png" alt-text="Screenshot that shows the New branch button.":::
```

This example renders as follows:

1. View your repo's branches by selecting **Repos** > **Branches** while viewing your repo on the web.



2. Select the **New branch** button in the upper right corner of the page.



When should I not use numbered lists?

Don't use numbered lists for:

- CLI-based procedures
- Any other procedure where each step is a block of code

The following is an example of document steps in a CLI-based procedure.

````
Use the `branch` command to create the branch and `checkout` to swap to that branch.

    ```console
    git branch feature1
    git checkout feature1
    ```
````

This example renders as:

Use the `branch` command to create the branch and `checkout` to swap to that branch.

Console

```
git branch feature1
git checkout feature1
```

General

- You should only number tutorials in the TOC when they are sequential.
- You shouldn't number article headings (H2 or H3 headings).

Tutorials

Tutorials include a special list to outline what you will learn in the article. This list requires specific format that is found in the templates.

Sample bulleted list for tutorials:

- OS disks and temporary disks
- Data disks
- Standard and Premium disks
- Disk performance

Include reusable content in articles

6/24/2021 • 4 minutes to read

You can embed other Markdown files within a Markdown file by using the Markdown INCLUDE syntax. This is useful for shared content, such as a note or warning you want to include in multiple files. You can include other file content either as a block or inline with other text.

Includes syntax

```
Block include is on its own line:

[!INCLUDE [<title>](<filepath>)]

Inline include is within a line: [!INCLUDE [<title>](<filepath>)]
```

- The `<filepath>` is the relative path to the file.
- The `<title>` is a label to provide context to the author. It's arbitrary and isn't visible to the reader, but could be useful for the content author to note context of the include file.

Here's an example of a block included file:

```
[!INCLUDE [notes](../notes.md)]
```

`INCLUDE` must be capitalized and there must be a space before the `<title>`:

```
```markdown
[!INCLUDE [notes](../notes.md)]
```
```

Guidelines for using includes

Includes allow you to write, review, and localize content once, then use it in multiple places. This practice can save time and money and help ensure consistent wording across articles.

Do use include files:

- For boilerplate content such as notes and warnings that apply in many scenarios.
- For steps that apply in many procedures.
- For announcements that apply to an entire content set, such as deprecation notifications or other breaking changes.
- For content that requires legal review and must have precise wording.

Don't create include files:

- Before you're ready to use them. Uncalled includes add clutter to a repo and may be removed.
- To be called from outside documentation, for example from a product UI. A new feature, chromeless zones, is being developed for this scenario.
- For content that only needs to be in one file.
- For content that only needs to be in a few files, unless there's a specific need for exact consistency, as in the

case of some legal text.
- For content that is longer than one or two paragraphs.
- For content that is shorter than one sentence.
- For single words or phrases, such as product names. This practice results in many small Markdown files that are difficult to manage.
- As the only content of an article. Includes are meant to be supplemental to the content in the rest of the article.

Other requirements and considerations:

- Includes won't be rendered in the GitHub rendered view of your article because they rely on Docs extensions. They'll be rendered only after publication.
- Write all the text in an include file in complete sentences or phrases that don't depend on preceding or following text in the article that references the include. Ignoring this guidance creates an untranslatable string in the article.
- Don't embed include files within other include files.

Folder structure for included files

All included Markdown files should be stored in an `/includes` folder within the docset. This way they can easily be identified and exempted from build functionality that doesn't apply to included files. Be sure to add the `/includes` folder to the excluded content array in the docfx.json file for the docset:

```
"content": [
  {
   ...
    "exclude": [
      "**/obj/**",
      "**/includes/**",
          "README.md",
          ...
    ]
  }
]
```

Content guidelines for included files

To render properly on the published page, included files must meet the following requirements:

- Can't contain an H1 heading (`#`).
- If included inline, can't contain any block formatting, such as headings or alerts.
- Can't use an absolute path for `<filepath>`.

Using images in included files

Because `/includes` folders are excluded from build as described above, images stored in `/includes` folders and referenced in included files won't be displayed in published content. Store images in a `/media` folder outside the `/includes` folder.

As with regular articles, don't share media between include files. Use a separate file with a unique name for each include and article. Store the media file in the media folder that's associated with the include

Includes metadata

Included files should contain a subset of the standard article metadata. For more information, see Metadata for included files.

Cross-repo includes

Using cross-repo reference (CRR), you can configure dependent repositories. This configuration allows .md files from an outside repo to be built or used as includes as if they are in the same local repo.

1. Locate the **.openpublishing.publish.config.json** file in the root of the local repo to edit it.

2. Add the following JSON configuration to point to the outside repo to consume the include file(s) from. Choose a friendly name as the identifier for the repo. Set the URL of the outside repo, and the branch name to pull the files from.

```
"dependent_repositories": [
  {
    "path_to_root": "friendly_name",
    "url": "https://github.com/MicrosoftDocs/outside_repo/",
    "branch": "master",
    "branch_mapping": {}
  }
]
```

   For more examples, see azure-docs-pr.

   For more information on the configuration format, see Cross-Repository Reference.

   > **WARNING**
   >
   > If the dependent repository is moved, the branch is renamed, or the include file is moved, the builds of the docset will show a build warning. It can be confusing to interpret the warning since the problematic missing file location is not local to the docset that was being built.

3. In the local repo, edit the .md file where you want to use the included outside file. Use the `~/friendly-name/` token to reference the dependent repo.

```
[!INCLUDE[My included file](~/friendly_name/folder/include-file.md)]
```

Tabbed Conceptual

5/12/2021 • 5 minutes to read

Tabs enable conceptual content that is multi-faceted. They allow sections of a document to contain variant content renderings, each covering the same core concept or task, but allowing for customer reading preferences on tooling, platform, etc. This feature provides richer information to customers and should also improve writer productivity, reducing maintenance by eliminating duplicate content.

In this article you will:

- See the tab experience and learn how to use it in your content.
- Discover how to configure dependencies between different tab groups.
- Understand the limitations and future plans for tabs.
- Receive guidance for how to appropriately leverage tabs.
- Learn about approved tabs and the proposal process.

Basic Usage

Here's an example of the tab experience:

- Linux
- Windows

Content for Linux...

The above tab group was created with the following syntax:

```
# [Linux](#tab/linux)

Content for Linux...

# [Windows](#tab/windows)

Content for Windows...

---
```

Tabs are indicated by using a specific link syntax within a Markdown header. The syntax can be described as follows:

```
# [Tab Display Name](#tab/tab-id)
```

A tab starts with a Markdown header, `#`, and is followed by a Markdown link `[]()`. The text of the link will become the text of the tab header, displayed to the customer. In order for the header to be recognized as a tab, the link itself must start with `#tab/` and be followed by an ID representing the content of the tab. The ID is used to sync all same-ID tabs across the page. Using the above example, when a customer selects a tab with the link `#tab/windows`, all tabs with the link `#tab/windows` on the page will be selected. Try it out below and compare to the tab selection above.

- Linux
- Windows

Content for Linux...

The first instance of a link-header with the above syntax not only defines a tab, but defines the start of the group of tabs. Each successive link-header with this format, starts a new tab within the current group. The group itself is terminated with a Markdown horizontal rule `---`.

> **NOTE**
>
> When a customer selects a tab, the url's query string will be updated to capture the selection and when a customer navigates to a page with these query string values, the appropriate tabs will be auto-selected. This means that you can link to an article with a particular set of tabs pre-selected and customers can bookmark particular combinations as well.

Limitations of Tabs

1. You **can** use includes inside of a tab's content, but you **cannot** include a Markdown file within the content of a tab that itself contains tabs.
2. You **cannot** use H1s or H2s inside the content of a tab.
3. You **cannot** use tabs *inside of* nested includes.

Dependent Tabs

It's possible to make the selection in one set of tabs dependent on the selection in another set of tabs. Here's an example of that in action:

- Azure CLI
- PowerShell
- Portal

Azure CLI content for Linux...

Notice how changing the Linux/Windows selection above changes the content in the Azure CLI and PowerShell tabs. This is because the tab group defines two versions for each Azure CLI and PowerShell, where the Windows/Linux selection above determines which version is shown for Azure CLI/PowerShell. Here's the markup that shows how this is done:

```
# [Azure CLI](#tab/azure-cli/linux)

Azure CLI content for Linux...

# [Azure CLI](#tab/azure-cli/windows)

Azure CLI content for Windows...

# [PowerShell](#tab/azure-powershell/linux)

PowerShell content for Linux...

# [PowerShell](#tab/azure-powershell/windows)

PowerShell content for Windows...

# [Portal](#tab/azure-portal)

Azure Portal content, independent of platform...

---
```

Dependent tabs are indicated by using an extension of the original tab link syntax. The syntax can be described as follows:

```
# [Tab Display Name](#tab/tab-id/dependency-tab-id)
```

To set up a dependent tab, you would create multiple versions of the header and add a second URL segment to the tab link, targeting the ID of the dependency. For example, both `#tab/azure-cli/linux` and `#tab/azure-cli/windows` represent the singular Azure CLI tab. Which one is displayed is based on whether linux or windows is selected above.

> **IMPORTANT**
>
> Remember that the segments in the tab link are IDs. In order for selection synchronization to work, the IDs bust be the same across tab groups and dependent tab groups.

Limitations of Dependencies

1. For dependent tab groups to function properly, the dependency must appear in the document prior to the dependent group.
2. You can only have one level of dependencies.

Guidance

Tabs are intended to represent multiple variations of the same content, based on different ways of approaching the task or problem being discussed. For example, if the article is "Create a Virtual Machine", then there are many ways to do this. Is it a Linux or Windows VM? Do you want to create it with Azure CLI, PowerShell, or directly through the Portal?

Here are a few things to use tabs for:

- **Do** use tabs for showing differences between platforms for the same task.
- **Do** use tabs for showing different tooling approaches for the same task.

> **NOTE**
>
> Make sure to order your tabs so that the tab you want to be the default selection is always first.

Here are a few things **not** to use tabs for:

- **Do not** use tabs to show ways to accomplish a task for different versions of the same product. Use the native versioning support of docs.
- **Do not** use more than four tabs. This will cause a terrible mobile experience. We hope to address this in the future.
- **Do not** use tabbed conceptual in combination with the `selector` or `multi-selector` Markdown extensions. While we are not deprecating the selector extensions, we think you'll find that the tab experience handles these scenarios better in most cases, and we encourage you to investigate migrating content over to this new approach.

Here is a **soft rule**:

- **Try not to** use tabs to show different programming language examples for the same task. First, consider using the native dev_langs selector for this.

Approved Tabs

Although we don't currently validate tab values, we recommend that you use existing values if possible. See the tab allowlist for existing values.

If none of these values are appropriate you can create new values and add them to the list via a PR. However, please do so sparingly. The strategy for content switchers is under review and is likely to change in the future.

Combining Rule

In some scenarios, content may be the same for multiple tabs. An example of this is iOS and macOS or Browser and Node.js. Rather than inventing new tabs for these scenarios, you can combine approved tabs together. The rule can be described as follows:

```
# [Display Name / Display Name / Display Name](#tab/tab-id+tab-id+tab-id)
```

- The display names should be concatenated using a `/`. There should be a single space on each side of the `/`.
- The tab IDs should be concatenated with a `+`. No spaces should be found anywhere.
- The order of concatenation should always follow the order that the items appear in the approved list.
- You cannot combine the `Other` tab.

Examples

- `[macOS / iOS](#tab/macos+ios)`
- `[Browser / Node.js](#tab/browser+nodejs)`

Converting SMS content to markdown

5/12/2021 • 2 minutes to read

Prerequisites:

Install pandoc on your machine http://pandoc.org/installing.html

Steps:

1. Log into SMS and switch to batch mode
2. Select each of the articles in your target dev center using the left navigation pane
3. Click “Export” and save the resulting CSV to a file.
4. Open PowerShell ISE and paste the following script into the script editor. This will output HTML files to a new directory called MSDN.

```
function getHTMLFromSmsExport($file) {
    cd $PSScriptRoot
    $export = (import-csv $file)

    new-item -ItemType Directory -Force -Path MSDN

    foreach ($row in $export) {
        $url = "https://msdn.microsoft.com/en-US/library/" + $row.ContentId
        echo $url

        $out = "MSDN\"+$row.ContentId+ "-" + $row.ContentName+".html"

        $httpContent = Invoke-WebRequest -URI $url -TimeoutSec 15
        $htmlContent = $httpContent.ParsedHtml.getElementById("content")
        if ($htmlContent -ne $null) {
            $html = $htmlContent.OuterHTML
            $html | Out-File -FilePath $out -Encoding utf8

            $images = $httpContent.ParsedHtml.getElementById("content").getElementsByTagName("img")

            foreach ($image in $images) {
                $chunks = $image.src.Split("/")
                $outName = "MSDN\" + $chunks[$chunks.Length -1]

                Invoke-WebRequest -URI $image.src -OutFile $outName -TimeoutSec 15
            }
        }
    }
}
getHTMLFromSmsExport("smsexport.csv") #Update this to point to your export CSV
```

5. Next, you can convert the html content (which is just the page content without any chrome) to Markdown, also using PowerShell

```
cd MSDN
$files = dir

foreach ($file in $files) {
    if ($file.Extension -eq ".html") {
        $out = $PSScriptRoot + "markdown\" + $file.BaseName + ".md"
        echo "pandoc $file -f html -t markdown -o $out"
        pandoc $file -f html -t markdown-raw_html-native_divs-native_spans+startnum-link_attributes-header_attributes+hard_line_breaks --atx-headers -o $out
    }
}
```

6. Now that your content is converted into markdown, it can be added to your git repository.

Text formatting guidelines

7/7/2021 • 5 minutes to read

> **TIP**
>
> You may refer external docs contributors to this public contributor guide page for similar information. Keep the information in sync between these two pages as things change.

Consistent and appropriate use of bold, italic, and code style for text elements improves readability and helps avoid misunderstandings.

Bold

Use bold for UI elements, like menu selections, dialog box names, and input field names.

Examples using bold

- **This**: In **Solution Explorer**, right-click the project node, and then select **Add** > **New Item**.
- **Not this**: In Solution Explorer, right-click the project node, and then select Add > New Item.
- **Not this**: In *Solution Explorer*, right-click the project node, and then select *Add* > *New Item*.

Italics

Use italics for:

- Introducing new terms along with a definition or explanation.
- File names, folder names, paths.
- User input. See also Placeholders later in this article.

Examples using italics

- **This**: In App Service, an app runs in an *App Service plan*. An App Service plan defines a set of compute resources for a web app to run on.
- **Not this**: In App Service, an app runs in an "App Service plan." An App Service plan defines a set of compute resources for a web app to run on.
- **This**: Replace the code in *HttpTriggerCSharp.cs* with the following code.
- **Not this**: Replace the code in `HttpTriggerCSharp.cs` with the following code.
- **This**: Enter *ContosoUniversity* for the **Name**, and then select **Add**.
- **Not this**: Enter "ContosoUniversity" for the **Name**, and then select **Add**.

Code style

Use code style for:

- Code elements, like method names, property names, and language keywords.
- SQL commands.
- NuGet package names.
- Command-line commands.*
- Database table and column names.
- Resource names (like virtual machine names) that shouldn't be localized.

- URLs that you don't want to be selectable.

Why? Older style guides specify bold for many of these text elements. However, most articles are localized, and code style tells the translator to leave that part of the text untranslated.

Code style can be *inline* (surrounded by `) or *fenced* code blocks (surrounded by ``` or ~~~) that span multiple lines. Put longer code snippets and paths in fenced code blocks.

* In command-line commands, use forward slashes in file paths if they're supported on all platforms. Use backslashes to illustrate commands that run on Windows, when only backslashes are supported. For example, forward slashes work on the .NET CLI on all platforms, so you would use `dotnet build foldername/filename.csproj` rather than `dotnet build foldername\filename.csproj`.

Examples using inline styles

- **This**: By default, the Entity Framework interprets a property that's named `Id` or `ClassnameID` as the primary key.
- **Not this**: By default, the Entity Framework interprets a property that's named *Id* or *ClassnameID* as the primary key.
- **This**: The `Microsoft.EntityFrameworkCore` package provides runtime support for EF Core.
- **Not this**: The *Microsoft.EntityFrameworkCore* package provides runtime support for EF Core.

Examples of fenced code blocks

- **This**: No commands are sent to the database by statements that just change an `IQueryable`, like the following code:

````
```csharp
var students = context.Students.Where(s => s.LastName == "Davolio")
```
````

- **Not this**: No commands are sent to the database by statements that just change an **IQueryable**, like **var students = context.Students.Where(s => s.LastName == "Davolio")**.

- **This**: For example, to run the `Get-ServiceLog.ps1` script in the `C:\Scripts` directory, enter:

````
```powershell
C:\Scripts\Get-ServiceLog.ps1
```
````

- **Not this**: For example, to run the **Get-ServiceLog.ps1** script in the **C:\Scripts** directory, type: "C:\Scripts\Get-ServiceLog.ps1."

- **All** fenced code blocks must have an approved language tag. For a list of support language tags, see How to include code in docs.

Placeholders

If you want the user to replace part of an input string with their own values, use placeholder text marked off by angle brackets. For example:

```
az group delete -n <ResourceGroupName>
```

You can note that the brackets must be removed when substituting real values. We recommend the use of italics for user input. You can format italics within angle bracketed inline code:

> `<ResourceGroupName>` is the resource group where...

If you use italics without code format, escape the starting angle bracket by putting a back-slash in front of it (`\<`).

We discourage curly braces { } for use as syntactical placeholders. Readers can confuse curly brace placeholders with the same notation used in:

- Replaceable text
- Format strings
- String interpolation
- Text templates
- Similar programming constructs

You can separate placeholder names with hyphens ("kebab case") or with underscores, or you can do it by using Pascal case. Kebab case might generate syntax errors, and underscores could conflict with underlining. All-caps can conflict with named constants in many languages, though it may also draw attention to the placeholder name.

> `<Resource-Group-Name>` or `<ResourceGroupName>`

Headings and link text

Don't apply an inline style like italics or bold to headings or hyperlink text.

Why?

People rely on standard hyperlink text to identify text elements as clickable links. Styling a link as italics, for example, can obscure the fact that the text is a link. Headings have their own styles and mixing other styles in them looks bad.

Examples of headings and link text

- **This**: The *function.json* file is generated by the NuGet package Microsoft.NET.Sdk.Functions.
- **Not this**: The *function.json* file is generated by the NuGet package *Microsoft.NET.Sdk.Functions*.
- **This**:

  ```
  ### The Microsoft.NET.Sdk.Functions package
  ```

- **Not this**:

  ```
  ### The *Microsoft.NET.Sdk.Functions* package
  ```

Keys and keyboard shortcuts

When referring to keys or key combinations, follow these conventions:

- Capitalize the first letter of key names.
- Don't apply an inline style like italics, bold, or code.
- Use "+" to join keys that the user selects at the same time.

Examples of keys and keyboard shortcuts

- **This**: Select Alt+Ctrl+S.

- **Not this**: Press **ALT+CTRL+S**.
- **Not this**: Hit `ALT+CTRL+S`.

For more information, see Describing interactions with UI.

Exceptions

Style guidelines aren't rigid rules. In contexts where they harm readability, do something different. For example, an HTML table with mostly code elements might look too busy with code styling everywhere. You might choose bold styling in that context.

If you choose an alternate text style where code is normally called for, make sure it's okay to translate the text in localized versions of the article. Code is the only style that automatically prevents translation. For scenarios where you want to prevent localization without using code style, see Non-localized strings.

GitHub workflow for MicrosoftDocs

7/7/2021 • 7 minutes to read

MicrosoftDocs uses GitHub to host our Git repositories (repos). As a contributor you'll create and update files in GitHub hosted repos, and the publishing system reads the files from the GitHub repos.

This article describes some basic GitHub concepts and the GitHub workflow specific to docs.microsoft.com. The process becomes easier to follow when you understand the location, or where the content resides. The GitHub workflow is illustrated below:

Git

Git is a distributed version control system and is a command-line tool. A set of content and associated files is known as a repository. Git helps manage and track content changes made by various authors and allows for collaboration.

GitHub

GitHub is a cloud-based Git repository hosting platform. It provides a web interface to manage repositories and collaborate.

Repository

A repository, also known as a repo, houses a content set for an organization. For example, MicrosoftDocs, is the organization and docs-help-pr is one of the repositories. Repos are the main/default/published/current version of the content. In a repository, you can store and organize the necessary files for a content set. Within a repository, there are folders and files, images, and link to videos – everything the content project needs. The main/default branch is also referred to as Upstream, especially when using commands in the command-line interface.

Markdown source files are stored in Git repositories. *Git* is a version control system for tracking changes and coordinating work in files among multiple users. A Git *repository*, also known as a *repo*, is a collection of related files. For example, there is one repo for almost all the Azure documentation and one repo for all of the ASP.NET Core documentation.

A Git repo that publishes to docs.microsoft.com contains the following kinds of files:

- Article source files. In Markdown, with YAML metadata.
- Image files for diagrams and screenshots in articles. Usually PNG format.
- Code files for use in articles and as accompanying sample projects.
- Table-of-Contents source files. Mostly in YAML, some legacy Markdown.
- Config files. For example, one config file provides URL redirection rules for deleted articles.

The repo folder structure corresponds to URL structure. For example, the App Service Environments introduction is in the following repo folder and at the following URL:

```
articles/app-service/environment/intro.md
```

```
https://docs.microsoft.com/azure/app-service/environment/intro
```

Private and public GitHub repos

Most teams use a private repo for internal contributors. Private repos aren't visible to the public. If an unauthorized person goes to a private repo URL, they'll receive a **404 Not Found** error. Most private repos are identified with a -pr. For example, the Docs Platform Help Documentation repository is docs-help-pr.

Most private repos have a corresponding public repo that people outside of Microsoft can use to propose content changes. The public repo is regularly synced with the private one.

Some product teams work with a public repo only. For example, there are no private repos for ASP.NET. And some content is hosted in a private repo only, such as legal documentation that isn't open to external contributions or feedback.

See Should I work in a private or a public repository? for information on why and how we work with private repos.

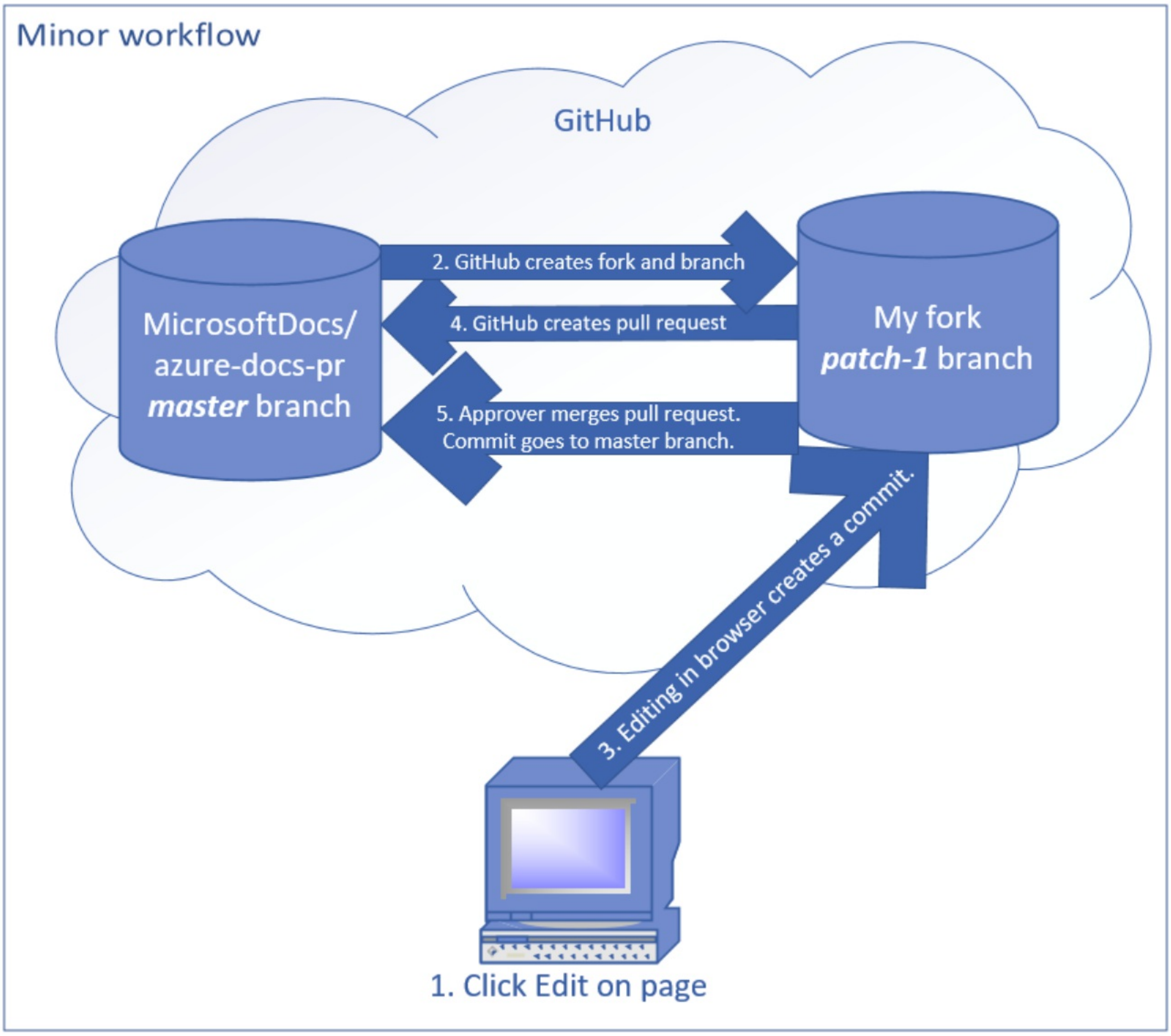
Fork

To edit an article, you must make changes in the GitHub repo it is published from. But in most cases, you don't have write permissions to that production repo. So, you make a copy of the repo in your own GitHub account and make your changes there. That copy of the production repo is called a *fork* in GitHub. A fork is a complete copy of the repository and is a snapshot in time. If changes have been made to a repo after you have forked, you need to pull the latest copy to your fork. You can fork a repository from within the GitHub UI.

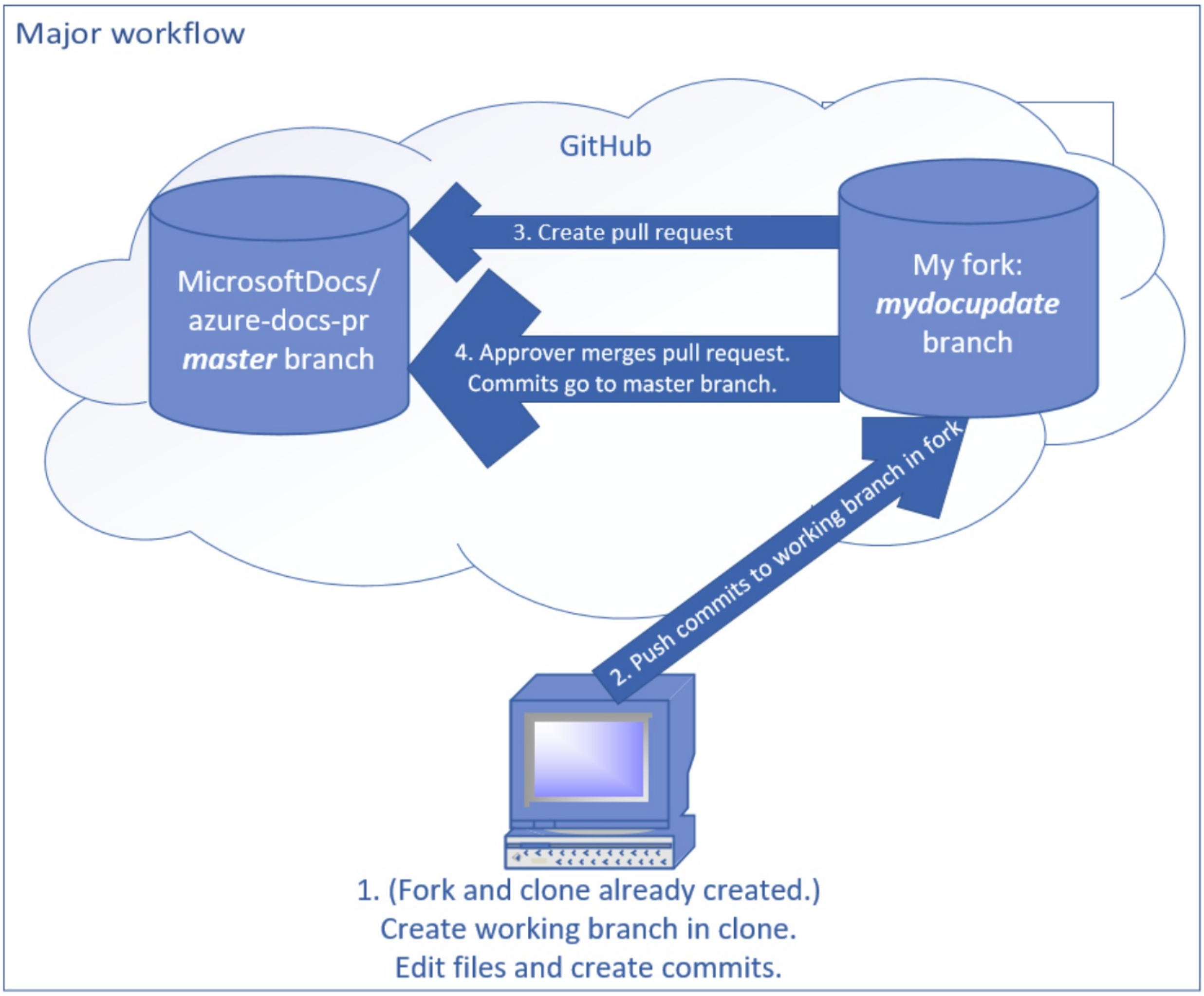
For example, the production repo (which we call the *upstream* repo) for this contributor guide it is MicrosoftDocs/docs-help-pr. When you fork this repo, you will have a full copy in your GitHub account: **YourUserName**/docs-help-pr.

Minor and major workflows

If all you want to do is fix a typo or make some similarly trivial change, you can edit a doc directly in the browser. For this *minor workflow*, you don't need a local clone repo, and a fork repo will be created for you automatically. For more about the minor workflow, see Edit an article on Docs. The following diagram illustrates the minor workflow.



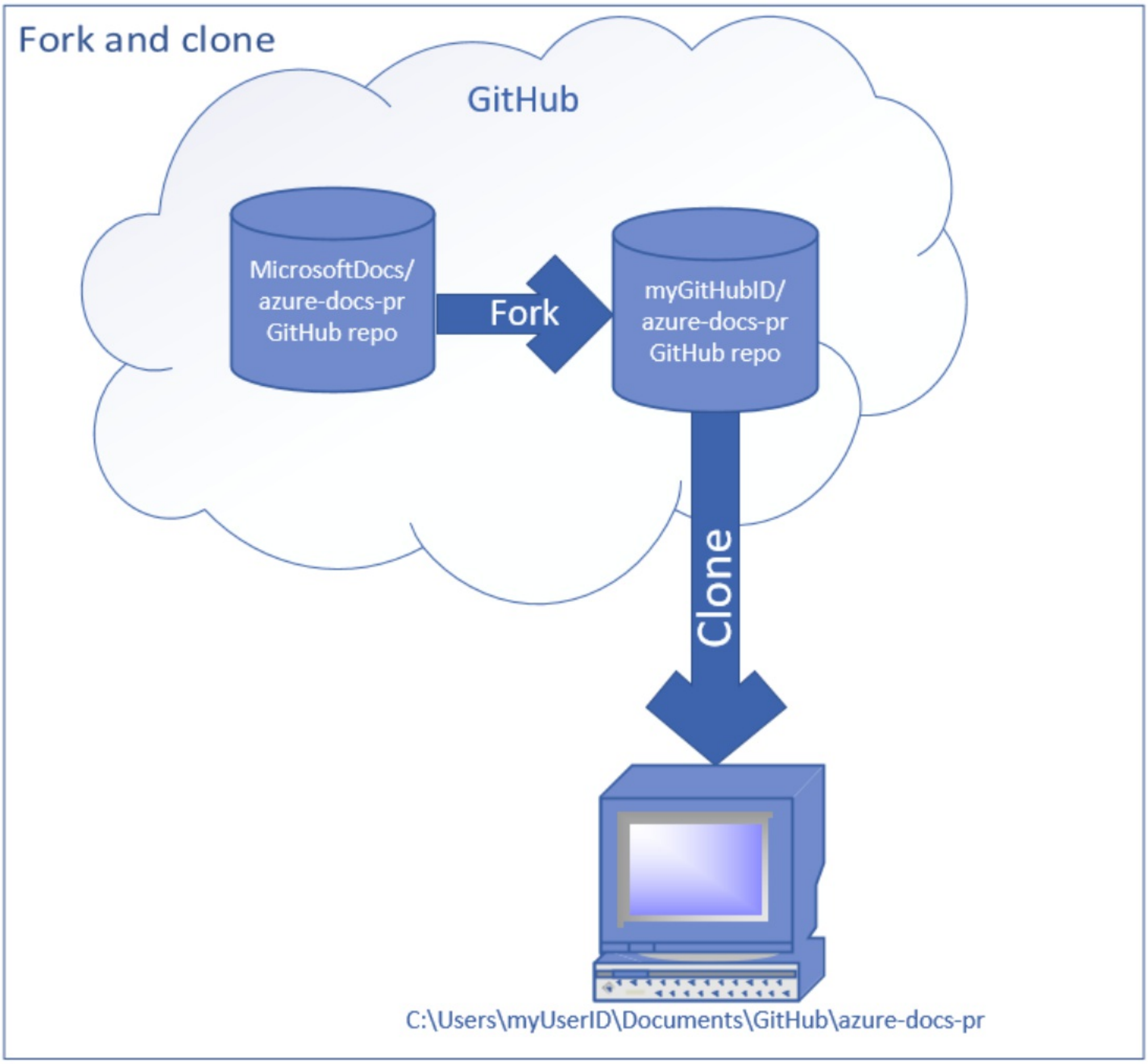
For more complex changes, it's more efficient to work with a specialized Markdown editor that has Git integration, such as Visual Studio Code. It will also be necessary at times to enter Git commands at the command prompt. For more about the *major workflow*, see Make changes to an article published on docs.microsoft.com. The following diagram illustrates the major workflow.



Clone

Although you have *write* permission to your repository fork, you can't use a full-featured text editor like Visual Studio Code directly in GitHub online. So, for major changes, you instead *clone* (download a copy of) the repo to your local computer and edit files there. Clones are copies of the fork in your local environment (for example your c: drive.). Since it's local, you can edit files using Visual Studio Code or some other Markdown editor. When ready, you *push* your changes to your fork, and then you create a pull request to merge those changes to the upstream repo.

The following diagram illustrates the relationships between a repository, fork, and clone.



To clone a fork, you'll need the clone URL from the GitHub UI and run the clone command in VS Code Terminal or Git bash. Step-by-step instructions are provided in the Prepare your environment article.

Remote

Remotes are aliases that point to repos in GitHub. You will have on named *origin* that points to your fork, and one named *upstream* that points to the production repo. Step-by-step instructions are provided in the Prepare your environment article.

Commit

When you create a commit, Git records the changes that you made locally in your working branch. Before you run the commit command, you must clearly tell Git, which file(s) to commit. So, you'll first use the Git add command with the file path followed by Git commit. You can use the VS Code Terminal or Git bash to commit the article(s) you changed. Step-by-step instructions are provided in Make changes to an article.

Push

After committing your changes to your local branch, you push the changes to the remote that points to your fork. Step-by-step instructions are provided in Make changes to an article.

Pull request

After you push the content changes to your fork, go to **GitHub.com** to create a pull request. A pull request basically asks the production repo to pull in the changes from the working branch in your fork. A pull request compares and displays differences between your fork and the production repo so you and others can easily review the changes.

A pull request also notifies the content author and reviewers that the content is ready to be reviewed. All reviews, comments, and feedback can be provided directly within GitHub.com. After the feedback and reviews are incorporated, the changes are *merged* to the main repository. Step-by-step instructions to create a pull request are provided in Creating a pull request.

Merge conflicts

With many people working on the same files, it's inevitable that at times different people will make different changes to the same part of a file. When that happens, a pull request can't be merged until the conflict is resolved. There are two ways to resolve the conflict:

- Create a new commit that resolves the conflict.
- Rebase your existing commits.

Searching the contributor guide

The internal-only contributor guide that you're reading right now isn't indexed by Google or Bing. You have two options for finding something specific in it: the TOC filter and site search.

Table of contents (TOC) filter

Directly above the TOC on the left side of the page is a text box with a "Filter by title" watermark. This is not a full-text search. Enter a word here and the TOC is limited to nodes that contain that word in the title, plus nodes where that word was deliberately entered into TOC metadata. Enter "merge", for example, and you see the *Resolve merge conflicts* article but not articles with other titles that happen to mention merging.

Site search

For a full-text search, start on any contributor guide page, and select the **Search** link with magnifying glass icon at the top right of the page. Enter a search term such as "merge" here, and you get a list of more than 50 pages that use that word.

If you can't find what you're looking for, you can expand the search scope. When you select **Search**, notice the **Contributor X** at the left of the text box. That means results will be limited to the contributor guide. To include other related doc sets, select the **X** to delete **Contributor X**. Search results from other doc sets such as OPS and localization will then be included. For "merge," the number of results now exceeds 350.

Migrating GitHub project documentation to Docs

7/7/2021 • 2 minutes to read

Per our secondary site policy, technical documentation for consumers of a Microsoft product or service needs to be on docs.microsoft.com (Docs). It shouldn't be on secondary sites. Developer docs in GitHub repos that host a product or service's tools or programming libraries belongs on Docs.

This guidance only applies to GitHub projects supporting products and services whose documentation is already onboarded to Docs. If you need to onboard an entirely new product or service to Docs, use this template to open a user story to get started with the onboarding process. Visit the Repo admin guide for more info.

The guidance in this article doesn't apply to Azure sample GitHub repos. Create and manage that content per our Create and publish sample code guidance.

What can live on GitHub? What must move to Docs?

Content written for contributors to the GitHub project should remain on GitHub. Content written for consumers of the project should live on Docs. Examples include install and usage documentation, library reference, and any getting started documentation for new users to the project.

Bring your GitHub content into an existing content on Docs

If the product or service that your GitHub project supports is already on Docs, contact the content developers and PMs servicing that content. Find out the best way to bring your content over from GitHub. They'll need to approve the new content on Docs and the changes to the table of contents that your material requires via a pull request. Once the PR is approved, your changes will be merged into the existing content repo.

If your GitHub project is a library or an API that has reference docs, review Reference Onboarding.

Is your GitHub project a standalone utility, sample application, or integration with an existing third party or open-source tool? If so, work with the content team for your product or service to figure out how to bring usage of the tool on to Docs.

How to rebase and squash

7/7/2021 • 7 minutes to read

Rebase means replace a commit with an equivalent one based on a different, usually newer, commit. Rebasing a working branch on the current master branch lets you deal with merge conflicts in advance. Doing so prevents the conflicts from showing up in a pull request.

Squash means replace multiple commits with a single commit. Squashing simplifies the commit history in the repository. Squashing can also be useful if you accidentally committed a change containing personal data to your local repository. You can make a second commit removing it, then squash the two. The personal data doesn't show up in history.

This article shows how to rebase and squash in one action. Two sections at the end show how to change the procedure so you can rebase without squashing or squash without rebasing.

Prerequisites

The example git commands in this article assume that you:

- Are working in a repository clone that has `origin` pointing to your fork and `upstream` pointing to the production repository.
- Have checked out a working branch based on master and named *myworkingbranch*.
- Made the working branch track its corresponding remote branch in `origin` by using the `-u` option with `git push`.
- Have added multiple commits to this working branch.
- Have pushed the branch to your fork in GitHub and created a pull request.

Set up an editor

You'll use the *interactive* option to squash. Git will create a text file with a list of commits that are in *myworkingbranch* but not master. Git opens this file in a text editor so you can specify which commits you want to keep, drop, or squash. When you save your changes to this text file and close the editor, Git proceeds to rebase and squash as directed.

Before doing squash and rebase for the first time, make sure Git will spawn an editor that you want to work with. Notepad is an option, but we recommend Visual Studio Code because it has a helpful UI for handling conflicts. The following returns a list of the global settings in your local Git config:

```
git config --global -l
```

Check the value of `core.editor`: if it's Visual Studio Code as shown in the following example, leave it as-is.

```
C:\Source\dotnet\docs [master ≡]> git config --global -l
user.name=Nancy Davolio
user.email=nancy.davolio@microsoft.com
core.editor=code --wait
```

If `core.editor` is missing or is something different, set it to Visual Studio Code.

```
git config --global core.editor "code --wait"
```

Update your master branch

It's best to rebase on the latest version of the repository's master branch, so you need to bring that version into your local clone's master branch:

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

Begin rebase and squash

```
git rebase master -i
```

This command means rebase the current branch (*myworkingbranch*, for example) on the master branch. The `-i` flag means do it interactively. If all you want to do is rebase, you can omit the `-i`. In that case, Git will rebase all commits on master without squashing them.

The command creates a text file and opens it in Visual Studio Code. The file looks like the following example:

```
pick 0e66f24b initial
pick d7ff40a5 typo
pick e9f708e2 fix link

# Rebase a3aad717..e9f708e2 onto a3aad717 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .       create a merge commit using the original merge commit's
# .       message (or the oneline, if no original merge commit was
# .       specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
#    However, if you remove everything, the rebase will be aborted.
#
#
# Note that empty commits are commented out
```

To squash all of the commits, change the second through the last one from "pick" to "s", as in this example:

```
pick 0e66f24b initial
s d7ff40a5 typo
s e9f708e2 fix link
```

Save your changes and close the file.

Rebasing and squashing starts, and Git reports its progress as it applies each commit. If you're rebasing after merging master into a release branch, rebase finds no changes when it reaches the merge commit. In that case, you have to do a `rebase --skip` to skip the merge commit and continue with later changes.

When Git finishes processing all of the commits, it generates another text file and opens it. In this file, you specify the comments that will go with the new commit. Here's what this file looks like:

```
# This is a combination of 3 commits.
# This is the 1st commit message:

initial

# This is the commit message #2:

typo

# This is the commit message #3:

fix link

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Dec 5 11:52:28 2018 -0800
#
# interactive rebase in progress; onto a3aad717
# Last commands done (3 commands done):
#    squash d7ff40a5 typo
#    squash e9f708e2 fix link
# No commands remaining.
# You are currently rebasing branch 'myworkingbranch' on 'a3aad717'.
#
# Changes to be committed:
#    new file:   help-content/contribute/rebase-and-squash.md
#
```

Make any changes you want to the commit comments, and then save and close the file.

Git applies the comments to the new commit, and the rebase-with-squash process is done.

Force push

Now you're ready to push the new rebased and squashed commit to your fork, but `git push` doesn't work. The remote repository has commits based on where the master branch was when you started. The local clone has a new one based on where the master branch is now. The `git push` command can't just add commits to the end of your fork's commits so it fails. To get past that you add the `--force` option.

```
git push --force
```

Before doing a force push, make sure you don't need to access the original commits again. They'll be permanently gone when the push is finished.

Conflicts

During the rebase action, a conflict similar to a merge conflict may happen if the current master branch includes a change in a file that one of your commits changes.

When a conflict occurs, Git opens the file in Visual Studio Code. Visual Studio Code gives you several options for handling the conflict. Here's an example of the UI:

```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
It exists solely to demonstrate concepts in Joshua's blog posts:
=======
It exists solely to demonstrate concepts in git blog posts:
>>>>>>> 348a372... commit 1 (Incoming Change)
```

- **Accept Current Change** - *Current Change* is the change in master that conflicts with the commit that you're rebasing.
- **Accept Incoming Change** - *Incoming Change* is the one in the commit that you're rebasing.
- **Accept Both Changes**
- **Compare Changes** - Opens a read-only compare window.

For each conflict in the file:

- Select the best **Accept...** choice.
- If you need to preserve elements from both changes, edit the file.

After you've resolved the last conflict in the file:

- Save and close the file.
- Stage your changes and continue the rebasing process:

  ```
  git add .
  git rebase --continue
  ```

- Git may generate a COMMIT_EDITMSG file and open it in Visual Studio Code. Enter a commit message appropriate for the way you resolved the conflict, and save and close the file.

  Git applies your comment to the commit and continues rebasing and squashing until it hits the next conflict or is finished.

Rebase without squashing

You may not want to squash commits. Sometimes there are commits from other contributors alongside your commits in the same branch. If those contributors don't want to lose the record of their contribution, don't squash the commits.

To rebase without squashing, follow all of the same procedures as rebase-with-squash except omit the `-i` (interactive) flag on the rebase command. Git won't present you with a list of commits to edit. Rebasing without squashing will replay multiple commits. You may need to resolve the same merge conflict area multiple times, once for each conflict.

Squash without rebasing

One way you can squash without rebasing is to rebase-with-squash on the parent of the first commit in the branch. In effect, you're not rebasing because the rebase is on the same commit the branch is based on.

To get the SHA1 of the first commit in a branch named *squashnorebase*, run the following command:

```
git log master..squashnorebase --oneline
```

The last line in the output will contain the SHA1 of the first commit. For example:

```
b836b8f7 (HEAD -> squashnorebase) third commit
90dee821 second commit
ec8c84d5 first commit
```

Then follow the same procedures as rebase-with-squash with one difference. Be careful to rebase on that first commit's parent by using its SHA1 with a tilde (`~`) appended:

```
git rebase -i ec8c84d5~
```

Next steps

For information about how to resolve merge conflicts without rebasing, see Resolve merge conflicts in Git and GitHub.

Best practices for writing/publishing content to Docs.Microsoft.Com

6/24/2021 • 3 minutes to read

This guide provides tips and best practices for Docs.Microsoft.Com contributors to publish quality content, on time.

Hit your publish date

These tips help you publish your content on time while avoiding last-minute frenzy. Avoiding last-minute frenzy is especially important when your content is required to align with a product/service release on a specific date.

1. **Author your article in Markdown** (*.md) from day one—do not use Microsoft Word to write your article.

   Why: You avoid the time required to convert from Word to Markdown, as well as the additional vector for introducing errors when you perform the conversion.

2. Use the major or long-running changes workflow, and **create a pull request (PR) for your article on day one.** Regularly push your changes to GitHub so that you can review it on staging as you go.

   Why: You can see the rendering of your article on docs.microsoft.com and receive error notifications from the automated builds. Staging your articles helps you avoid last-minute fixes required by the PR reviewers, and get things like broken links fixed well before sign-off.

3. Use GitHub's built-in pull request review feature to perform your article review with your team members and the content team.

   Why #1: It's a slick, built-in feature of GitHub, and allows you to track all requested changes (on even a line-by-line basis) from multiple reviewers.

   Why #2: Your article is already in Markdown, on GitHub, and staged on docs.microsoft.com, so you—and your reviewers—can review the article as it will appear when it's published live.

   Why #3: You aren't authoring in Word anymore, and you need a replacement for Word's review feature.

4. C+AI/Content & Learning contributors should **start the PR review process early**, and `#sign-off` on your article at least **two days** before your target publish date.

   Why: This gives you plenty of time to integrate any changes required by the PR Reviewers team (this is when they kick back your article and issue a `#hold-off` comment). Rarely are articles approved on the first try, even for those on the content team. Give yourself time to fix and resubmit the article for review via `#sign-off` well before your target publishing date. A resubmitted article does not get placed at the top of the review queue, so you have to wait in line with everyone else.

Review content using GitHub pull requests

Use these recommendations for reviewing content in GitHub pull requests.

How do I invite others to review my content?

Once you've written, or updated, your article and created the GitHub pull request you can invite others to review your work and provide feedback.

To invite reviewers via your pull request, provide the following information in the PR comment section:

- **Review request**
- **Reviewers:** @githubalias1 @githubalias2 (Use @mention as the best practice to provide a record of invitees.)
- **Review deadline:** 2017-08-27
- Any other information that will help the reviewers understand your request.

For example:

> **mmacy** commented 2 days ago • edited
>
> **Review request**
> **Reviewers: @Thraka @sethmanheim @cwatson-cat @davidmu1 @robinsh @neilpeterson @iainfoulds @BryanLa @tamram**
> **Review deadline**: Friday, 2017-08-11
>
> Per email "doc reviewers wanted: Using GitHub for content reviews," would love some eyes on this article: Best practices for writing/publishing

Create a PR review request template with GitHub saved replies

When you regularly solicit content reviews using GitHub pull requests, you can save time by adding the review request template to your saved replies. Add the following code block, select it from the drop down, and add the specific details.

```
### Review requested
---
- **Reviewers:** @githubalias1 @githubalias2
- **Review deadline:** YYYY-MM-DD
- **Notes**: some notes here
   more notes on a second line
---
```

How do I review content and provide feedback?

If you're asked to participate in a content review via GitHub, you review the article and provide your feedback in the pull request.

1. Start a review and add comments at the line-level
2. Submit the review
   - Leave review comments in the **Review summary** (this notifies the author)
   - Select appropriate review type (Comment, Approve, Request changes)
3. (Optional) Add *thumbs-up* or *thumbs-down* (reactions) to other reviewers' comments to indicate that you agree or disagree.

> **NOTE**
>
> If you have no changes, indicate that in the PR comment. That way, the author will know that you reviewed the content.

How do I review and incorporate suggestions from a review?

Once reviewers have provided their feedback (or the review deadline has passed) you review and integrate the appropriate changes and reply to comments.

1. Reply to review comments with:

- Fixed
- Won't fix (add explanation)
- Or, ask for clarification

2. Edit and push the changes to GitHub
3. @mention the reviewer(s) to notify them that changes are ready for review
4. Iterate steps 1-3 as necessary
5. `#sign-off` on PR

> **TIP**
>
> Keep your Outlook rules in mind. For example, you might not see a notification if you use rules to route email to **Deleted Items**.

Additional resources

- Pull request best practices

GitHub for content reviews

6/24/2021 • 5 minutes to read

GitHub features pull request reviews that allow you and your team to comment on the content in a pull request (PR), approve the changes, or request additional changes before the PR is merged. Similar to the review feature in Microsoft Word, you can comment on specific sections of an article, and collaborators can comment on your comments.

Why use GitHub reviews

Now that you author content in Markdown instead of Word, you need a way to collaboratively review and comment on articles with your team. By using GitHub for reviews and keeping your content development cycle completely in Markdown, you can:

- Significantly lessen publish time
- Avoid Word-to-Markdown conversion errors
- Preserve complete review history along with published articles
- Support a consistent review model in Content & Learning

You can publish faster, eliminate errors introduced by format conversion, and your articles remain the "single source of truth."

GitHub review features

GitHub provides several review features for the content in a pull request. With GitHub reviews, you can:

- Approve the changes in a PR, or request additional changes
- Leave single (atomic) comments
- Leave comments as part of a larger review
- Reply to comments
- React to others' comments (e.g. thumbs-up)

In the following sections, we describe the different ways reviewers can leave comments on your content.

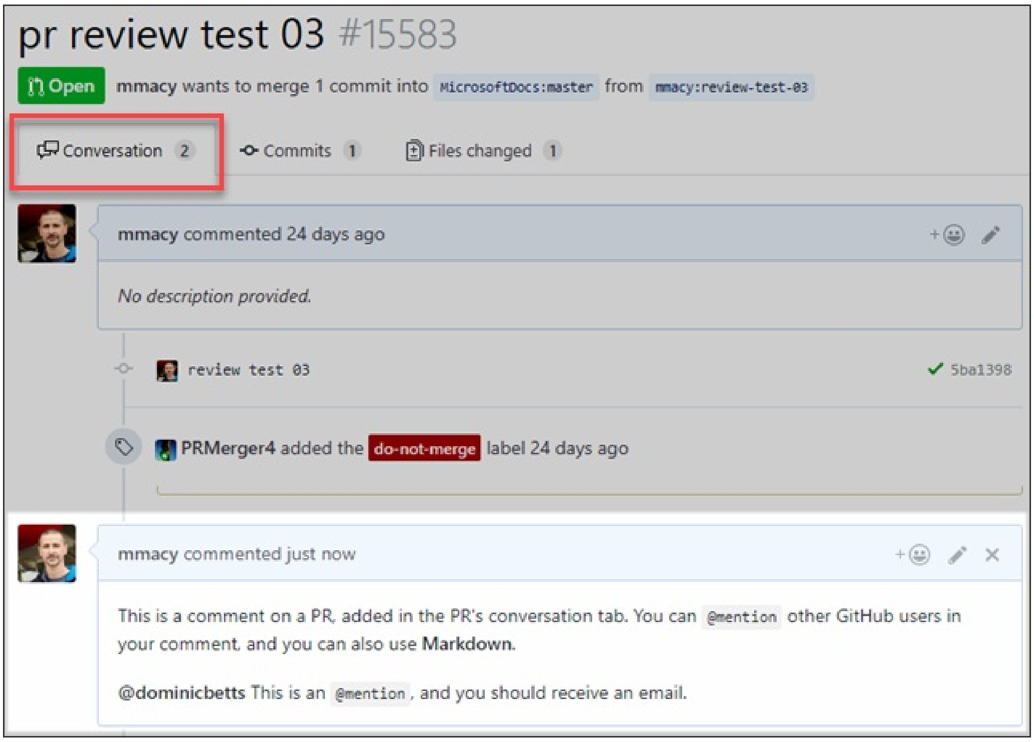
GitHub comments

Whether feedback is provided as part of a review or a simple one-off, reviewers use the comment feature of GitHub to do so. You can add both pull request comments and line comments. Line comments can further be added as part of a larger review.

You can *@mention* other contributors in your comments, as well as use Markdown. Mentioned GitHub users will receive an email containing your comment (use GitHub aliases when you *@mention*), and your comment's Markdown is rendered once you submit the comment.

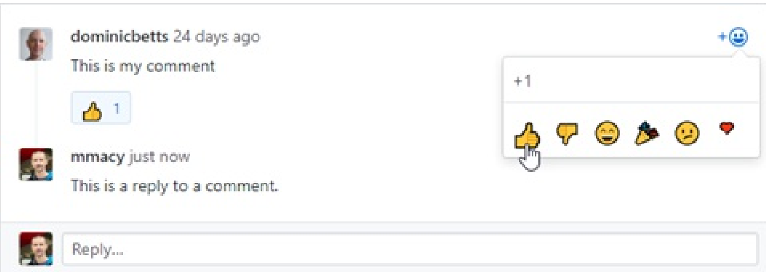
Pull request comments

To comment on a PR as a whole, add a comment to the PR's *Conversation* tab to add a **pull request comment**.



A comment in the Conversation *tab of a PR*

You can reply and react to any comment, even your own. Here's a comment with a reply and one "thumbs-up" reaction. Reaction selection is also shown.

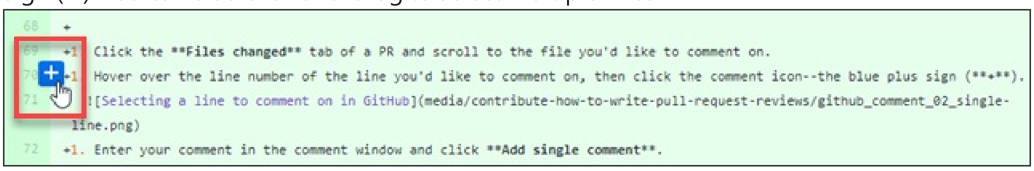


Reacting to a comment

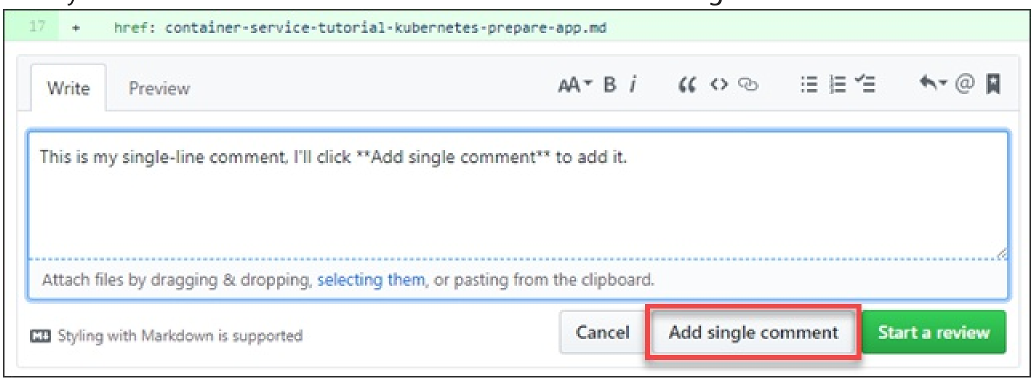
Line comments

You can add a **line comments** to call out on a specific section of a file. To add a line comment:

1. Click the **Files changed** tab of a PR and scroll to the file you'd like to comment on.
2. Hover over the line number of the line you'd like to comment on, then click the comment icon--the blue plus sign (**+**). You can also click-and-drag to select multiple lines.



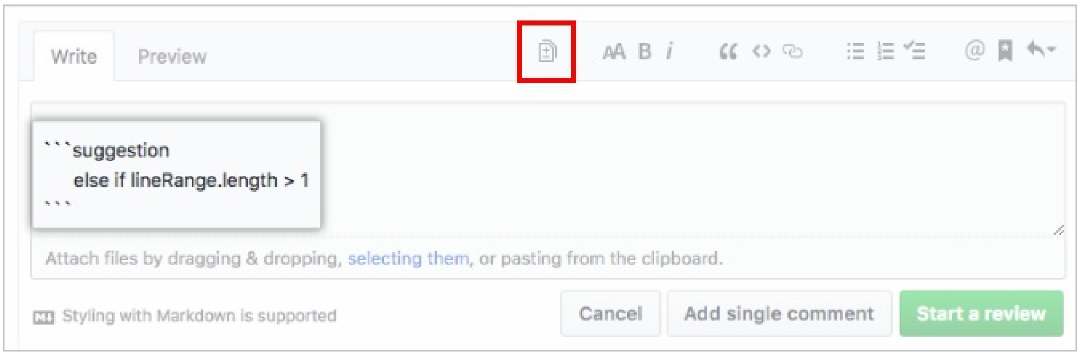
3. Enter your comment in the comment window and click **Add single comment**.



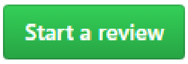
Suggest changes

In addition to commenting on a line, you can suggest specific changes. After you've selected the lines you'd like to comment on:

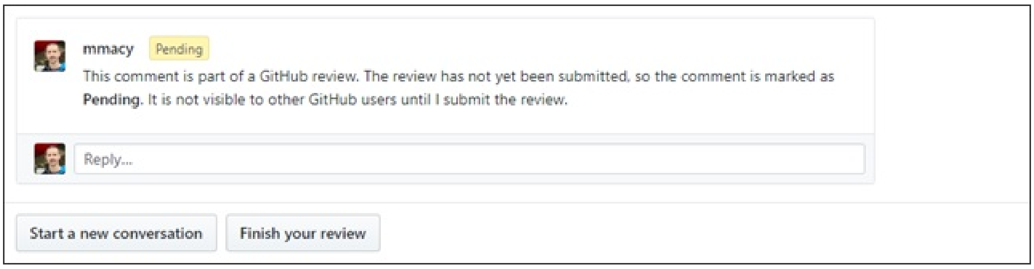
1. Click the documents-with-plus-sign icon to the left of the font picker, then edit the text within the suggestion block.



GitHub reviews



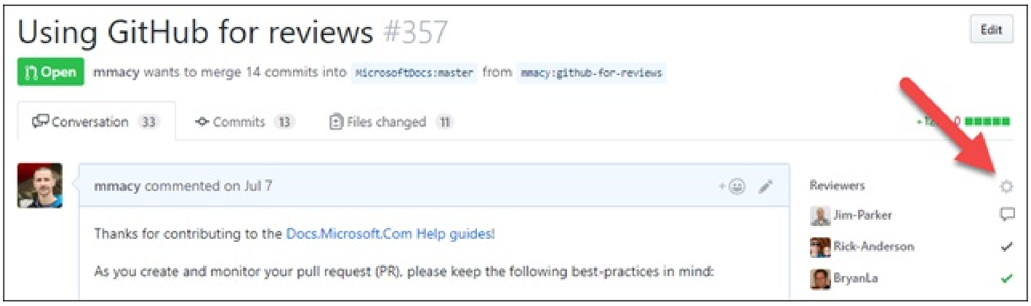
Reviews allow you to add several comments and submit them as a group. You can add as many comments as you like, each of which remains *Pending* until you submit the review. Pending comments are invisible to other GitHub users, and only become visible when you submit the review.



Invite reviewers

There are several ways you can invite others to review the content in your PR. Two of them, @mentions and email, are always available. A third, GitHub's *Request a review* feature, is available only if you have write permission to the repo.

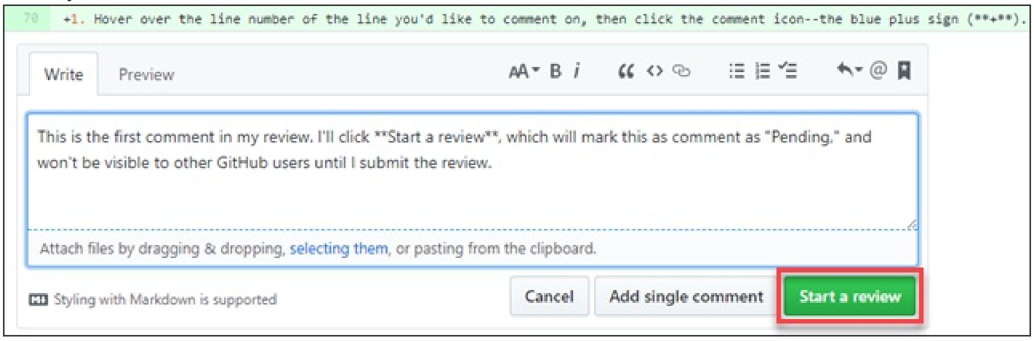
- **@mention** (recommended): Add a PR comment and @mention your reviewers with their GitHub aliases. You can find Microsoft-linked GitHub aliases in the GitHub contributor alias search. An @mention in a comment acts as a permanent record of reviewers directly in the PR, and your mentioned reviewers automatically receive an email notice from GitHub when you submit the comment. TIP: Use Markdown in your PR comment to provide friendly links to the staged article(s) in your review request (`[article title](https://review.docs.microsoft.com/.../article.md)`).
- **Email**: You might also wish to send an email directly to your reviewers. Provide a link to the PR, as well as links to the staged articles you'd like them to review.
- **Request a review** GitHub feature: This feature is available only if you have *write permissions* to the repo. If you have write permissions, a small gear icon appears under the **Reviewers** heading on the right-hand side of the PR. Click the gear and enter a GitHub alias to invite that person as a reviewer; they'll receive an email containing the review request.



Start a review

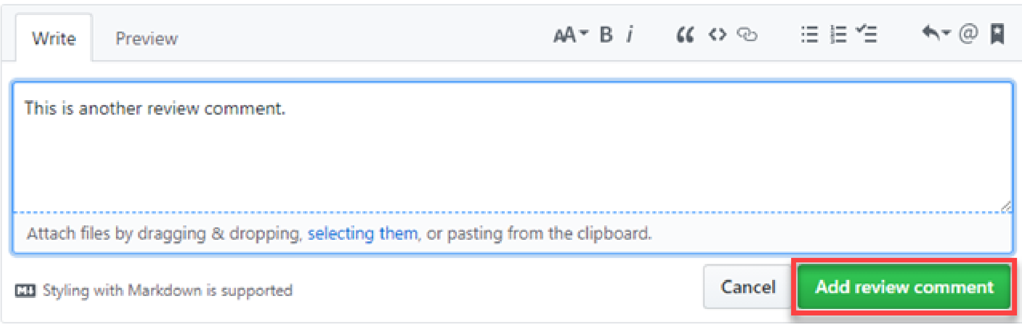
As a reviewer, start your content review by performing these steps:

1. Click the **Files changed** tab of a PR and scroll to the first file you'd like to comment on.
2. Hover over the line number of the line you'd like to comment on, then click the comment icon--the blue plus sign (+).
3. Enter your comment in the comment window and click **Start a review**.



Add review comments

To add additional review comments, repeat the steps to add a line comment, then click **Add review comment**.

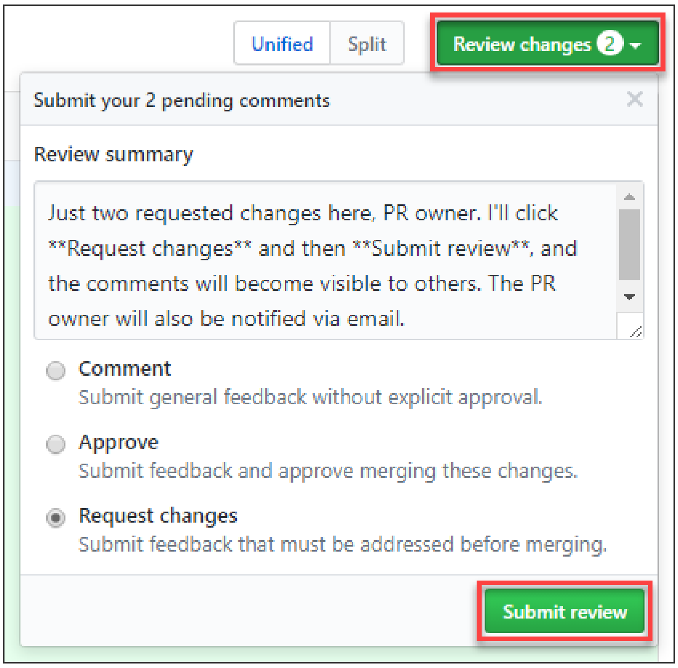


Add as many comments as you like to your review. Each comment is private and can't be seen by other GitHub users until you submit the review.

Submit a review

Once you've added all your comments to the review, submit your review.

1. Click the **Review changes** button in the upper-right of the *Files changed* tab to open the review submittal dialog.
2. Enter a descriptive **Review summary**.
3. Select the radio button for your desired review type, then click the **Submit review** button.



The pull requestor will receive notification of the review, and all of its comments can then be addressed as described above.

> **NOTE**
>
> Unless your GitHub repo enforces branch protections and required reviews, "Request changes" does not actually prevent a merge. Most of our repos do **not** enforce branch protections and required reviews. As such, "Request changes" merely notifies the PR owner that you're requesting changes they should integrate before signing off on their PR.

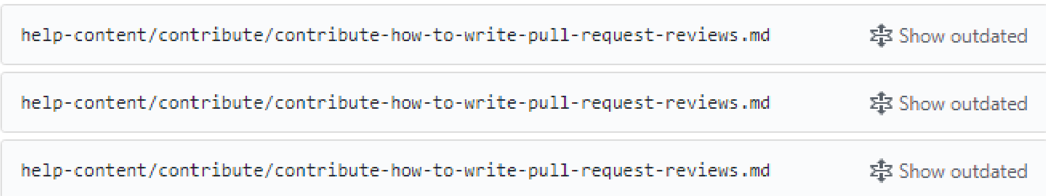
Respond to a review

When another GitHub user submits a review on your PR, you'll receive an email containing all the comments in the review. You are then free to navigate to the PR in GitHub and respond to those comments however you see fit. You can:

- Integrate changes suggested in the comments
- Reply and/or react to comments

If you agree with a comment, consider adding a "thumbs up" emoji rather than writing text, which triggers an unnecessary notification to the reviewer.

When you make changes to a line a reviewer commented on, and subsequently push your changes, the comment is shown as collapsed in the *Conversation* tab of your PR.



More info

Here are a few links to GitHub's documentation on commenting and reviews.

- Reviewing changes in pull requests
- Commenting on a pull request

- Reviewing proposed changes in a pull request

Git and GitHub glossary

6/1/2021 • 6 minutes to read

This article is a reference for common terminology and concepts that writers and contributors who work with GitHub will come across.

Acrolinx

A tool that checks spelling, grammar, style, tone, clarity, and key terminology use. It runs automatically for some repositories when you create a pull request. Acrolinx has a Visual Studio Code extension that enables you to run its checks on an article before you create a pull request.

Branch

A parallel version of a repository (or a fork or clone) that lives within the repository itself. A branch contains the actual files and folders that make up a project's content set. Branches are insulated from each other so you can edit a file in one branch without affecting that file in another branch. Branches are used to separate streams of work, and contributions are always made to a specific branch.

All repos contain a default branch (sometimes called *master* or *main*) and one or more branches destined to be merged back into the default branch. The default branch serves as the current version and single source of truth for the project. It's the parent from which all other branches in the repo are created.

Clone

A copy of a repository, including branches, that lives on your local hard drive as a folder. Often referred to as your "local repo" or "local branch." When you're working, the files that you're actually editing are in your cloned repository.

Commit

Confirmed changes made to your local working branch. Committing to GitHub is similar to saving a file to your hard drive. However, each time you commit changes, GitHub assigns a unique ID to the commit that tracks what changes were made, when, and by whom.

Default branch

The default branch that's created with every repository (and clone and fork). The default branch may be named *master* or *main*.

> **NOTE**
>
> You should **never** work in the default branch. Always create a *working branch* and then *merge* your edits into the default branch.

The default branch serves as the current version of content for the project. The content in the default branch aligns loosely with the organization of the articles in the corresponding Docs pages. Subfolders are used to separate articles, media content, and "include" files, which allow writers to reuse content.

Within the root folder, you'll find general articles about the service or product. There's also usually a series of subfolders that correspond to features, services, or common scenarios. Each writing group decides how to organize its folders.

Fetch

A Git command that copies changed files from a repository to your current branch *but doesn't merge them*. You'll usually see this command in the GitHub Desktop toolbar as **Fetch origin**.

Fork

A copy of a repository, including branches, that lives in your personal cloud space on GitHub. If you don't have write permissions to a repository, you can "fork" that repository and have full write permissions to the fork. Then, you can create a pull request to have an admin sync your changes back to the repository.

Git

An open-source, centralized version-control system that facilitates project collaboration through distributed version control of files that live in repositories. Git makes it possible to integrate streams of work done by multiple contributors over time for a given repository.

Unlike other centralized version-control systems (like SharePoint or Visual SourceSafe), Git has a unique contribution workflow and terminology to support its distributed model. For example, there are no files, which are usually associated with check-in/check-out operations. Instead, Git is concerned about changes at a finer level of granularity, comparing files byte by byte. Git also uses a tiered structure to store and manage content for a project. A repository is the highest unit of storage and contains one or more branches, which is where the actual files and folders for a project live.

Contributors interact with Git to update and manipulate repos locally using tools like GitHub Desktop and on GitHub, which integrates Git to reconcile contributions flowing back into the main repo.

GitHub

A web-based hosting service for Git repositories, such as those used to store docs.microsoft.com content. To edit and submit files, you must have a GitHub account.

All workflows begin and end at the GitHub level, where the main repo for any given Docs project is stored. Contributors create copies for their own use. These copies are distributed across multiple computers and eventually reconciled back into the project's default GitHub repo.

GitHub Desktop

A cross-platform GUI application for GitHub.

Label

A flag that's attached to a pull request in GitHub that indicates the status of the request.

Markdown

A user-friendly markup language that supports basic text formatting (like bold text, lists, headings). Markdown files use the file extension *.md*.

Merge

A command that combines the contents of two versions of a file.

Merge conflicts

Conflicts that can occur when two or more people have modified a file and try to merge it with the default branch. Some conflicts can be resolved automatically by GitHub, while others need to be manually merged. Keeping your branches synced regularly can help minimize merge conflicts.

Origin

When you clone a repository, the repository that you cloned ***from*** is called the origin. The origin is always relative to the branch you're currently in. For example, if you are in the default branch of your clone, the default branch on the primary repository is the origin. If you're in your working branch of your clone, the working branch of the primary repository is the origin.

Pull

A command that copies changed files **from** a repository **to** your current branch and merges them.

Pull request

The process of submitting a set of changes (also known as commits) to be merged from one branch into another branch (usually the default branch). The commits are stored in a contributor's working branch, allowing GitHub to first model the effect of merging them into the default branch. A pull request serves as a mechanism to provide feedback from the build and validation process to the contributor. The pull-request reviewer can resolve potential issues or questions before the changes are merged into the default branch. After the review is complete, anyone with *Write* permissions can merge the pull request.

Push

A command that copies changed files **from** your current branch **to** the target repository and merges them.

Repository

A storage area (think of it as a folder) on GitHub where all the files for a project are stored. Colloquially referred to as "repos." For example, the *OfficeDocs-ProjectServer-pr* repo stores the content and media files for Project Server and Project Online.

Repos can be *public* or *private*. Public repos are accessible to everyone using GitHub, while private repos are accessible only to Microsoft full-time employees and vendors. Most projects that publish to docs.microsoft.com use a separate private repo for internal contributors and a replica public repo for external contributors. Usually, private repos use the same name as the corresponding public repo but have "-pr" affixed to the end (as in *OfficeDocs-ProjectServer-pr*). As an internal contributor, you should be working with the private repo for your content area.

Stash

Command to temporarily shelve (or stash) uncommitted changes if you need to switch what you are working on. Stashed changes can be reapplied later.

Upstream

A name for the primary repository that contains the final changes for project files. Like origin, it's relative to your current branch. For example, say you have an environment with a primary repository, a fork, and a clone. Relative to your clone repository, the primary is the upstream and the fork is the origin. If you cloned directly from the primary repository, the primary is both the upstream and the origin.

Visual Studio Code

The editor we generally use to create and modify Markdown files. Colloquially known as "VS Code."

Working branch

A branch a user creates to isolate changes from the default branch.

How to recover a lost GitHub account

6/18/2021 • 2 minutes to read

This article discusses options for recovering your GitHub account if you can't sign in or are locked out.

Use your recovery codes

One requirement for contributing to private/internal Docs repos at Microsoft is to enable two-factor authentication (2FA) on your GitHub account. As part of this process, you should have downloaded and saved your 2FA recovery codes. GitHub creates 16 recovery codes for you. You can use these codes to authenticate to GitHub if you lose access to your 2FA credentials (for example, if you lose your phone where you configured 2FA).

Once you use a recovery code to regain access to your account, it cannot be reused. If you've used all 16 recovery codes, you can generate another list of codes. Generating a new set of recovery codes will invalidate any codes you previously generated.

Use another recovery option

GitHub provides several ways to recover your account if you lose access to 2FA. These include authenticating with:

- A fallback number
- A security key
- A verified device, SSH token, or personal access token
- An account-recovery token

Many of these options must be configured *before* you lose access to your account.

Create a new account

If none of the above methods succeed, follow these steps to create a new account.

Caution

If you disassociate your Microsoft corpnet and GitHub accounts, you may lose permissions and group memberships. If you have private repos created as forks, those will be deleted. And there may be other unexpected side affects related to your GitHub data. *Simply relinking the accounts does not undo these changes.* You will have to onboard from scratch. Use this procedure only as a last resort.

1. Go to https://repos.opensource.microsoft.com/link.
2. Select **Remove your link**, and then confirm. This action unlinks your GitHub account from your Microsoft corporate account.
3. Follow the instructions at Sign up for GitHub to create a new account.
4. Return to https://repos.opensource.microsoft.com/link and link your GitHub account with your Microsoft account. (If your accounts are already linked, the page will show "You're linked!")

Triaging new GitHub issues

6/24/2021 • 13 minutes to read

In this article you will:

- Learn about the triage process you need to use when you and your team work with GitHub issues.
- Understand the workflow and issue classification system we are using
- Learn how to work with issues in our open-source projects

Whether you're working in a large docset or a small one, you need to have a plan for triaging incoming issues.

Planning for triage means:

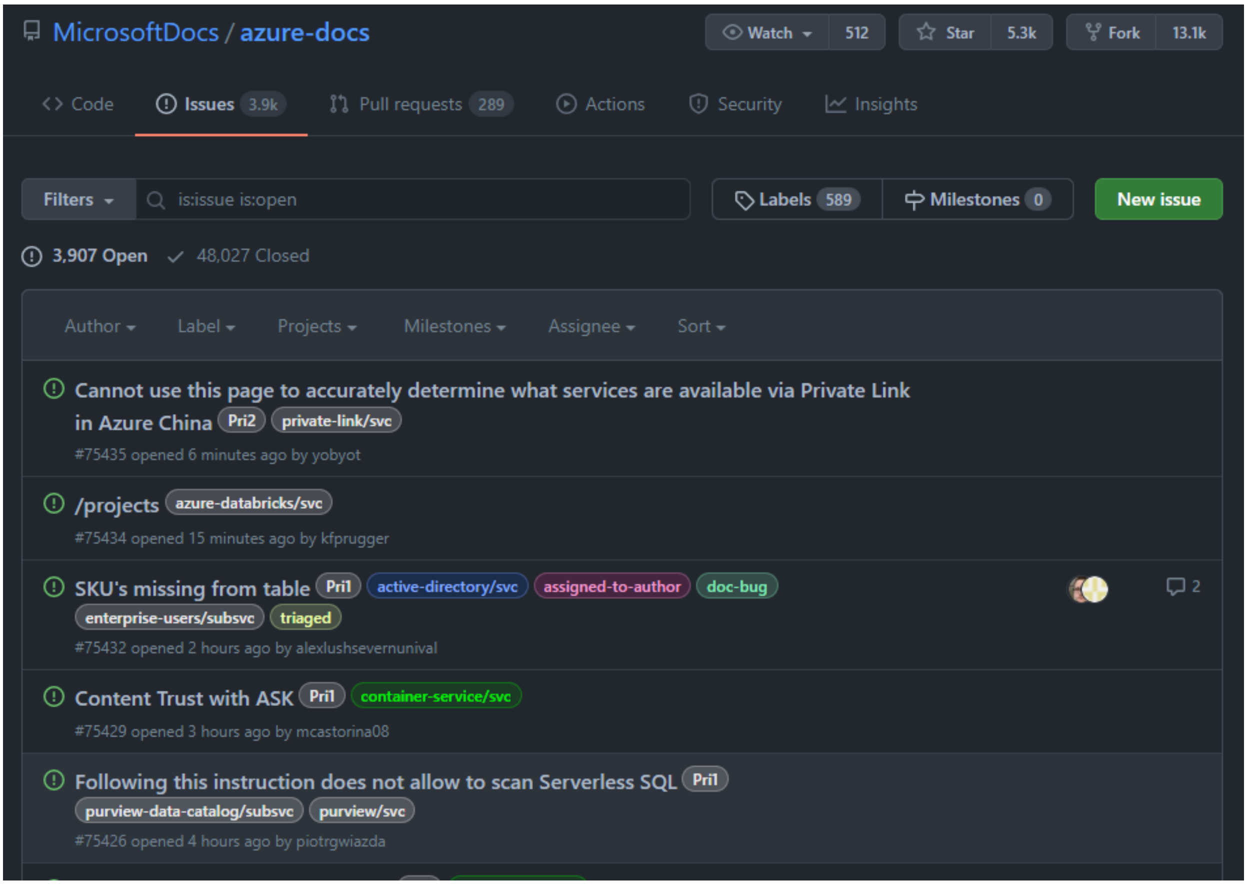
- You have people who are assigned to triage and respond to new issues regularly, daily if possible.
- Team members responsible for triage understand the issue classifications and process described later in this article; using our standard labels enables reporting and it allows you to identify feedback that needs to be handled by the right content partners or feedback endpoints.
- You have identified support channels you can direct product questions, support issues, localization feedback, site design feedback, and other feedback that isn't documentation related.

Types of feedback

If enabled, each topic on `docs.microsoft.com` gives readers the option of submitting feedback on either the topic itself or the product. This choice improves the flow of information directly to the product team while reducing the amount of non-content feedback "noise" we receive and have to triage.

- Selecting **This product** takes readers to a destination (for example, UserVoice, a GitHub repo, an email address) that allows them to provide feedback on the product itself. This feedback is independent of the content and has no relationship back to the original topic. The intent is for the product team to own this feedback process from end to end. This article doesn't discuss this type of feedback.
- Selecting **This page** opens the commenting user interface for the reader. They'll sign in to GitHub and submit feedback on the page.
- Selecting **View all page feedback** allows readers to see all feedback submitted on the topic. Anyone can read the feedback, but readers must sign in to leave feedback.

When a reader submits feedback on a topic, GitHub creates an issue in the public repo that contains the topic's docset. The issue links to the submitter of the feedback and the topic it was submitted on.



Alternatively, readers may create issues directly in your repo using GitHub. Regardless of how readers submit issues, triage them the same. Issues created in GitHub will lack a topic "Details" section. If the issue contains insufficient information, assign it to the creator and assign the `needs-more-info` label from the list of labels below.

Triage process

One of the advantages of building the docs.microsoft.com feedback system on top of GitHub issues is that all the patterns, practices, features, and tools available through GitHub are at your disposal to help you maintain your content, address your issues, and ensure your customer's satisfaction.

One of the most powerful tools in the open-source project maintainer's toolbox is issue and pull-request (PR) triage. Consistent triage, categorization, and directing non-doc issues to proper channels can be a great way for a content team to start the day and to ensure all incoming feedback receives a response. Whether it's a single person or an entire team committing their time to triage, you need a clearly defined process to ensure your team addresses all customer feedback and follows up in a timely fashion.

The triage process involves:

1. Assigning issues to yourself or others.
2. Labeling and classifying issues.
3. Replying to issues.
4. Staying engaged.
5. Closing issues.

This article details each step of the process below.

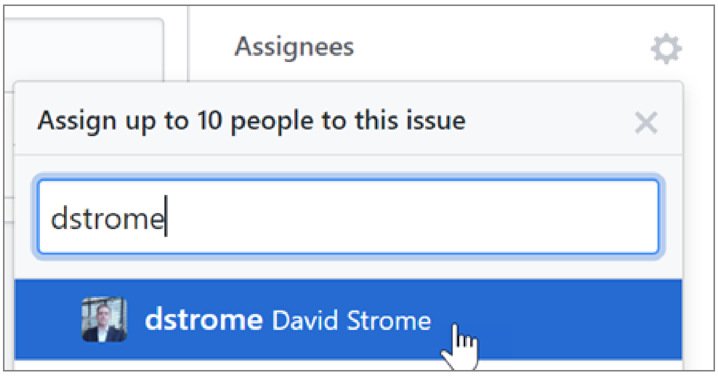
Assign issues

Depending on the repo, issues may be assigned to you by the repo admin, or you may need to assign yourself or someone else on your team. Either way, you'll receive a notification for the issue. Assignment is important because each issue needs an owner to drive it to resolution. Being assigned to an issue will send more notifications to the assignee when someone comments on the issue on docs.microsoft.com or on GitHub.

To assign an issue to yourself or someone else:

1. Navigate to the issue. Select the gear icon next to **Assignees** and type in your GitHub username (or the username of the person you want to assign this issue to) in the **Filter people** textbox.

2. Select your GitHub account when it appears in the list. Your GitHub username will appear in the **Assignees** list.



You're now assigned to the issue. If you assigned others, they'll be notified that you've added them to the issue. You can assign up to 10 people to an issue.

Label and classify issues

Labeling is key to triaging GitHub issues filed by customers against our content. Regular triage using our standard label set will improve your understanding of your GitHub documentation issue backlog. Regular triage also allows you to quickly direct customers to the correct location for their feedback.

Standard classification labels for GitHub issues

When you triage issues, you need to label each issue for its type - you need to classify them. Every team should align to the standard-issue classification labels when triaging issues created against their repo. The standard labels not only make it easier for your team to manage its issues but also supports standard concepts in broader discussions and our reporting.

Issues not related to documentation

- **code-of-conduct** - Assign to issues that are spam, trolling, or that otherwise violate Microsoft's code of conduct.

- **support-request** - A support-style question where the customer needs help with solving a problem. GitHub label color. Explain that a dedicated channel exists for their support request and direct them to it. Once you've provided the customer this information, you should close the issue.

- **docs-experience** - The issue contains feedback about the docs.microsoft.com experience or design. The customer should be directed to this location and the issue closed.

- **product-feedback** - Indicates that the feedback was related to the product itself. Explain that a dedicated channel exists for their feedback and direct them to it. Once you've provided the customer this information, you should close the issue.

- **product-question** - Typically, these questions are tightly focused on product capability or functionality, they aren't a general consulting request. Once you've provided the customer this information, you should close the issue.

- `needs-more-info` - The issue assignee, author, or product team has requested the customer provide more information about the issue they submitted before the issue can be classified into one of the following categories. These issues are assigned to the customer who filed the issue.
- `duplicate` - A duplicate of another piece of feedback.

Documentation-related issues

- `doc-bug` - If the feedback is about something that is out of date, unclear, confusing, or broken in the article, assign the bug label. This includes feedback about missing information that is blocking the customer's success.
- `doc-enhancement` - If the feedback is about suggested additions/improvements to the article, but there is no evidence the customer was blocked, assign the improvement label.
- `doc-idea` - The issue is a request to collaborate on or contribute new documentation.

Product or Service area labels

If your repository contains articles about different services or technology areas for a single product, you may want to use the GitHub automation app to have PR Merger apply labels for these values when a customer creates a new issue.

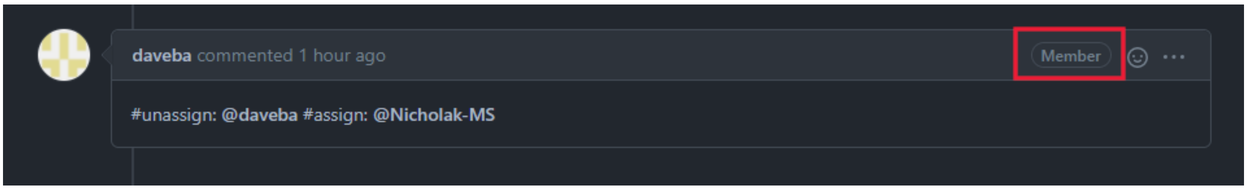
> **NOTE**
>
> The GitHub Issue and PR report uses both topic metadata and applied labels to enable sorting by product and service labels so using this automation is not a requirement.

Priority labels

The GitHub automation app can also apply Pri1 – Pri3 labels to issues based on the number of page views each month. If you want to sort your issues by priority in the GitHub Issues and PR report, you'll need to enable this automation.

Reply to issues

Anyone who has a GitHub account can reply to comments on docs.microsoft.com and on GitHub. Microsoft employees and vendors who are members of the MicrosoftDocs or Microsoft organizations have a "member" badge next to their name on their replies:

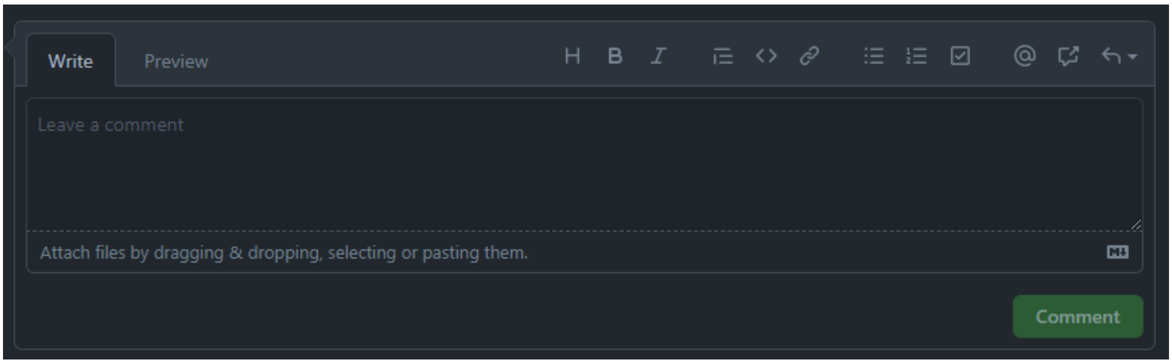


To reply to an issue:

1. Click on the link in the notification for your issue.
2. Log into your GitHub account if you're asked to do so.
3. Write a response in the editor field. In this field, you can:
   - Use Markdown by entering Markdown syntax or using the formatting bar at the top of the editor.
   - Attach files by dragging them into the editor field.
   - @-mention other GitHub users. People mentioned in this way will receive a notification.

4. Optionally, select **Preview** to see how your reply will look.

5. Click **Comment** to post your reply.



When you submit your reply, it will show up immediately as a reply to the original comment on the topic on docs.microsoft.com.

Remember that when you reply to an issue, you're speaking to Microsoft customers. Use the guidelines below to guide your response.

- **Keep the message concise** – Being concise exudes confidence and shows the customer that you know what you're talking about. Only explain what answers the customer's question, but ensure you have not left anything out. Concise doesn't mean short; it means ensuring you communicate just what's needed to make your point.

- **Be honest** – Honest answers build trust with our customers. The answer may not be the one the customer wants, but if we build trust with them, they'll have an easier time accepting an answer from us because they'll know it to be honest and true.

- **Be inquisitive** – If a customer is suggesting an idea to us on how something could be better, take some time to really understand what they're asking for. If we don't completely understand, ask them questions to clarify what they mean. **Example**: "You mentioned that instead of doing X, you did Y. What was the reasoning for doing Y? Can you expand on your current situation?"

- **Thank them** – Begin the response by thanking the customer for taking the take to make our content better. **Examples**: "Thanks for reaching out," or "Thanks for the question."

- **Acknowledge them with empathy** – If the customer is upset or struggling with using the content, it's important that we acknowledge their question. **Example**: "I'm sorry to hear that you're having a difficult time with our content."

- **If they're upset, align with the customer** - If a customer is upset, it's important to let them know that you hear their frustration and understand how hard it is for them. Careful consideration should be taken into account when aligning with the customer. Align with them only when it is obvious they are upset; otherwise, it may come across as pandering. **Example**: "I know that trying to learn a new technology is difficult, and confusing content can make the task even more challenging."

- **Know when to get help** – Some issues may require a response that you don't have. An example might be: "When is product X going to GA?" A response to this question has many implications. Careful consideration and discussion with the engineering team or other partners may be needed. Additionally, the question or issue may require another individual to respond to the customer. If so, contact to the person outside of the thread to ask for support.

Stay engaged

Usually, people who submit issues on GitHub are invested in the site and engaged in its community. They expect the same of us. Continue to monitor the issues you own after you've replied. Below are tips for staying engaged.

- Check your open issues daily for new replies, and respond as needed.
- If you need to research or follow up with others to resolve the issue, make your intentions clear when you first reply to an issue. Don't let more than a few days pass between updates. If it's taking longer than expected to get a response to an issue, update the thread letting the submitter know. Frequent communication builds trust with the customer.
- Know when to quit. Some customers may continue the conversation after you've closed the issue or if they're upset with the outcome. Carefully read their response to see if it requires a response. If the customer is venting or repeating their question, a response may not be warranted. If the situation gets too uncomfortable, contact your manager to review the response. If needed, we can always include CELA to help us determine how or if we should respond.

Close, edit, or lock issues

Closing an issue is equivalent to resolving it. When you close an issue, it will be moved from the **Open** list of comments on a topic to the **Closed** list. Closed issues on GitHub are filtered from the active list but can be accessed using the **Closed** filter.

Close issues when:

- The submitter agrees the issue is resolved.
- You've updated the content to reflect the feedback in the issue.
- The feedback isn't actionable, and the submitter hasn't responded to requests for clarification.

To close an issue, post a reply politely indicating why you're closing the issue. Next, depending on the repo, either select **Close and comment**, or use the hashtag command `#please-close` command in the issue comments.

If a customer follows up after the issue has closed and disagrees with your decision to close the issue, continue to engage with the customer to drive towards a mutually satisfactory resolution. When in doubt, reopen the issue and continue politely towards resolution.

Editing customer comments is discouraged and should be a rare occurrence. If for some reason you do edit a customer's comment, add a note that you edited it and describe what was changed.

Lock issues only if they violate the Code of Conduct. The community does not look favorably upon locking a conversation, so you must involve leadership in the decision. For more information, see Code of Conduct.

Azure-CXP maintained repos

In addition to the standard label set, the common label is a triage status indicator for public PRs:

LABEL	DEFINITION	ACTION REQUIRED
assessing_(C&L)	A PR reviewer has taken action and it's assigned to the author or another SME	No action required by the assignee. For the team helping to do assignments, it saves us from having to open lots of PRs, since we can easily see they are already in progress

Request an issue be assigned to another user or to another category

As repo readers, if you know an issue is assigned to the wrong person, you can reassign issues (`#assign:<GitHubID>` , `#reassign:<GitHubId>`), set priority (`#pri1` , `#pri2` , `#pri3`), or update labels. For more information, see Hashtag commands for managing issues with Read permissions.

If you need assistance with issue assignment, email the AzureCXPDocFeedback alias.

If you feel an article has been incorrectly categorized as a documentation-related issue, email the AzureCXPDocFeedback alias. Explain why you feel the article has been miscategorized, and indicate what the category should be. The CXP team will assign the `category-reassigned` label to these issues and assign the correct category label.

Contact Azure CXP team for issues in the azure-docs repo

To contact the Azure CXP team, send mail to AzureCXPDocFeedback alias.

Code of conduct violations (code-of-conduct)

The Microsoft Open Source Code of Conduct lists the following behaviors as disrespectful and unacceptable:

- 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 behaviors that are in violation of the Microsoft Open Source Code of Conduct. For more information, see the Code of Conduct FAQ, or contact opencode@microsoft.com with any questions or comments.

For guidance on how to handle cases where someone uses the feedback system in a way that violates the Code of Conduct, see Respond to inappropriate content (GitHub issues and comments).

Other GitHub-related features you can use with issues

Associating PRs with related issues

In your PR, you can use keywords in your commit message so that GitHub automatically closes the related issue. For the full list of keywords and instructions, see Linking a PR to an issue using a keyword. The keywords work across repositories, so you can use them in a private repo PR to close a public repo issue. The customer who filed the issue will receive a GitHub notification when the issue is closed in this way.

Notifications

When customers create an issue, they'll automatically receive email notifications on that issue. You can @-mention GitHub users in the issue description and comments, which will send those individuals email notifications as well. For any issue, you can visit the Issues page on GitHub and set the level of notifications you want or manually subscribe to that issue. If you want to be notified for a broader set of activity, you can subscribe to all issues on a repository, or you can subscribe to all activity for a GitHub organization using RSS.

Resolving "stale" GitHub issues

3/5/2021 • 4 minutes to read

This article defines the 'stale' issue concept and provides guidance for writers and teams to meet the challenge of managing the backlog of GitHub issues.

When is an issue considered 'stale'?

An issue is considered *stale* if it was created over 180+ days ago and has no new comments for 15+ days.

However, beyond simply using **created date** as your filter, you should take a moment to consider situations where an issue is stale, but the customer may still be 'checking in' on it. Taking recent activity, priority, or upvote/comment activity into consideration could help you avoid closing active issues on customers.

Scenarios

Since the goal of this process is to help our writers direct their available time to newer issues, we need to look for opportunities to not invest in 'stale issues'.

In all scenarios there should be a focus on acknowledging the experience the customer had, providing context, and concluding the exchange.

This process **should not be automated** and provided boilerplate examples should be adjusted as needed to fit the content of the issue.

The following scenarios categorize the different kinds of stale issues and suggested response messages to use when closing issues.

Scenario 1: Stale issue that is not triaged or assigned, but is still open

In situations where there has been no clear engagement on the issue, you have the option to close out.

Boilerplate example:

```
Thanks for your dedication to our documentation. Unfortunately, at this time we have been unable to review
your issue in a timely manner and we sincerely apologize for the delayed response. We are closing this issue
for now, but if you feel that it's still a concern, please respond and let us know. If you determine another
possible update to our documentation, please don't hesitate to reach out again. #please-close
```

Alternatively, you could also comment on the stale issue to ask if the issue is still current, and if a response is not received within a time range of your choosing, you can close the issue.

Scenario 2: Stale issue that is a doc-bug, doc-enhancement, or doc-idea that was fixed, but the issue was left open

This scenario is confirmed as fixed, either through issue comments or while triaging an untriaged stale issue.

> **NOTE**
>
> If an issue has already been triaged and comments do not identify resolution, it should be handled with Scenario 3

Boilerplate example:

```
Thanks for providing feedback that helps improve our documentation. This issue has been resolved and we are
closing the issue. #please-close
```

Scenario 3: Stale issue that is a doc-bug, doc-enhancement, or doc-idea and assigned to an owner, but not fixed

Similar to scenario 2, but in this instance the issue is not fixed.

Here you should review the potential doc action and decide if the work is already present in an existing work item for content, and if so, add the feedback to the item. Otherwise, create a work item.

The goal here should be to indicate why we are closing the issue, acknowledge, in a measured way, the low-quality feedback experience, and indicate that the issue raised is now being tracked internally.

Boilerplate example:

```
Thanks for your dedication to our documentation. Unfortunately, at this time we have been unable to review
your issue in a timely manner and we sincerely apologize for the delayed response. The requested updates
have not been made since the creation of this issue, and the timeline for resolution may vary based on
resourcing, so we've created an internal work item to incorporate your suggestions. We are closing this
issue for now, but feel free to comment here as necessary. #please-close
```

Scenario 4: Stale issue that is not doc related, but the writer or product team engaged

As an organization, we are not resourced to address or direct feedback that isn't doc-related. Typically issues of this type would be redirected quickly after they are created.

However, the last thing we want to do is close the door on an open engagement. If an assigned *owner* has clearly engaged the issue, the first step is to comment to the *owner* for a follow-up.

If the assigned *owner* is no longer in role, or at Microsoft, see Scenario 5.

Scenario 5: Stale issue that is not doc related and there is no clear engagement

The goal here should be to acknowledge the untimely response, explain why this issue type isn't within scope, and provide the correct channel for future feedback of that nature.

> **NOTE**
>
> Proper product and support feedback channels differ from team to team, so make sure the resource you provide is relevant to your space.

Boilerplate example:

```
Thanks for your dedication to our documentation. Unfortunately, at this time we have been unable to review
your issue in a timely manner and we sincerely apologize for the delayed response. Though the scope of our
feedback channel here on GitHub covers specific documentation fixes, we can help redirect you to the right
support channel to get an answer to your question: [reference to alternative product/service or community
endpoint]. Because this issue does not relate to documentation, we are closing this issue. #please-close
```

Pull request comment automation

7/7/2021 • 2 minutes to read

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 **#sign-off** in the comment stream, the **ready-to-merge** 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 *isn't* 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 **#hold-off** in a PR comment to remove the **ready-to-merge** label. In the private repo, **#hold-off** assigns the **do-not-merge** label.	Public and private
#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
#please-open	Authors can reopen a closed pull request.	Public
#label:"custom label text"	Authors can add a custom label up to 200 characters (shorter recommended).	Public and private

See the **Pull request processing** sections of the minor/infrequent and major/long-running contribution workflows, for more context on how comment automation is used in each.

PR (pull request) review quality criteria

7/14/2021 • 26 minutes to read

This article explains the list of quality criteria checked during the PR (pull request) review process.

These criteria are for:

- Authors who create and maintain Docs technical articles and Learn content.
- PR reviewers who provide editorial review of pull request content quality.

If your PR doesn't qualify for automatic merging, a human pull request reviewer reviews it against these basic quality criteria. The PR review isn't a technical review of the content. The review covers only what is new or changed. The reviewers call out only the blocking and non-blocking items that are listed in this article.

Request a pre-review

Content authors can request a pre-review of a pull request to get early feedback. The prerequisites for a pre-review are:

- The content should be mostly complete and almost ready for publish from the author's perspective.
- The author has requested a pre-review.

PR Review criteria update in progress

As of July 14, 2021, all Docs repos (but not any Learn repos) are running under the new, smaller set of PR Review criteria covered in this section. (Learn repos will tentatively adopt the change in August.) This project has phased into 29 repos, since beginning in January 2021. Data shows the streamlined criteria increase authoring speed, decrease repeat reviews and drive contributor satisfaction.

The primary change at this phase of the project is that PR Reviewers are empowered to self-fix 9 of the criteria, when there are no other blockers, and continue to merge. This eliminates contributor re-work and repeat reviews in many cases. Here are the criteria that reviewers can proactively fix on behalf of contributors:

- The title attribute should be sentence cased in YML and MD files
- No valid spelling errors remain in the Acrolinx report
- No remaining do-not-use terminology
- No Bylines
- Elements that should be numbered lists are markdown numbered. Elements that should be unordered lists are markdown bullets.
- Custom markdown elements are used in a limited fashion and must be formatted correctly
- The H1 title contains sufficient information to describe the content of the article, to differentiate it from other articles in our content set and to map to likely customer keywords
- The H2 headings, when rendered in the on-page TOC, should ideally wrap to no more than two lines
- All titles and headings are sentence case, per MSFT style

The other key change introduced by the project is that 12 criteria will no longer be Blocking, removing hurdles for contributors. This is because the criteria may no longer be relevant, may be process rather than a criterion that's evaluated by the PR Review team, or is better handled as a "best practice" tip elsewhere in the Contributor Guide. These criteria are:

- The "ready-to-merge" label is assigned to the PR (applied from automation after the #sign-off comment), and the validation status is "passed."

- The #sign-off comment must appear after the validation results and staging links. If the #sign-off comment appears before the staging links, the author didn't review the staged content.
- Articles cannot use the terms "General Data Protection Regulation" or "GDPR" outside the context of the CELA-approved includes listed in the GDPR guidance posted in the contributor guide.
- Switchers are used only for switching across multiple versions of the same article.
- In article sets that use switchers, the H1 in each article contains information that differentiates each article from the other articles in the set.
- The use of the "ARM" acronym, or the use of V1 or V2 as a reference to the classic and Resource Manager deployment models in Azure, is a blocking terminology item.
- PR reviewers might provide feedback on a few minor spelling, grammar, and other writing issues as non-blocking feedback. If there are more than a few editorial issues, reviewers log an edit request for the article for a post-publication edit.
- If the pull request could easily be configured to benefit from PRMerger automation, pull request reviewers provide feedback to the author about how to use branches so the changes can be merged automatically.
- All external hyperlinks resolve to content that exists and that appears to be high-quality and credible.
- For most contributors, the initial publish of a new module must merge to the master branch directly from a release branch, and content development work for a new module must merge to a release branch. A PR that targets the master branch with a new module from a personal fork is allowed only from contributors that have been vetted by the Learn team.
- For new modules, no more than one module per PR. Updates to existing content can go across modules in a single PR.
- Module content should be for the product area scoped for the repository.

Aside from what's mentioned in this section, the below still applies. This page will be updated as the project continues to roll out.

Blocking content quality items

The updates in the pull request must meet the following criteria to be merged. Pull request reviewers provide feedback in pull request comments for these items and type `#hold-off` in the pull request to return it to you (the author) with feedback.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Prerequisites	The "ready-to-merge" label is assigned to the PR (applied from automation after the `#sign-off` comment), and the validation status is "passed."	Ensures the author intends to hand off the PR for review and that the PR has passed the build tests.
All	Prerequisites	Close any PR against the live branch. The user should be redirected to the master branch or to a release branch.	Only publishing PRs are allowed against the live branch.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Learn	Prerequisites	For most contributors, the initial publish of a new module must merge to the master branch directly from a release branch, and content development work for a new module must merge to a release branch. A PR that targets the master branch with a new module from a personal fork is allowed only from contributors that have been vetted by the Learn team.	Prevention of live site incidents. Too many modules have been submitted against the master branch and accidentally #signed-off as ready by the author (because they didn't understand the implications of pointing to master), but the modules were not ready. This causes publishing escalations to deprecate the content from the site, implement redirects, and rebuild the release branch.
All	Prerequisites	The PR cannot be blocked by a merge conflict. If a merge conflict exists, refer the user to Resolve simple merge conflicts on GitHub for instructions on how to use the GitHub UI to resolve merge conflicts. PR reviewers don't resolve merge conflicts for contributors.	Impossible to merge, sets expectation for who is responsible for resolving the conflict.
All	Prerequisites	The `#sign-off` comment must appear after the validation results and staging links. If the `#sign-off` comment appears before the staging links, the author didn't review the staged content.	Process requirement to help encourage authors to review staged content.
All	Prerequisites	All articles in the PR must have an Acrolinx score of 80 or higher (where Acrolinx enabled in the PR queue). Limited exceptions are available.	Helps ensure baseline editorial quality while allowing wiggle room for items that are correct but not in the Acrolinx dictionary.
All	Repo integrity	The PR contains no obvious content regressions, such as unintentionally reverted dates, branding changes, etc.	Helps ensure content integrity.
All	Repo integrity	No article-related files, images, or folders are being added to the root directory of the repo.	Helps support appropriate repo management and folder structure so the root does not become cluttered.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Repo integrity	The repository administrator and FTE program manager have to review PRs that make a non-routine change in a configuration file or in the root folder, such as `docfx.json`, `.openpublishing.publish.config.json`, and any `.git*` file. For example, the admin needs to review PRs that enable a new content type. Routine changes can be merged by the review team. Routine changes include changes to the render context feature for TOCs and breadcrumbs, updates to product feedback links, and adding links to samples repositories.	Helps ensure major configuration changes are reviewed by a repository owner/admin.
All	Repo integrity	The PR does not include an embedded repo or any unusual, extraneous files. All file updates should be restricted to the articles and includes folders in the repo. Items to watch for: `.DS_Store`, `desktop.ini`, `.gitignore`, the entire repo embedded in the root folder.	Prevents repo from becoming random file storage, prevents cloning mistakes by users from being replicated to other users.
Docs	Repo integrity	The PR contains fewer than 100 changed files, unless the PR is intentionally updating a release branch from master. See the section in this article about large pull requests.	Ensures iterative workflow and reviewability of changes.
Learn	Repo integrity	For new modules, no more than one module per PR. Updates to existing content can go across modules in a single PR.	Ensures iterative workflow and reviewability of changes.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Repo integrity	If articles are deleted in the PR, the deletions must be by the listed author. Where PRMerger is running, authors who want to delete an article must list themselves as the author in a commit that precedes the commit to delete the articles. This change allows you to avoid a validation warning.	Ensures that deletions are intentional, in the past people have signed off on deleting files that were not part of their intended file changes.
All	Repo integrity	When an author deletes an article, the master redirect file must contain a redirect for the deleted content. If the repo uses a master redirect file, redirects use only the master redirect method. File-based redirects are not allowed in repos that use master file redirection.	Prevents usage of outdated redirect method.
All	Repo integrity	Only markdown (.MD), YAML (.YML), and image files are allowed in content repos.	Prevents repo from becoming random file storage. Ensures content is meant for publication as official docs.microsoft.com content.
All	Naming	New files and folders introduced into the repo follow the File name and path guidelines.	Supports consistency, repo management, SEO guidelines, and prevents Git problems related to casing.
Docs	Metadata	All includes must contain a metadata section. See details on include metadata requirements.	Ensures that includes have basic metadata so we know who owns them and what service they are used with.
Learn	Metadata	Markdown (.MD) and YAML (.YML) files must meet the detailed Learn metadata requirements.	Basic compliance for module titles, navigation, and to support SEO metrics.
All	Metadata	The `title` attribute should be sentence cased in YML and MD files.	Basic compliance with editorial guidelines for text that may be displayed in search results.
Learn	Metadata	YAML (.YML) files must not include `ROBOTS: NOINDEX`.	The platform automatically applies this attribute. Manual settings revert the global update.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Acrolinx scorecard	No valid spelling errors remain in the Acrolinx report. The author must fix the spelling errors.	Supports basic editorial quality for content credibility.
All	Acrolinx scorecard	No remaining do-not-use terminology. The author must correct the terminology.	Ensures key terminology is used per Microsoft guidelines published in the Cloud style guide.
Docs	Content	The article is a technical document covering technical subject matter and therefore is in the correct content channel. See the what goes where guidance.	Ensures that content is published to the correct content channel.
Learn	Content	Module content should be for the product area scoped for the repository. See the detailed mapping of content to repo. learn-pr: Cloud & AI Content learn-bizapps-pr: Power Platform learn-dynamics-pr: Dynamics 365 learn-m365-pr: M365 LearnShared: Learn landing, support pages If content appears to be in the wrong repo, contact learn-repo-managers.	Ensures content is localized in the correct language sets and is easy to find for maintenance and support.
All	Content	All information in an article is meant for the general public. docs.microsoft.com is exclusively for technical documentation that is available to the general public. Do not publish private preview content, content subject to NDA, or content that is otherwise confidential to the site.	Helps prevent disclosure of embargoed content or content protected by NDA.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Docs	Content	TOC files: when a new article is added, a new TOC entry is added; when a file is deleted, the TOC entry is either removed, or it is modified to point to the replacement article. In either case, the updated TOC file is in the same PR as the new/deleted file. In the content model, samples articles are not listed in the TOC, they are listed in a curated page.	Prevents the publishing of hidden content; all content is meant to be discoverable on docs.microsoft.com.
Docs	Content	Content related to Cognitive Services must be reviewed and approved by a member of the Cognitive Services content team before it can be reviewed by the PR review team.	Ensures alignment of content written by infrequent contributors.
Docs	Content	If a hub page is modified, approval by the designated business approver is required. Minor fixes to hub pages such as spelling fixes and link correction or replacement do not need approval. Addition of any new content or removal of existing content requires approval. To identify a hub page, the **ms.topic** metadata is set to **hub-page**. Landing pages no longer require business approval.	Ensures design alignment for hub pages.
All	Content	Bylines are not permitted. If an article calls out the name of the author or any contributor in the text, that attribution needs to be removed. Articles published from the tech content repo are considered to be authored by "Microsoft." Contributors who have committed updates to the article are recognized automatically on the contributor bar of the published article.	Supports automatic author/committer recognition functionality on the site. In the past, more obvious author recognition led customers to view our docs as blogs and they asked where our docs went. This led to the current minimizing of authorship on the page.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Learn	Content	In markdown (MD) files, only H2 and lower headings are allowed; do not allow authors to add H1 headings.	H1 headings are automatically applied by the system, using the `title` value defined in the associated YAML.
Docs	Content	The article contains an introductory paragraph and a procedural or conceptual body of content. The article needs to contain sufficient, complete content to stand on its own as an article. It should not be a small fragment of information. (An exception is a "Limits" article if it's in the context of a large article that lists all of the limits of a service.)	Ensures content is minimally complete prior to publish. Very short articles are known to generate low CSAT. Very short articles often indicate another solution is needed for the content problem (such as a redirect).
All	Content	Elements that should be numbered lists are numbered. Elements that should be unordered lists are bulleted.	Supports basic editorial quality so procedural steps are set up correctly.
All	Content	Unusual or novel graphics, information architecture or structures, or nonstandard designs need to be vetted with the PR review program manager. Teams that are experimenting with new things need to have a plan in place for evaluating experiments. Contacts: Docs - justinc; Learn: barlan + asjohnso.	Helps ensure alignment with general content standards by providing PR reviewers a path for escalating content that is far outside the norm for further review by content leads.
Learn	Content	Each new module must contain at least one knowledge check or task validation.	Provides consistency for learners.
All	Legal/compliance	Articles cannot use the terms "General Data Protection Regulation" or "GDPR" outside the context of the CELA-approved includes listed in the GDPR guidance posted in the contributor guide.	Ensures content aligns with CELA-required guidelines for content about GDPR.
Docs	Site/design functionality	Switchers are used only for switching across multiple versions of the same article.	Ensures alignment with intended purpose of docs UI element.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Site/design functionality	CodePen functionality is a new feature not authorized for broad usage at this time in C + AI technical content. Only approved pilot content should use the CodePen iframes (Location-Based Services is the approved pilot). Any pull request that contains CodePen iframes must be approved by Martin Ekuan.	Prevents overuse of a feature that is intended for limited usage at this time.
Docs	Site/design functionality	In article sets that use switchers, the H1 in each article contains information that differentiates each article from the other articles in the set.	Ensures that published article titles are unique for SEO purposes and so customers can differentiate between flavors of the article.
Docs	Site/design functionality	Each new include file is used in at least one Markdown article. New include files can't be added to the repo before they are used in an article. An exception exists in repos where a new include file that has "Applies to" content (used to identify the platform support that appears at the top of most articles in the docset) can be added without an update to an article.	Prevents orphaned includes.
Docs	Site/design functionality	A manually authored on-page TOC is not permitted in an article. The article must rely on H2 headings for its on-page TOC.	Ensures the automated TOC based on H2 headings is the only on-page navigation; prevents duplicate stacked TOCs in mobile and narrow page views.
Docs	Site/design functionality	If H2 headings are present, the article contains at least two H2 headings. Using one H2 heading creates a single-item article TOC. H2 headings must be used before H3 headings to ensure that a TOC is created.	Basic editorial guidance, ensures on-page navigation offers at least two clickable options.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Markdown	Source content contains minimal HTML at the block level. Minor inline HTML is permitted - such as superscript, subscript, special characters, and other minor things that you can't do with Markdown. HTML elements that are permitted by the docs platform (HTML allowlist) are not allowed in most situations. HTML tables are allowed *only* if the table contains bulleted or numbered lists. Often, an HTML table is an indication that the content needs to be simplified so the source can be coded in Markdown.	Ensures alignment with basic content authoring guidelines for docs.microsoft.com. Markdown is our authoring standard.
All	Markdown	Custom Markdown elements are used where appropriate. For example, notes are coded through the [!NOTE] extension, not as plain text.	Ensures consistent customer experience and rendering where customer markdown extensions provide design functionality on docs.microsoft.com
Docs	SEO	In all repositories, ensure that a branded product name is present in either the **title** or **titleSuffix** attribute. Note that the **titleSuffix** value may also be set globally in the docfx file for the repository. In Azure content, "Azure" must be present in one of the attributes (Intune, StorSimple, and Microsoft Genomics articles in the Azure repo are exceptions). Because branding can be defined for a folder or a repo, PR reviewers verify branding in staged content. Exception: the cpp-docs-pr repo does not use product branding for the C++ content.	Supports key SEO guidance.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Docs	SEO	The H1 title contains sufficient information to describe the content of the article, to differentiate it from other articles in our content set and to map to likely customer keywords. For example, "Overview" as the H1 title is generic and provides no useful information to a customer or to search.	Supports SEO and discoverability of our content in search.
All	Terminology	The use of the "ARM" acronym, or the use of V1 or V2 as a reference to the classic and Resource Manager deployment models in Azure, is a blocking terminology item. Exception: these terms are allowed in nonvisible text, such as the **displayName** field in TOC files, to support SEO.	Supports key Azure terminology guidance.
Learn	Terminology	Do not use the word **quiz** anywhere in the content. The only acceptable use is the YAML **quiz:** property.	Learn has standardized on using the term `knowledge check` for public facing content since it is friendlier and does imply compliance.
All	Images	Images have clear resolution, are free of misspelled words, and contain no private information. All images must render and be legible in both light and dark views on the docs site.	Supports basic quality and usability of content.
All	Images	SVG is the preferred format for conceptual art – it scales regardless of the size of the browser window. However, SVG files that are scripted (where `<svg>` elements are present with enclosed `<script>` tags) are not permitted. PNG is also fine for conceptual art and the preferred format for screenshots (to avoid the need to convert the screenshot to vector format).	Code around SVG files is stripped from the build automatically, this prevents wasted effort troubleshooting content that won't render.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Images	All media links must be to media within the repository; no hyperlinks to media hosted on my-sharepoint.com, personal URLs, other websites, or other repositories.	Ensures all media is managed by Microsoft, ensures media links are validated, future proofs for migration.
All	Staging	The content preview must be clean on staging in both light and dark view. It cannot contain any obvious formatting issues: - A numbered or bulleted list that appears as a paragraph - Code in a code block appearing partly in the code block and partly outside it - List steps numbered incorrectly due to faulty indentation - Leftover merge-conflict markers - Content must preview legibly in both light and dark views. Pay particular attention to conceptual artwork rendering correctly in dark view	Supports basic usability and credibility of published content.
Learn	Videos	Channel 9 is the preferred internal platform; RedTiger is supported but no longer required. External video links must go to Microsoft channels (that is, a Microsoft-sponsored channel on YouTube).	YouTube is not available to all learners globally, and using an internal platform allows us to handle providing different content to global learners (localization, content, etc.)

Non-blocking content quality items

For these items, PR reviewers provide feedback and instructions for the author to follow up with fixes in a later PR. This feedback does not block the decision to merge. Authors should follow up within three business days with fixes.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Docs	Content	Articles should have a "Next steps" section at the end with one to three relevant and compelling next steps. Brief text should be included that helps the customer understand why the next steps are relevant.	Best practice to drive engagement with additional content.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Content	PR reviewers might provide feedback on a few minor spelling, grammar, and other writing issues as non-blocking feedback. If there are more than a few editorial issues, reviewers log an edit request for the article for a post-publication edit.	Supports basic editorial quality, content credibility.
All	Images	Images use the correct callout style and color, and screenshots use the correct border and placeholder style. See the guidance for screenshots.	Basic consistency.
All	Images	Images include alt text. See the guidance for screenshots and the guidance for conceptual art.	Basic accessibility.
Docs	Site/design functionality	The H2 headings, when rendered in the on-page TOC, should ideally wrap to no more than two lines. Longer headings make the article TOC harder to scan.	Basic design and usability, MVC content model support.
All	Style conventions	All titles and headings are sentence case, per MSFT style.	Basic consistency.
All	Process	If the pull request could easily be configured to benefit from PRMerger automation, pull request reviewers provide feedback to the author about how to use branches so the changes can be merged automatically. See the PR best practices article.	Educating authors to make use of the system.
All	Content	All external hyperlinks resolve to content that exists and that appears to be high-quality and credible.	Basic quality.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Learn	Content	Each new module must contain at least 3 units (YML files) and 3 corresponding MD files. One unit must be an introduction, one unit must be a summary, and the introduction unit must have learning objectives and prerequisites. (The inclusion of at least one knowledge check is a blocking item.)	Provides consistency for learners.

PR review and build status policies

The OPS build provides for three potential build results: error, warning, and suggestion.

STATUS	EFFECT ON PR REVIEW PROCESS
Error	Not possible to merge; the system won't allow it. The build error must be resolved.
Warning	The PR review policy is that no PR will be merged with outstanding warnings. If a warning occurs unexpectedly on a day with a major release, the repository admin can decide to merge a PR with a warning to ensure content for the release goes live. This should not be done during routine daily publishing. The warning must be fixed as soon as possible afterwards. PR reviewers will not merge PRs with warnings.
Suggestion	Suggestion status is used when new validation rules have been turned on in a repository to help educate authors. All suggestions, after an introductory period, will be turned into warnings. Users are encouraged to fix items called out in suggestions, but acting on them is optional. PR reviewers will not block PRs that have suggestion results.

PRs with more than 100 changed files

Large PRs have a demonstrated history of containing unintended changes that break, damage, or delete content.

Therefore, the best practice is to submit changes in MUCH smaller batches iteratively so that the changes can be easily and quickly reviewed. New articles should be submitted in batches of five articles or fewer - for more information, see Content release planning + process.

PRs that include 100 or more files have to be handled specially to protect the integrity of the repository. These guidelines apply to pull requests against the master branch and to all release branches.

- **Bulk file moves:** PRs that move many files will be reviewed per the process listed in the Moving or refactoring files in a repository article. The PR review team can accommodate broken link warnings when multiple PRs are used, as recommended; in these cases, links are verified only in the final PR.
- **"Dirty PRs":** When a release branch has a merge conflict with master, the PR review team verifies the "dirty PR" that is required to resolve the merge conflict and bring the release branch up-to-date with

master. PRs of this type will be reviewed per the process listed in the Resolve merge conflicts in Git and GitHub article.

- **Bulk updates**: Because the GitHub UI now shows diffs beyond 100 files under certain conditions, the PR review team may be able to review bulk changes in a large PR. To qualify, the changes in the PR must be scoped to a limited and clearly identifiable set of changes across all the files.
- Any PR that contains numerous changed or new files outside these conditions (bulk file moves, dirty PRs, and bulk updates) does not qualify as reviewable and must be broken up into easily reviewable chunks. This enables multiple reviewers to work on the reviews, it allows feedback to be provided in manageable chunks, and it facilitates an iterative workflow. If you have to break up a large PR, see Break up a single large pull request into smaller PRs.
- PRs that present special problems beyond the scope described here will be referred to the repository administrator, delaying review of the content changes. The administrator will review the scope of the PR and make a recommendation about how to proceed in a way that ensures the quality of the content and the integrity of the data in the repository. The repo admin is responsible for merging PRs like this.
- The file limit guidelines do not apply to PRs in Docs repos where an upstream release branch is being merged to an upstream master branch for publishing. The content of the release branch should have been reviewed as each change was merged to the release branch along the way, so a final PR review is not required.

Templates and stubs in release branches

Occasionally, writing teams may want to add templates for new articles in a release branch. This allows the writing team to stub out content so that file names, metadata, and TOC links are done correctly. The partners authoring the new articles can then focus on writing the needed content.

PRs that contain stubbed out content can be merged to a release branch if they meet these criteria:

- The PR contains only the stubbed out articles and TOC links to those articles.
- The PR is against a release branch.
- The stub articles use correct file naming.
- The stub articles contain complete metadata. The title, description, author, and other information should be complete.
- All content in the stub articles should be instructions about the content to write, not the actual content.
- The content team manager (business approver) must approve the PR to ensure follow-up so that the stubs are converted to full articles before the merge of the release branch.

An article is not considered a stub or template article if it contains partially written content. Articles that are partly written don't qualify for merge under this template/stub guidance.

The following PR is an example where the writer has correctly submitted stubbed out articles to a release branch: https://github.com/MicrosoftDocs/azure-docs-pr/pull/48223/files.

This guidance helps ensure that all new content goes through the PR review process and helps prevent incomplete content from being published.

PRs that require repository admin review, merge, or escalation

- Configuration file updates
- Any case where a user wants an exception to these PR criteria

Azure Architecture Center business approvals

All new articles in the Azure Architecture Center require business approval.

Process:

1. Contributor creates a new article, makes a pull request, and enters the #sign-off comment.
2. PR reviewer picks up the pull request and notices that it contains a brand new article.
3. PR reviewer checks the PNP GitHub team – if the new article is from one of the members of the GitHub team, they proceed with the review, and no business approval is needed. If the new article is from anyone else, the PR reviewer goes to the next step.
4. PR reviewer assigns the "pending-content-team/business-approval" label to the pull request and sends mail to the pnp@microsoft alias to request business approval to move forward with the review process.
5. Someone from the alias responds to the mail with approval to move forward and enters #sign-off again in the pull request.
6. Once someone from the alias signs off, the PR review team proceeds to review the new content per the standard process.

Automated PR criteria

The following pull request review criteria are now part of the automated validation checks, rather than human reviewed.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	DEPRECATION REASON
Docs	Metadata	A metadata section is present at the top of the file. The metadata section starts with three hyphens, is followed by a list of metadata that includes at a minimum `title`, `description`, `author`, `ms.author`, `ms.date`, `ms.topic`, and `ms.service` *or* `ms.prod`, and ends with three hyphens.	Replaced by build validation in the Developer Relations/CGA Reporting ruleset. Removed from list 11/8/2019.
All	Localization	Links to pages on microsoft.com websites are coded as locale agnostic. Do not include `en-us`, `en-gb`, `en-in`, or any other locale in links to these sites. TechNet/MSDN forum links are an exception--locales cannot be removed from forum links on these sites.	Ensures correct localized content experience. Replaced by build validation in the Developer Relations/CGA Reporting ruleset. Removed from list 12/20/2019. Note that this automation is not yet enabled in Learn repos. However, when discovered in the human PR review process, URLs that include locales are flagged as blocking.
Docs	Content	The article contains only one H1 heading.	Multiple H1 headings break page rendering and are bad for SEO. Enabled 6/29/2020.

Overview of PRMerger

7/7/2021 • 8 minutes to read

PRMerger is workflow automation for our GitHub content repositories that supports review workflows in public and private repositories. In private repositories, it merges pull requests automatically. The merge occurs after the author signs off when the pull requests meet the criteria for automatic merge.

PRMerger problems/bugs

If PRMerger isn't working as expected, file a bug!

1. Go to https://aka.ms/sitehelp.
2. Select **Submit request or issue**.
3. Choose **Select Service** > **OPS**.
4. Choose **Select Service Category** > **PRmerger/repo automation**.

PRMerger private repo workflow

Here's how PRMerger works in *private* repos:

- It allows read-level contributors to assign the `ready-to-merge` label to a pull request. The `ready-to-merge` label lets the pull request reviewers for the repository know the PR is ready for review.
- It applies business rules (criteria) to allow certain pull requests to be merged automatically.
- Pull request reviewers focus their reviews on new content and significant updates.

After you configure a private GitHub repository to run PRMerger, the service monitors that repo for new pull requests. PRMerger offers a review workflow that relies on labels to indicate where in the process a pull request is. The following list describes the process:

1. Adds the `do-not-merge` label to all new pull requests in all branches except live.

2. Checks the `ms.service` / `ms.subservice` or `ms.product` / `ms.technology` metadata fields. If one or more is present with a value, it adds the value as a label. This helps reviewers route PRs appropriately. PRMerger appends a suffix of `/svc`, `sub/svc`, `/prod`, or `/tech` to the label value to assist with label filtering in some CGA reports.

3. Assigns the PR and requests a review from the authors of articles in the PR. There can be up to five people automatically included in that review. PRMerger notifies them of the change and gives them a chance to provide feedback, even if the PR gets automerged. The process skips this step if there are more than 50 articles in the PR. The process assumes that it's a bulk update and doesn't need author review.

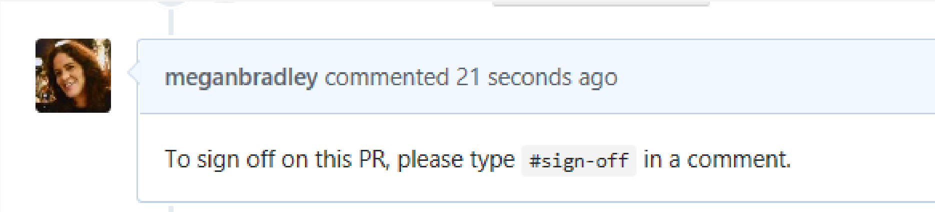
   Any Microsoft employee or vendor with Read permissions can include multiple assignees by typing `#assign:<GitHubID>`. Or, reassign the PR and remove current assignees with `#reassign:<GitHubID>`.

4. Next, PRMerger waits for a successful OPS build to complete for the pull request. If the build has errors or warnings, the OPS build writes the result to the pull request. You can't sign off until there's a clean build.

5. After a successful build, PRMerger applies its business rules. The business rules determine whether the pull request qualifies for automatic merge or whether it requires human review:

   - If all conditions are met, PRMerger adds the `qualifies-for-auto-merge` label.
   - If not, PRMerger adds the `needs-human-review` label.

6. At this point, you should review the staged content.
7. If the content looks good, type `#sign-off` as a comment in the pull request as plain text.

   > **NOTE**
   >
   > If you type `#sign-off` within any formatting, such as inline code, bold, or an HTML tag, it won't be recognized as a sign-off. The sign off doesn't happen because PR reviewers often need to provide sign-off instructions to contributors in GitHub comments, and need a way to do so without actually signing off on the PR. For example, if you type a comment like the following example:
   >
   > ```
   > To sign off on this PR, please type `#sign-off` in a comment.
   > ```
   >
   > The comment displays like this example and won't be treated like a sign-off:
   >
   > 
   >

   - If the `qualifies-for-auto-merge` label is assigned to the pull request, PRMerger checks if the GitHub `merge-ready` flag is applied. If so, your pull request is immediately merged after you enter the `#sign-off` comment.

     > **NOTE**
     >
     > In some cases there are delays on the GitHub side that prevent `merge-ready` from being applied. PRMerger will retry three times. If the PR is still not merge-ready, you'll see a message indicating a delay, and PRMerger will try again in two hours.

   - After the pull request is merged, PRMerger assigns the `merged-by-prmerger` label to the pull request.
   - If the `needs-human-review` label is assigned, your pull request will enter the queue for human review and feedback.

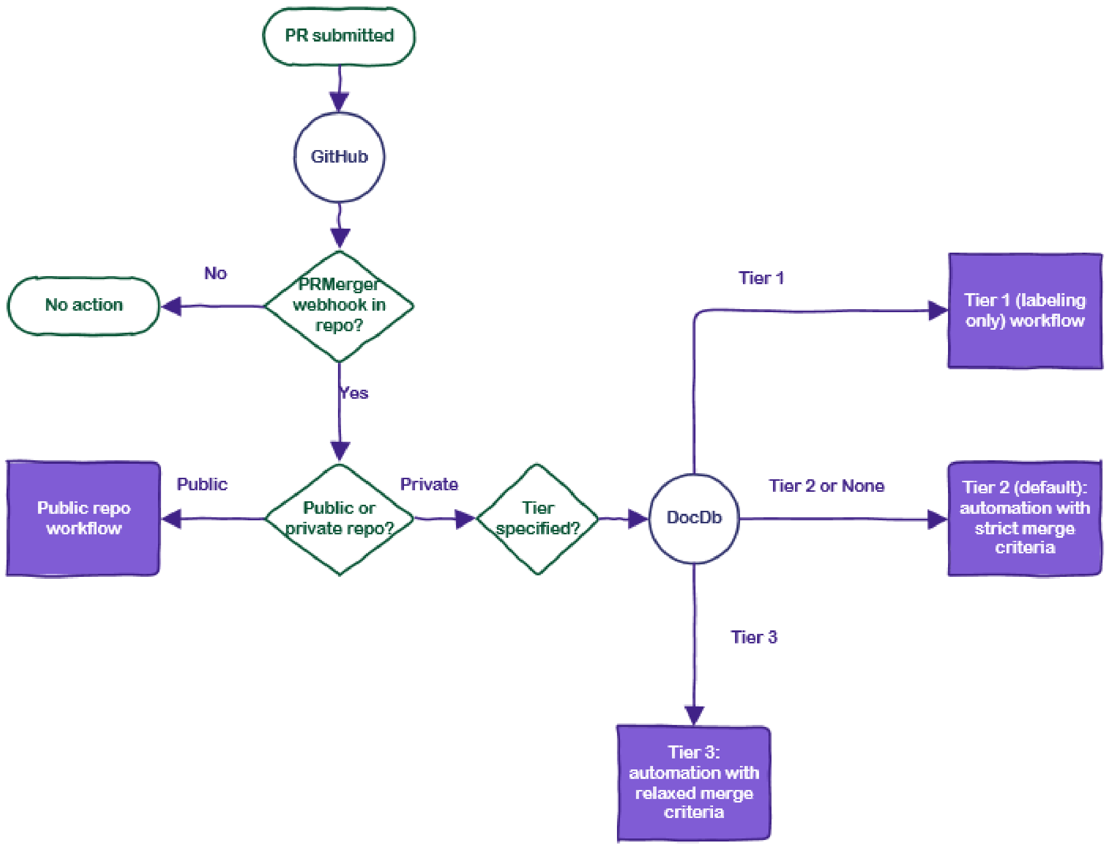
8. After the review of a `needs-human-review` pull request is complete, one of the following occurs:
   - If the pull request meets the pull request criteria, the pull request reviewer merges the pull request.
   - If the pull request doesn't meet the criteria, the reviewer provides feedback in the pull request comments and types `#hold-off` to reassign the `do-not-merge` label. After you address the feedback, the article must go through the full workflow again.

A pull request can only be merged if **the following conditions** are met:

- The build is successful.
- All PRMerger validation filters pass.
- The author adds a `#sign-off` to the pull request, which assigns the `ready-to-merge` label.
- Pull requests that require human review must meet the pull request criteria.

Private workflow diagram

PRMerger high-level



PRMerger business rules

Pull requests need to meet the following business rules to qualify for automatic merge. If a pull request fails any of these rules, it's labeled for human review.

1. PRMerger first adds a label called `do-not-merge` to all new pull requests. Based on the business requirements, a pull request can only be merged if **the following conditions** are met:

   - Build is successful.
   - All PRMerger validation filters pass.
   - The author adds a `#sign-off` comment to the PR, which triggers the process to assign the `ready-to-merge` label to the PR.

2. The Pull Request triggers the OPS build. If the build is successful the PRMerger automation kicks in, otherwise the author is notified of the failed build.

3. After the successful build, the PRMerger business filters run.

   a. If all of the *filters are successful*, PRMerger adds a label `qualifies-for-auto-merge`.

      The author reviews and adds a `#sign-off` comment. Once PRMerger sees the sign-off comment, it adds the `ready-to-merge` label, confirms the filters were successful, and the build has passed. PRMerger will then continue to auto-merge this Pull Request and add a label `merged-by-prmerger`.

   b. If the *filters aren't successful*, PRMerger adds a label `needs-human-review`.

      When this label is assigned, the selected pull request reviewers for the repository have to review the pull request. If there are no content issues, they merge the Pull Request manually. If there are, they add a comment `#hold-off` and notify the author to fix. This reassigns the `do-not-merge` label so that PRMerger won't continue with merging the PR.

PRMerger filters

For a pull request to qualify for automatic merge, the pull request must meet these criteria:

- Affects no more than 10 files.
- Contains no more than 15 commits.
- Contains no more than 20% change.
- Contains no new files or deleted files.
- Doesn't change, add, or remove any images.
- Contains no index pages, hub pages, home pages, or landing pages.
- Contains no config files, breadcrumb files, or TOC files.
- PR title doesn't contain the words "edit pass".
- There are no unsupported file extensions being checked in.
- No articles were deleted by someone who isn't listed as the author of the article.
- No files submitted into the root folder of the repo.
- No file in the PR has an Acrolinx score of less than 80 (if Acrolinx is enabled on the repo).
- No references to the General Data Protection Regulation (GDPR).
- No file in the PR contains a CodePen reference.
- No changes to the casing of file or folder names.

PRMerger public workflow

This workflow is applied to all *public* repos when PRMerger is enabled.

1. When an author submits a PR, check the `ms.service` / `ms.subservice` or `ms.product` / `ms.technology` metadata fields. If one or more is present with a value, add the value as a label. The label helps reviewers route PRs appropriately. A suffix of `/svc`, `sub/svc`, `/prod`, or `/tech` is appended to the value in the label to assist with label filtering in some CGA reports.

2. Check the submitter's GitHub ID to determine if the submitter is the `author` of one or more files in the PR. If yes, close the PR with a message that the author should use the private repo.

3. If the submitter isn't the author of one of the articles in the PR, check whether `ALL CLA Requirements Met` (determined by CLA logic - pass if the submitter is a Microsoft employee OR has signed a CLA). If no, no action. If yes, continue with the next step.

4. Assigns the PR and requests a review from the authors of articles in the PR. There can be up to five people automatically included in that review. The process applies the `Change sent to author` label. The process also skips the PR assignment if there are more than 50 articles in the PR. It assumes that the PR is a bulk update and doesn't need author review.

   > **NOTE**
   >
   > Any Microsoft employee or vendor with Read permissions can add more assignees by typing `#assign:<GitHubID>`, or reassign the PR and remove current assignees with `#reassign:<GitHubID>`.

5. An author of the PR adds the `#sign-off` comment, confirms they're the author of at least one file in the PR, and adds `ready-to-merge` label. If someone other than an author adds `#sign-off`, a comment is added: `@<submitter>: Only the author, @<author>, can sign off on this pull request`.

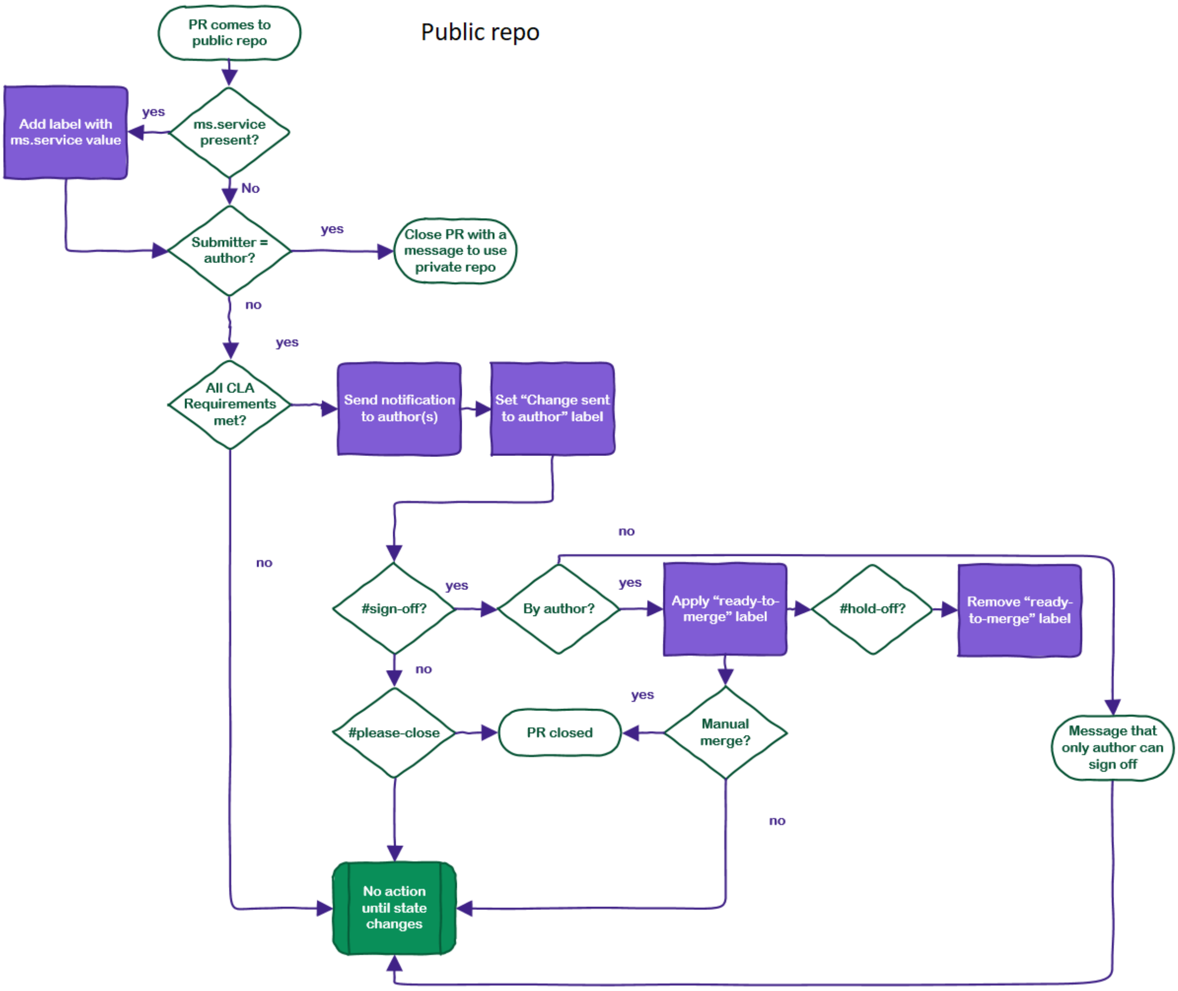
6. The PR isn't automatically merged. First, review the PR and then merge it manually with a request from the relevant product or content team member.

7. Cancel the `#sign-off` comment by adding `#hold-off` in a GitHub comment for the PR.

8. Any user can type `#please-close` to close the PR if the changes can't be merged.

9. Any user can type `#please-open` to reopen a closed PR.

Public workflow diagram



How do I turn on PRMerger?

Discuss your workflow with your team before turning on PRMerger. Then, decide how the PRMerger automation will be integrated. Some teams use the Content & Learning PR Review vendor team, while some teams triage and review their own PRs.

For more information, see How to turn on GitHub automation.

Release notes

June 6, 2019

- In both public and private repositories, PRMerger automatically assigns the pull request to the listed author. You'll always receive a GitHub notification when someone creates a pull request against an article that you own - in both public and private repos. This change addresses longstanding feedback from authoring teams.
- The `needs-human-review` or `qualifies-for-auto-merge` labels can be applied in any branch except for `live`. Previously, only `main` and `release` branches were covered. Now, PRMerger has full functionality in `Learn` repos where there's a new convention for release branch naming.
- PRMerger adds product, service, subproduct, and subservice labels to pull requests in the private repository workflow. The functionality is now present in both public and private workflows. The labels let you sort and

search PRs by service.

- PRMerger relied exclusively on in-article metadata for assignment and notifications. PRMerger now supports metadata assigned through the `docfx.json` file.

Pull request review process when PRMerger isn't available

7/7/2021 • 4 minutes to read

Many repositories don't have PRMerger webhooks enabled. This article describes the PR review process for those repositories. The PR review process was designed with the following goals in mind:

- Publish high-quality content from our team, product team members, and community members.
- Provide timely, actionable feedback to authors in a consistent manner.
- Enable discussion between authors and reviewers.

The processes continue to evolve as teams innovate and as the platform matures.

Reviewers

One of the content team members reviews every PR. Content team members may request a review from the specific product-team members to verify technical accuracy. The content team uses GitHub's Code Ownership feature to automatically request reviews from content team members. As part of this process, a reviewer may tag other team members to review internal PRs. Team members see requested reviews from team members and community members in the same queue.

Community members can review PRs and provide feedback as well. However, at least one member of the core content team must approve any PR before it's merged.

Review process

Reviews follow one of three paths based on the scope of the PR:

- **Small PRs** - Small PRs are a single bug fix: typos, broken links, small code changes, or similar changes.
- **Major contributions** - Major contributions are significant edits to an existing article, new articles, or edits to multiple articles.
- **Draft** - Authors can request an in-progress review by opening a draft PR in GitHub.

The processing used by the Contributor License Agreement (CLA) bot is a good guideline for the distinction between "small" and "large" contributions. PRs that don't require the CLA to be signed are likely "small." PRs that do require the CLA are likely "large."

Small PRs

The changes in small PRs are reviewed for accuracy. They're checked using the build on the review site. Because they're small, these PRs don't trigger a review of the entire article.

Reviewers may notice other small changes that would improve an article. If the changes are also small, the reviewer should suggest the changes as review comments. However, if the suggestions are large, the reviewer should open an issue and address it later.

Larger changes

Larger PRs undergo a more thorough review. The following are all checked:

- Sample code must be included in the PR, in source and as a downloadable zip file.
- Sample code compiles and runs correctly.
- The article clearly describes the goals for the reader, and the goals are met.

- The article meets style and grammar guidelines (see CONTRIBUTING.MD in the root folder of the repo.)
- All links should resolve correctly.

Content team members review the PR and submit the review with comments. If the reviewer requests only minor changes, team members may "approve" the PR with the feedback. The author can then address the feedback and merge the PR. Most reviewers request changes and when the author makes the changes, the reviewer reviews the updates again.

If the edits require a technical review, the content team reviewer requests a review from the appropriate product team member.

Review draft pull requests

New authors may want feedback earlier in the process than after the updates are finished. If they want an early review, the author can open a draft PR in GitHub. Then, the author can request an early review in a comment.

These early reviews aren't as thorough as a full PR review. The content team will comment, but won't "approve" or "request changes" using GitHub's review feature. These early reviews focus on the structure of the article: the outline, the overall content, and the samples. These reviews don't include a thorough check for grammar and correct links.

Explain suggestions

GitHub lets you enter comments in triple-back-tick blocks of type `suggestion`. The suggestions are displayed as a diff and can be merged by clicking a button. On short lines, GitHub does a good job of highlighting the changes. On longer lines, such as a long paragraph in one line of text, GitHub doesn't highlight the changes. When entering a suggestion for a long line, notice whether your changes are clearly highlighted. If the changes aren't highlighted, include comments outside the suggestion block explaining what you changed. Without an explanation, it's often time-consuming for other reviewers or the PR author to figure out what the changes are.

Respond to reviews

The author updates the PR to respond to comments and marks each addressed comment with the "+1" reaction. The author marks the conversation as "resolved" when they've addressed the review comments. If the author disagrees with the comment or addresses the comment in a different PR, the author adds a comment to explain the change. In those cases, the reviewer will mark the conversation as "resolved" when the comment has been addressed.

The author @-mentions the original reviewer in a comment to request a new review.

Response time expectations

Content team members typically review new PRs in under two business days. If it takes longer to review, one of the team members adds a comment to the PR and sets expectations.

Once a review has been submitted, authors should try to respond to comments in a week or less. Volunteers may not meet these expectations because of other commitments. In those cases, team members ask if the community author will update the PR. If not, the team member updates and merges the PR for them.

Contribute Azure CLI and Azure PowerShell sample scripts

6/24/2021 • 6 minutes to read

IMPORTANT

This article only applies to the `Azure-Samples/azure-cli-samples` and `Azure/azure-docs-powershell-samples` repositories.

Scripts for Azure command line tools (Azure CLI and Azure PowerShell) are a separate product from Azure documentation, and have their own repositories. Even when you're writing a script to be shown in an `ms.topic: sample` article, users may come to scripts through organic search or the Azure Samples portal and never see the documentation site. Because of this, scripts are reviewed separately from documentation PRs and have different requirements.

This document outlines the general steps to get your scripts ready for submission and what the pull request process is like for samples, including a breakdown of the review process.

Follow style guides

The first step for submitting to Azure tool repos is to make sure that your scripts meet the style conventions:

- Azure CLI style conventions
- Azure PowerShell style conventions
- Best practices for Azure CLI and Azure PowerShell scripts

Following the style guides is a good way to make sure that your pull request is processed as fast as possible. For further style guidance, review the style guide hierarchy article.

Keep in mind that no style guide is exhaustive. Reviewers are allowed to block a PR as long as there's a clear reason for requested changes. Part of the review process is encouraging a dialogue between the reviewer and the submitter to evolve these style guides and the sample scripts, so that they reflect the best of what we have to offer users.

Test script changes

Testing is the most important part of any code submission process. There's no automated testing available for the Azure tool samples at this time, so testing is up to you and the pull request reviewer. Azure CLI and Azure PowerShell have different testing requirements.

Make sure that all scripts in your PR are tested as described.

IMPORTANT

Make sure you **don't** test in an elevated session ('admin' on Windows, or through `sudo` or as `root` on Linux/macOS.) Testing a script as an elevated user can hide problems like writing to directories not accessible to a normal user.

When you can skip testing

You can skip testing if your PR has **only non-code changes**. Non-code changes include comments, whitespace, and files that aren't executable scripts such as `README` documentation. **All other changes require testing**, no matter how small they are.

Testing Azure CLI samples

When testing Azure CLI samples, **make sure** you have the latest version of the Azure CLI. It's recommended to test scripts on Azure Cloud Shell since it's an environment guaranteed to have the latest version.

Test on **at least one of** the following environments:

- Linux (any distribution)
- Azure Cloud Shell `bash`
- macOS (10.13 or later)
- Windows Subsystem for Linux (v1 or v2)

Testing is done by running the script directly. **Don't copy-paste the script.** Copy-paste can cover up some subtle formatting errors that affect execution. Start a shell session and test a script with:

```
bash ./script-name-here.sh
```

Your script should complete without unexpected errors when:

- Author-supplied values for variables aren't modified
- Correct user-supplied values are used where required

Testing Azure PowerShell samples

When testing Azure PowerShell samples, **make sure** you have the latest version of Azure PowerShell. It's recommended that you test on Azure Cloud Shell for your PowerShell 6.x platform.

Test on **all of** the following environments:

- PowerShell 5.1 (Windows-only)
- PowerShell 6.x (any supporting OS)
- PowerShell 7.x (any supporting OS)

Make sure that you record which version of PowerShell 6 you test with, and on which platform. Azure Cloud Shell is considered an OS/platform, for this purpose. This information is required for the pull request submission.

Testing is done by running the script directly. **Don't copy-paste the script.** Copy-paste can cover up some subtle formatting errors that affect execution. Start a PowerShell session and test a script with:

```
.\script-name-here.ps1
```

Your script should complete without unexpected errors when:

- Author-supplied values for variables aren't modified
- Correct user-supplied values are used where required

Create your pull request

Create your pull request against the appropriate repository. There's a pull request template that has the following requirements:

- A short description of your changes
- A checklist signing off on testing, security, and important script conventions

- A list of the tool versions and platforms you tested on

> **IMPORTANT**
>
> If your PR is blocking acceptance of changes to another repository, *make sure* you link to that pull request in your description to expedite PR processing.

The templates are hosted on GitHub:

- Azure CLI checklist
- Azure PowerShell checklist

All of these sections need to be filled out before pull request processing begins.

Designated reviewers

Azure PowerShell samples

- Submit PR to Azure/azure-docs-powershell-samples
- Reviewer is Mike Robbins @mikefrobbins

Azure CLI samples

- Submit PR to Azure-Samples/azure-cli-samples
- Reviewer is Delora Bradish @dbradish-microsoft

Pull request review and processing

Because of a lack of resources, there's no SLA for when PRs will be reviewed. To speed up pull request processing, make sure that you test *before* submission. Also complete the checklist and make sure it's accurate when creating your pull request. Most pull requests have review done within the first day of checklist completion.

Pull requests are tagged with GitHub labels to help make it easier to understand where in the process the pull request currently is.

Reviewer responsibilities

Reviewers should handle PR processing in a timely fashion. When reviewers leave comments on a pull request, they must:

- Be clear about what needs to be changed, with a rationale.
- Make it clear if a review comment references a **required** change, a **suggested** change, or an **upcoming requirement**.
- Make it clear if a review comment asking for clarification of a command/comment requires an answer before merge.

Reviewers **don't** review scripts for technical accuracy ("does what it claims.") That is the responsibility of the script author and their team - reviewers can't be experts on all Azure services. If a reviewer sees an obvious technical error they're still expected to call it out.

Reviewers are allowed to block PRs for anything they consider to be critical to change or clarify. If there's a dispute between the submitter and the reviewer, the sample repo owner is the final arbiter of whether a change is required or not.

Review steps

Review follows these steps:

1. **Submitter**: Fill out the `Description`, `Checklist`, and `Testing information` fields when you create your pull request. If your PR *only* consists of whitespace, comments, or non-script changes, you may replace the `Checklist` and `Testing` sections with `N/A`.
2. **Repo owner**: Assigns the PR to a reviewer if the checklist is complete. If the checklist isn't complete, the submitter is told what is missing and the PR is labeled as `incomplete-checklist`.
   - **Submitter**: When the checklist is completed, @ the repo owner in a comment with `#sign-off`, and goes back to step 2.
3. **Reviewer**: The reviewer starts review when they have availability, and labels the PR with `in-review`. The reviewer:
   a. Checks for compliance with the style guides, leaving comments for anything that needs to be changed. If there are no required changes, or the requested changes don't block testing, moves on to:
   b. Performs review. Reviewers check for style guide compliance, obvious errors, and use their best judgment to request changes. If there are no required changes, or the requested changes don't block testing, moves on to:
   c. Tests PR scripts. All scripts are tested as described in Test script changes. All platforms the submitter recorded in `Testing` are retested. For PowerShell, reviewers also test an OS that was **not** checked by the submitter.
4. **Reviewer**: If the PR requires changes, the `in-review` label is removed and the `requires-rework` label is added. The submitter is @ed in a comment.
5. **Submitter**: Address any issues, and then @ the reviewer in a comment with `#sign-off`. Review goes back to step 3.
6. **Reviewer**: Once there are no issues, all labels are removed and the PR is merged.

Pull request submission best practices

7/7/2021 • 3 minutes to read

To publish changes to content, you submit a pull request from your fork. Pull request submission and processing are covered in both the minor/infrequent changes and major/long-running changes workflows.

Pull requests that add new articles, new images, or that significantly update existing content are human-reviewed before being merged. Read this article to learn how to work with pull request reviewers. Also, learn how you can create pull requests that are easier and faster to review. The pull request queue works better for everyone when you create easier pull requests.

Work with pull request reviewers

Pull requests commonly go through a technical quality review or a content quality review, or both, to ensure complete quality.

Technical quality review

Some teams require a technical review of pull request content before it can move to the content quality review and merge stage. If your team implements a technical review, you'll likely receive an invitation and link to the PR to provide your feedback. The feedback mechanisms used to review and update the PR vary, depending on the tool that's used. Work with your technical review team for specifics.

Content quality review

Here are the basics you need to know about working with PR reviewers during the content quality review process:

- **Understand the role of the pull request reviewers.** The PR reviewers:
  - Ensure the basic quality of the content.
  - Prevent regressions in the repository.
  - Provide feedback before merging.

  Content quality reviewers are in a content governance role. The primary intent isn't to merge submitted content as quickly as possible. Expect feedback that will require you to make updates, especially for new and heavily revised articles. The content quality reviewers are also responsible for the final step of merging the pull request, after it passes all content quality criteria.

- **Plan ahead with the pull request reviewer team:**
  - For high-priority pull requests.
  - For pull requests for timed/dated releases.
  - For pull requests that change or add lots of files.
- **It may take a few days to have a pull request merged**

  The publishing process happens at least two times a day. It can still take some time to address blocking feedback from the PR reviewer team before your pull request is ready to merge.

Make the pull request queue work better for everyone

There are two basic realities in the PR queue:

- Pull requests that are small in scope and that contain similar changes take less time to review.

- Pull requests that are large in scope or that contain mixed kinds of changes take more time to review.

You can help make the pull request queue work better by following these best practices:

- Separate minor updates to existing articles from new articles or major rewrites. Work on these changes in separate working branches.
- When you delete articles or images, avoid mixing the deletions with content additions or updates. Handle the deletions in a separate PR that gets merged after the PR for related additions and changes. For example, update the redirects file and check that the redirect is working before deleting an article.
- If you're a Microsoft employee:
  - Contact the PR review team to speed up pull requests only when necessary. You can request sped up PR handling for Red Zone, privacy, legal, and security issues; for truly broken customer experiences; and for executive escalations.
  - For releases or refactoring of content, plan ahead with the publishing team to create a release branch. Coordinate merge-to-master times with the team so your content is published at the right time. See Use release branches for details.
  - If you're trying to coordinate out-of-band dates made by others with changes you're making, which are interdependent, you must coordinate that work ahead of time with the PR review team. Otherwise, you risk having many broken links.
  - For any significant and/or timed release, you can request a "pre-review" from the PR review team to make sure you have time to address any blocking issues.

In a hurry? Submit PRs that can be accepted automatically

If your repository supports it, use the PRMerger automation rules to get more of your day-to-day PRs merged automatically.

PRMerger can accept your PR automatically, if:

- It affects 10 files or fewer.
- It has 15 commits or fewer.
- Less than 20 percent of text changes.
- Selectors/switchers aren't updated.
- No files are deleted or added.
- No images are new, changed, or deleted.

Need to make many little changes?

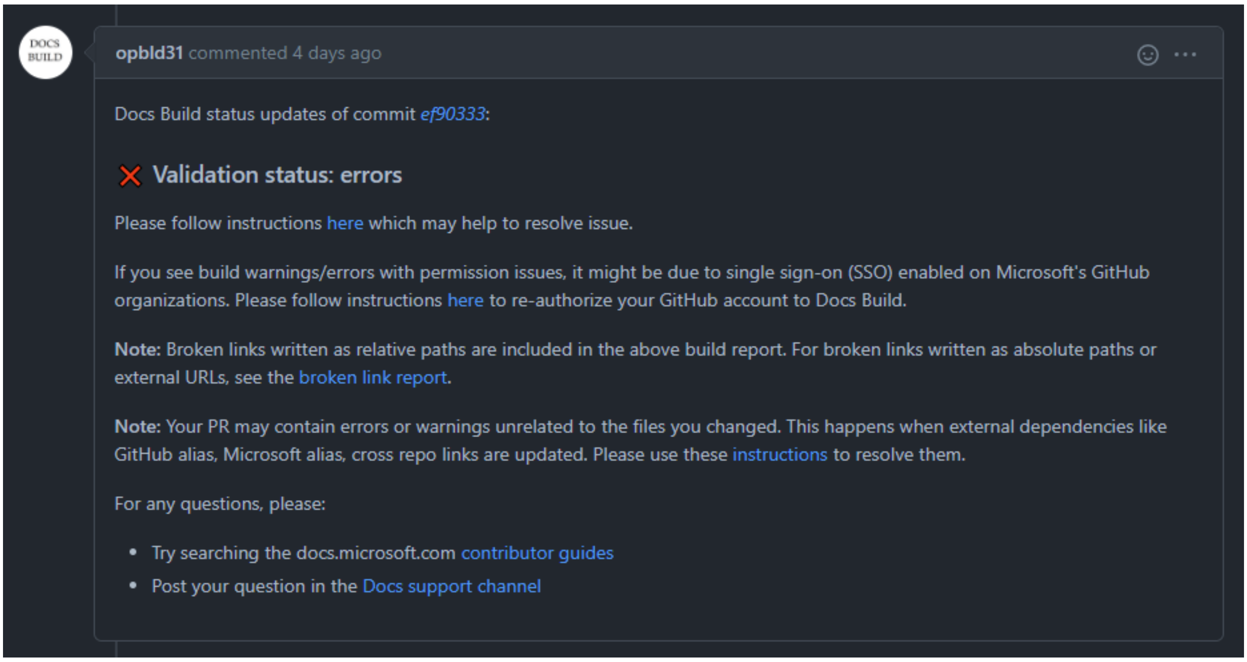
Take your cue from the preceding PRMerger automation rules, and follow these steps:

- Submit articles with light changes together in a PR with 10 or fewer files.
- Create a separate PR for articles in which images or selectors change. These changes require human review.
- Create a separate PR for new or deleted articles. These changes require human review.

Resolve a stuck pull request

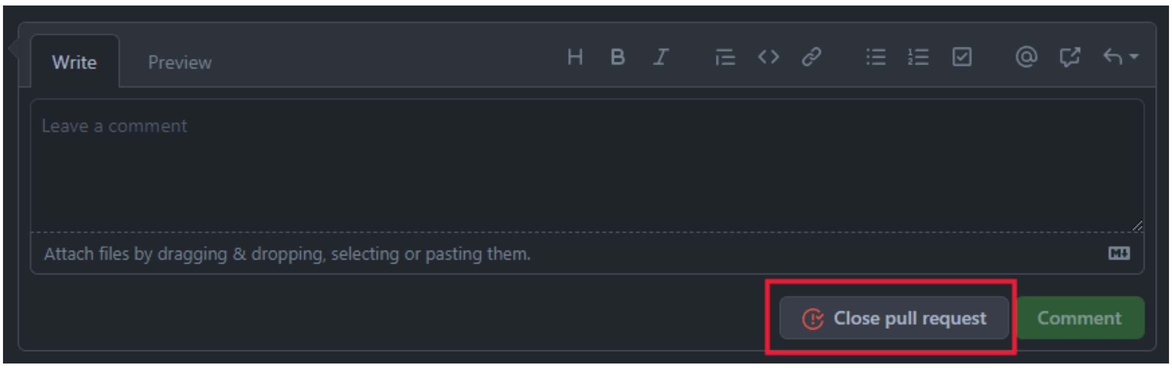
6/24/2021 • 2 minutes to read

Occasionally, your pull request (PR) may appear stuck or frozen and can't be merged. When this happens, the Docs Build bot sends a generic error message that looks like this:



When you get this generic error, try these steps to resolve the issue:

1. Refresh the GitHub page in your browser.
2. Check the Docs portal for queue or performance issues. For example, check if builds are triggering in the repository as expected. Use your GitHub credentials to sign in if necessary.
3. Close and reopen the PR to try to trigger a build. At the bottom of the GitHub page for your PR, select **Close pull request**. Wait for GitHub to close the PR, and then select the button again after its label toggles to **Reopen pull request**.



4. Check GitHub's status.
5. Check Azure's status.

If none of these steps resolves the build error, post to the Docs Support channel for help.

Resolve merge conflicts in Git and GitHub

7/8/2021 • 11 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.

What is a merge conflict?

A merge conflict sounds like a bad thing, but it really is a good thing - merge conflicts are part of the collaboration process.

Merge conflicts can occur in multiple situations, but boil down to the same thing. Different users have submitted changes that Git can't reconcile automatically. It requires human judgment to determine which is the right change. Or, human judgment may determine that the author has to rewrite the two changes to integrate both updates into the article.

Git marks the merge conflicts in the source using conflict markers. Delete the markers and edit the text to be what you want. Sometimes, you pick one option in the conflict, in other cases, you have to delete both and rewrite the final text you want in the article. Here's an example of what the markers look like:

```
<<<<<<HEAD

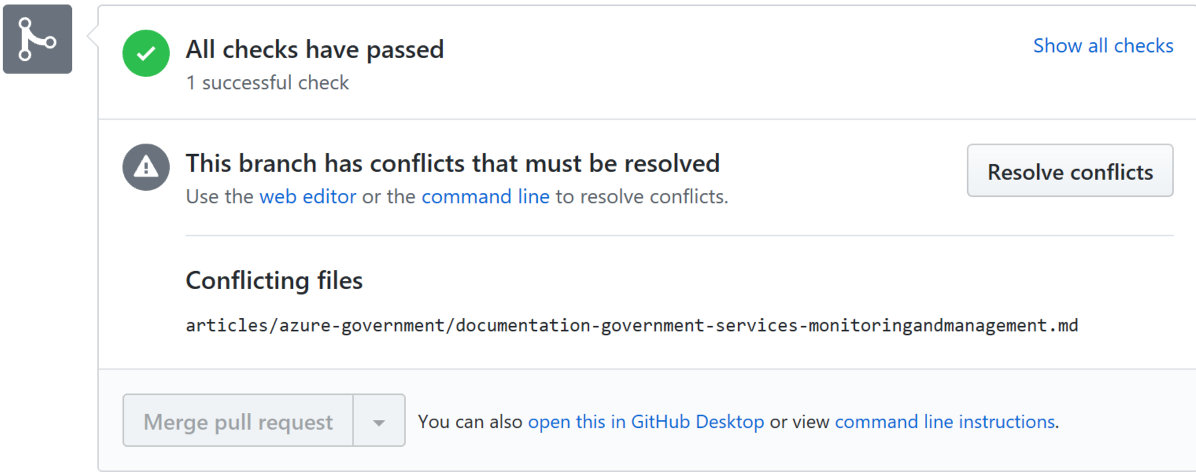
==========

>>>>>>GUID-numbers-and-letters
```

This article addresses the common merge conflict scenarios and provides tips, tools, and warnings.

Caution using the GitHub UI to resolve merge conflicts

If you have write permissions in your repo, you'll have access to the GitHub merge conflict resolution tool. **Resolve conflicts** appears in pull requests when a merge conflict is present:



The only time it's safe to use this tool is if the pull request is against the `master` branch.

For a six-minute video about how to resolve a merge conflict in the GitHub UI, see: Resolve a merge conflict.

> **WARNING**
> **Don't use the GitHub UI to resolve merge conflicts in pull requests filed against release branches.**
>
> When you commit the merge conflict fixes, GitHub automatically commits the contents of the release branch against master in a direct commit. This means you are unintentionally merging the release branch to master, causing your content to go live before you meant to.

Scenario 1 - Your pull request has a merge conflict

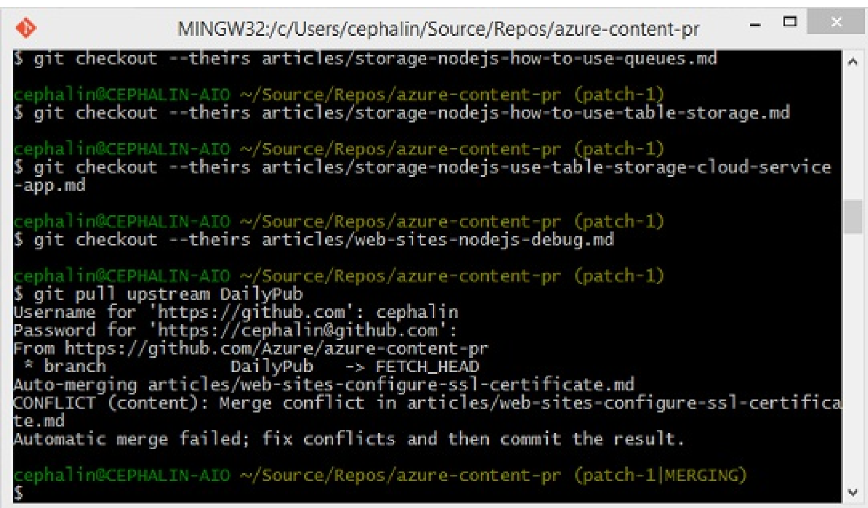
As a first step, make sure your local branch is in sync with the upstream master branch.

1. In Git Bash, make sure you're working in the local branch that is causing the conflict in the pull request. For example, if you're trying to merge content from your fork's 'snoopy' branch to the main repo's 'master' branch, make sure you're active in your local 'snoopy' branch.

2. Run our favorite command:

```
git pull upstream master
```

3. The prompt will probably change to include `| MERGING`, indicating you have to manually resolve the merge conflict. The command-line output should list the specific files that have a conflict:



If you can't tell which files have a conflict, run this command to list the conflicted files:

```
git ls-files -u | sort -u
```

4. Open the conflicted file in a Markdown editor.

5. Find the merge conflict markers. Delete the text you don't want to keep. Then delete the merge conflict markers, including the beginning marker, the divider, and the end marker.

   If you forget to delete them, they'll publish in the article. The only text left from the conflict should be the correct text you want to show in the published article.

6. Save, add, commit, and push to your fork. Create a pull request to the appropriate branch in the upstream repository.

Sometimes, it will look like nothing happened - the prompt doesn't change. If it looks like nothing happened, just run the standard commands to add, commit, and push. Running the commands should unblock the pull request so it can be merged.

Scenario 2: You're trying to resolve a merge conflict in another person's pull request

There are many ways to accomplish the same task in Git. Here's two methods for pulling the contents of another person's pull request to your computer so you can resolve the merge conflicts. These methods are good to know because you can follow the same steps to pull another person's changes from a PR to your local computer.

Method 1 - Pull their branch

1. Create a new local branch from master, just like you do for normal, daily work, then check it out and let your fork know it exists:

```
git pull upstream master
git checkout -b <branch name>
git push origin <branch name>
```

2. Pull the other contributor's pending changes from their fork and working branch into your fork and working branch. When you enter the command, get the capitalization right - Git is case-sensitive:

```
git pull https://github.com/<contributors-github-account>/azure-docs-pr.git  <contributors-working-
branch>
```

3. If there's a merge conflict, open the conflicted files, resolve the merge conflict, save, add, commit, and push.

Method 2 - Fetch the upstream references

1. Locate the PR number you want. GitHub list the PR number prominently in the pull request. It's also part of each pull request's URL: https://github.com/Azure/azure-content-pr/pull/25614.

2. Fetch that specific pull request and put that PR into a new local working branch by running these two commands:

```
git fetch upstream refs/pull/25614/head
git checkout -b 25614 FETCH_HEAD
```

3. Run `git pull upstream master` now.

4. List the files that have conflicts:

```
git ls-files -u | sort -u
```

5. Resolve the conflicts, then add, commit, and push to YOUR remote origin (your fork). In this example, the command created a working branch named 25614, so the push command would be:

```
git push origin 25614
```

Scenario 3: You're resolving a merge conflict in your upstream release branch (authors)

In repositories where the Content & Learning Content Production Service (CPS) team provides publishing support, the team runs a script to update all the existing release branches with the latest content from the master branch. They run this script after each publishing run. Multiple times a week, they come across merge conflicts when they run the script. The merge conflicts prevent them from updating the affected release branch.

> **WARNING**
>
> **Do not use the GitHub UI to resolve merge conflicts in pull requests filed against release branches.**
>
> When you commit the merge conflict fixes, GitHub automatically commits the contents of the release branch against master in a direct commit. This means you are unintentionally merging the release branch to master, causing your content to go live before you meant to.

Historically, the CPS team resolved the merge conflicts to help the process along. However, the team lacks the domain and release-specific knowledge to accurately resolve the conflicts. To ensure correct conflict resolution, the writer or PM responsible for the content in the release branch needs to resolve these merge conflicts. The CPS team will contact you to let you know a merge conflict exists in your release branch. Once they do, here's how to resolve these sorts of merge conflicts:

1. On your computer, open Git Bash or Visual Studio Code. Then, change to the affected repository using the command line:

```
cd <repo name>
```

2. Check out your local copy of the upstream release branch:

```
git checkout <release branch name>
```

   If you don't have a copy of the release branch in your local clone, follow the steps in the Use release branches article to create it.

3. Update the local release branch with the latest upstream content:

```
git pull upstream <release branch name>
```

4. Pull the master branch into your local release branch:

```
git pull upstream master
```

At this point, all the changes from master will come into your local release branch. The output of the command will list the files that contain conflicts.

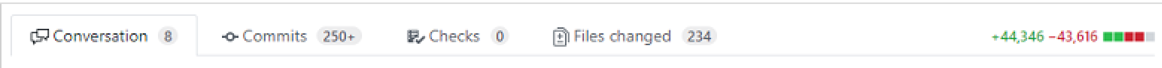
5. In Visual Studio Code, open each conflicted file, resolve the conflict, and save your changes.

   Sometimes, the files that show conflicts aren't files that are part of your content set. And, you didn't intend to change them in your release branch. If this problem happens, you need to check out the versions of those files that are in the master branch to restore them to the correct version. Run this command to restore the files to match the master branch:

```
git checkout upstream/master articles/<service>/<filename>.md
```

6. At the command line, add and commit the changes. Then, push the changes to your fork.

7. Compare your branch with upstream master. This step helps to validate the changes you expect to see. When you look at your PR, GitHub is likely to display hundreds of commits and file changes. It makes it hard to learn what actually changed.



   To see the diff of your branch and upstream master, use these commands:

```
git fetch upstream
git diff upstream/master --name-status
```

8. In the output of the `git diff` command, Git Bash displays the type of change for each line like this:

   - A: addition of a file
   - C: copy of a file into a new one
   - D: deletion of a file
   - M: modification of the contents or mode of a file
   - R: renaming of a file
   - T: change in the type of the file

   The following condensed example output shows two *modified* files, two *deleted* files, and one *addition* of a file:

```
$ git diff upstream/master --name-status
M .openpublishing.redirection.json
M articles/active-directory-domain-services/TOC.yml
D articles/active-directory-domain-services/active-directory-ds-admin-guide-configure-secure-ldap-enable-ldaps.md
A articles/active-directory-domain-services/breadcrumb/TOC.yml
A articles/active-directory-domain-services/compare-identity-solutions.md
[...]
```

   Review the list of changes to upstream master. The only changes listed should be files that you, or others in your team, have worked on as part of your release branch. If there are other file changes, it's an indication you may have pulled across some other changes by accident.

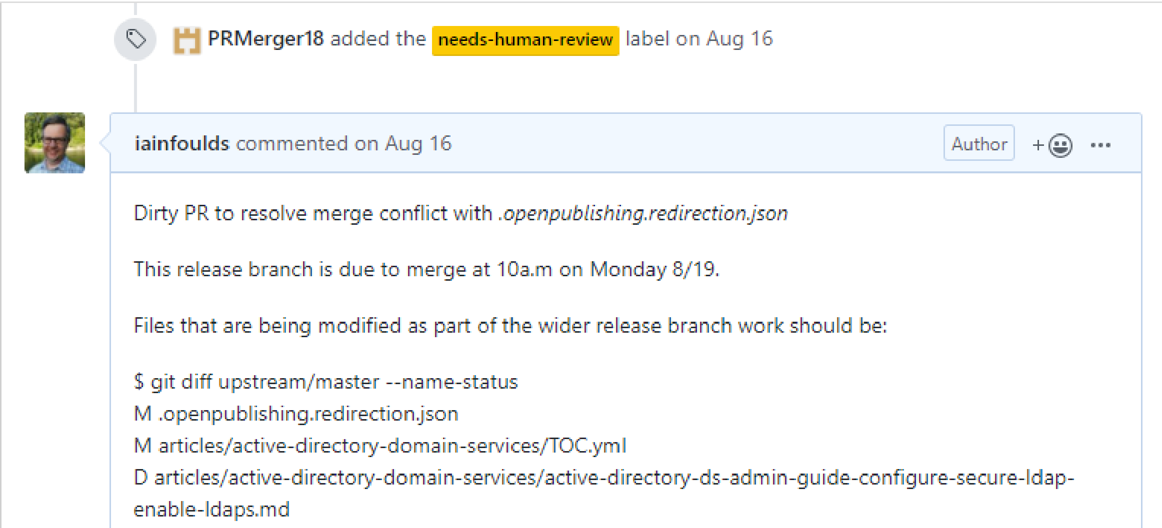
9. In GitHub, create a pull request *from* your forked copy of the release branch *to* the upstream copy of the release branch. Make sure you clearly title and describe the pull request so your repo admin knows what you intend the pull request to do. For example, *[ServiceName] Dirty PR for release branch merge conflict with upstream master*.

This pull request contains commits and changes from other authors - they represent all the changes in master that you're bringing into the release branch via your fork. This type of pull request is known as a *dirty PR*. They aren't optimal, but in this case they're sometimes unavoidable.

10. After the PR passes build validation, check the staged articles in which you resolved conflicts. Make sure that you correctly handled the conflicts and the articles are rendering without any remaining merge conflict markers.

11. Sign-off on the pull request per the normal process.

    To help the repo admin or PR reviewer learn about the scope of your changes, include the output of your `git diff` commands and list of changes. If you've deleted or modified files outside of the core working area, add that note in the sign-off comments.



The repo admin or PR reviewer should carry out similar steps, shown in the next section, and the list of changes should align.

PR review team instructions for reviewing the PR created in Scenario 3

The steps in the previous section create a "dirty PR" that lists all the commits from master that haven't been merged to the release branch because of the merge conflict. PR reviewers need to follow these steps to ensure the pull request is safe to merge.

1. On your computer, open Git Bash and change to the repository the pull request is in.

2. Add the author's fork of the repository as a remote. This process relies on your GitHub personal access tokens:

```
git remote add <username> https://<your-user-name>:<your-access-token>@github.com/<username>/azure-docs-pr.git
git fetch <username>
git fetch upstream
git diff upstream/master...<username>/<branch-name> --name-status
```

   In the output, Git Bash codes the type of change for each line:

   - A: addition of a file
   - C: copy of a file into a new one
   - D: deletion of a file
   - M: modification of the contents or mode of a file

- R: renaming of a file
- T: change in the type of the file
- U: file is unmerged (you must complete the merge before it can be committed)
- X: "unknown" change type (most probably a bug, please report it)

As a PR reviewer, you're looking for anomalies to what seems to be the pattern in the pull request. For the dirty PRs that bring release branches up to date with master and fix merge conflicts between master and the release branch, look at these items:

- What is the release branch name? It usually indicates a service area, and the changed files typically will clearly map to the service area. If there's a large variety in the services affected, check the pull request for information and clues. If it's a release branch where branding updates are being made, you might see broad changes across services. For new feature releases, scope would normally be tight so you wouldn't expect to see broad changes across services. Ask the author if in doubt.

- In most cases, for a release branch, you're expecting mostly new and updated files. If you see deletions, ask the author to confirm that the deleted files for the release branch are intentional deletions.

- If there are no changes listed, something odd is going on. Most likely, the author made a mistake and resolved the merge conflict by making the release branch match the master branch. It's likely a mistake because it's eliminating any differences between the release branch and master. Check with the author to check if they intended to essentially wipe out differences in the release branch compared to master.

You're trying to resolve merge conflict with no change between branches

Sometimes contributors with write permissions to a repo will see another case of merge conflict with **no change**. In this special case, the target branch already includes everything in source branch and usually happens during merge of master into live branch. It occurs because of the below reasons:

- Master branch doesn't have a common history with live branch.
- One contributor created a PR from master to live branch. But another contributor already pushed the same change directly on live branch.

To resolve merge conflict on PR with **no changes** between master and live branch, just go ahead and merge the PR into live branch.

Use Visual Studio Code to resolve merge conflicts

Visual Studio Code includes a lightweight merge conflict resolution UI - once the conflict is present on your computer locally, just open the conflicted article. See the VSCode documentation for a screenshot and description of the merge conflict resolution UI.

Use Visual Studio to resolve merge conflicts

If you're running VS, you can resolve merge conflicts in a visual editor. See the Visual Studio documentation. You probably only want to resolve merge conflicts this way if you use VS for other work.

Break up a single large pull request into smaller PRs

7/7/2021 • 5 minutes to read

One cause of long turnaround times for pull request reviews and associated publishing delays is when a single pull request changes more than 100 files. These PRs require repository admin review before merging. Reviewing large PRs is also difficult to do in GitHub's web interface. PR reviewers often reject these PRs because all the requested changes cannot be reviewed.

This article provides proactive guidance on how to prevent these types of PRs. It also includes a tutorial on how to break down a PR with numerous changed files into smaller, reviewable chunks.

Avoid branches that update large numbers of files

Bulk text or link changes drive PRs with large numbers of changed files. Bulk find and replace actions can be dangerous. It's best if you carry out these changes and PR them per directory in your content repo so you can review every change before merging.

To avoid large PRs when teams are working together on a content release, use the release branch process and merge changes into the release branch. The team reviews the changes in the context of the overall shared effort as the work progresses instead of all at once when merging to the master publishing branch.

Fix a branch that updates a large number of files

To fix a pull request with many modified files, you'll need to be comfortable with using the Git command line and familiar with basic Git concepts. The goal is to take your branch with numerous changes and create a set of smaller branches that collectively have all of the changes of the original branch. You then open pull requests for each of those branches.

You can also use this process if you pushed one or more files to the wrong branch, or if you just want to clean up a PR by restricting it to files that have related changes.

> **NOTE**
>
> This process will not retain the original commit history of your changes once merged into the master publishing branch in your repo.

Prepare your local repo

Run the following commands to create a state in your repo where the changes in the PR are unstaged. These steps assume that `upstream` is the publishing repo and `origin` is your fork of the publishing repo.

```
git checkout master
git fetch upstream # this may be necessary (depending on your git config) to receive updates on
upstream/master
git pull upstream master

# Merge the feature branch into the master branch.
git merge <pull_request_branch>

# Reset the local repo to the current head of the publishing repo, leaving behind changed files as unstaged.
git reset upstream/master
```

If you run `git status` at this point, you'll be on the master branch in your local repo with a list of unstaged file changes. These file changes are the diffs between the pull request branch and `master`.

Group the unstaged files into multiple, smaller branches

1. Create a new branch off `master`:

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

   > **TIP**
   >
   > If you're breaking up a large PR into more manageable chunks, use a similar branch name with an incrementing counter at the end to name the branches.

   You're now in the new branch, and the unstaged files have come along for the ride.

   ```
   ~/visualstudio-docs-pr (master)
   $ git checkout -b regroup-1
   Switched to a new branch 'regroup-1'

   ~/visualstudio-docs-pr (regroup-1)
   $ git status
   On branch regroup-1
   Untracked files:
     (use "git add <file>..." to include in what will be committed)

        file1.md
        file2.md
        file3.md
        file4.md
        file5.md
        file6.md

   nothing added to commit but untracked files present (use "git add" to track)
   ```

2. Add, or stage, a subset of the files using `git add` or your favorite graphical Git tool (like GitHub desktop or Visual Studio Code).

   ```
   ~/visualstudio-docs-pr (regroup-1)
   $ git add file1.md

   ~/visualstudio-docs-pr (regroup-1)
   $ git add file2.md

   ~/visualstudio-docs-pr (regroup-1)
   $ git status
   On branch regroup-1
   Changes to be committed:
     (use "git reset HEAD <file>..." to unstage)

        new file:   file1.md
        new file:   file2.md

   Untracked files:
     (use "git add <file>..." to include in what will be committed)

        file3.md
        file4.md
        file5.md
        file6.md
   ```

3. Commit the staged files:

```
git commit -m "<your commit message>"
```

4. Push the branch to `origin`:

```
git push origin <branch name>
```

5. Now, switch back to `master` branch:

```
git checkout master
```

The unstaged files will come along for the ride.

```
~/visualstudio-docs-pr (regroup-1)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1326 commits.
  (use "git push" to publish your local commits)

~/visualstudio-docs-pr (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1326 commits.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)

     file3.md
     file4.md
     file5.md
     file6.md
```

6. Repeat steps 1-5 until you've staged and committed all of the unstaged files.

Create a pull request for each of the pushed branches

Create a pull request for each of the pushed branches on github.com. The sum of the changes in the new pull requests is equivalent to the file changes in the original (large) pull request, but since the changes are smaller it's more easily reviewed.

If you're submitting multiple pull requests for new articles, you can request an exception to the PR review criteria. The criteria requires every new article to have a TOC entry and submit the TOC in the last and final pull request. Email the C+L Pull Request Review Team team to request the exception. This exception applies only when someone's asked you to break up a large PR. Normally, you'd work iteratively and submit each new article with a new TOC entry. Each exempted PR must contain a link to the PR that contains the TOC updates for all the PRs.

Working with someone else's pull request

If you're working with a pull request opened from another user's fork, you first need to pull down their branch to your local repo. See Checking out pull requests locally for instructions on how to take another user's pull request and bring it down to a local branch. From there, you can follow the steps above in your local repo.

Optimize for the pull request review process

4/16/2021 • 2 minutes to read

When you're ready to publish new modules or updates to existing modules, you'll submit a pull request (PR) to your fork. Most PRs in Learn repositories (repos) are human-reviewed before they are merged. This article describes how to work with the PR review team techdocprs@microsoft.com and create publish-ready modules more efficiently.

Work with PR reviewers

PRs typically go through a technical quality review or a content quality review, or both, to make sure quality standards are met. Here are the basics you need to know about working with the PR review team during the content quality review process:

- **Understand the role of the PR review team**
  - Ensure Microsoft leadership team content quality requirements are met
  - Prevent regressions in the repo
  - Provide feedback before merging
  - Maintain the integrity and functionality of the repo

  The PR review team's role is to make sure your content meets the quality requirements defined for Learn content, not to merge as quickly as possible for the next publishing cycle. Expect that your PR will have review feedback that will require you to spend time making updates to satisfy these quality requirements —and plan with this possibility in mind. The PR review team is also responsible for the final step of merging your PR, after it passes all content quality criteria.

- **PRs may take a few days to be merged**
  - Although content is published live at least twice daily, you might need lots of time to resolve PR review feedback before your PR is ready to be merged
- **Plan ahead with the PR review team**
  - For high-priority content
  - For timed/dated releases of your modules and learning paths
  - For PRs that contain multiple modules

Make the PR queue work better for everyone

There are two basic realities in the PR queue:

- PRs that are small in scope and contain similar changes take less time to review.
- PRs that are large in scope or contain mixed kinds of changes take more time to review.

Follow these best practices:

- Contact the PR review team to expedite PRs only when absolutely necessary. You can request expedited PR handling for Red Zone, privacy, legal, and security issues; for truly broken customer experiences; and for executive escalations.
- For timed releases, plan ahead with the Content Publishing Service (CPS) team to coordinate out-of-band publishing.
- For any significant and/or timed release, request a pre-review from the PR review team to make sure you

have time to resolve any blocking PR review feedback.

- For new learning paths, provide a list of the PRs that merged individual modules to the release branch. Scanning those PRs will help the PR reviewer complete the review much faster, as issues should've already been identified and addressed as part of the PR review process.

References

- Prepare to write a Learn module
- Content requirements for Microsoft Learn
- Authoring guidelines
- Pre-publication checklist
- Formal editorial review for Microsoft Learn content
- Pull request review checklist

How to work with public pull requests

6/24/2021 • 9 minutes to read

Most content sets have two repositories - a public repo provided mainly for external community contributions, and a private repo for contributions from Microsoft contributors.

Employees working in the public repo

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. Or, work in the private repo, but request that the author review your changes using an @mention.

Workflow overview

1. The customer files a pull request in the public repo against one or more articles.
2. The contributor license agreement (CLA) automation runs and determines whether a CLA needs to be signed.
3. After the CLA requirements are met, the author listed in the metadata is notified of the pull request.
4. The author reviews the PR, thanks the contributor, and decides what the next step is:

   > **IMPORTANT**
   >
   > Reminder, changing steps in the articles may impact dependent repositories such as `Azure-Samples`. Please ensure to step through the edited article to ensure that everything is functioning as it will be once the contributions are merged.

   - The changes are correct and relevant as-is. The author lets the PR reviewer for the repo know to merge the pull request.
   - Changes are needed before the PR can be merged. The author works with the contributor to get the PR to the point where it can be merged and then lets the PR reviewer for the repo know to merge the pull request.
   - The author determines that the changes cannot be merged. The author lets the contributor know why, then asks the PR reviewer for the repo to close the pull request.
5. Where there are private/public repository pairs, the public repo and private repo are merged together as part of the publishing process.

Automation to support this workflow

Content & Learning has automation to support the workflow described in the previous section. After the automation is set up, the automation supports the process as follows:

1. When a PR comes in, the automation checks the GitHub account of the contributor to determine whether the listed author is the person submitting the change. The automation checks against the value entered in the **author** metadata attribute. If the listed author submits the changes, the PR is closed with a message that the author should use the private repo. If the contributor is not the listed author, the automation goes

to the next step.

2. The automation checks to see if the CLA requirements are met. If a CLA must be signed, the PR sits until the CLA is signed. If the changes are small and do not require a CLA, or if the CLA has already been signed, the automation goes to the next step.

3. The automation writes a comment to the pull request to acknowledge the contribution. This comment causes a GitHub notification to be sent to the person whose GitHub account is listed in the author metadata attribute. An additional notification and instructions email are also sent to the alias listed in the ms.author metadata attribute. Finally, the automation applies the `Change sent to author` label.

4. After review, these comments are used to drive the next action:

   - Type `#sign-off` to assign the ready-to-merge label (listed author only).
   - Type `#please-close` to automatically close the pull request (anyone).
   - Type `#sign-off` can be canceled with `#hold-off` (anyone).

   The comments run off a timed job, so there may be up to a 30-minute delay. If someone other than an author adds `#sign-off`, that person receives a message indicating only the listed author can sign off. If you are a subject matter expert qualified to evaluate the change, you can use the exception process to sign off on the pull request - send an email to the techdocprs alias and request it be merged. Let the reviewers know you are on the product team or writing team for the product or service.

5. The pull request reviewer for the repo queries for PRs that have the `ready-to-merge` label and merges them.

Pull public commits to the private repo for updates, staging, publication

This section of guidance applies in cases where your repo has both a public and private repo, and your primary workflow is in the private repo. This is currently the case for most Content & Learning teams.

Sometimes, you need to modify changes submitted by a community member, or you need to pull new articles and major rewrites submitted through a public repo to the private repo for fixing, staging, Acrolinx, and merge.

Build validation is not turned on in public repos at this time, and Acrolinx is not turned on. Neither tool works in public repos in a way suitable for customers. The majority of changes have typically been small in scope. To help ensure quality, when a new article or major rewrite is submitted through a public repo, you should pull the changes to your private repo and run it through the standard process of validation, staging, Acrolinx, and PR review.

You can use these basic steps to pull changes from a public to a private repo:

1. In your local private repo, create a new local branch from master, just like you do for normal, daily work, then check it out and let your fork know it exists:

   ```
   git pull upstream master
   git checkout -b <branch name>
   git push origin <branch name>
   ```

2. Pull the other contributor's pending changes from their fork and working branch into your fork and working branch. When you type the command, get the capitalization right - git is case-sensitive:

   ```
   git pull https://github.com/<contributors-github-account>/<public-repo-name>.git <contributors-working-branch>
   ```

3. Push to your fork again, and create a pull request to master. Once you have a successful build, review the staged content, make updates required to meet the Acrolinx min score, and sign off when you are ready. The PR review team will provide feedback if the PR review criteria require updates.

In case the other contributor plans to add more content before submitting a pull request to master, you can submit a PR to the same branch of contributor's fork instead of upstream. This way you can stack commits from different authors into a single branch before submitting a PR to master. When submitting a PR to the contributor's fork, it may not be available in the drop-down by default. However, you can submit the PR by updating the URL of the page. Here is the URL format to submit a PR from your fork to someone else's fork:

```
https://github.com/<contributors-github-account>/<public-repo-name>/compare/<contributors-branch-name>...
<Your-github-account>:<Your-branch-name>
```

There are plans under discussion around how to make the public and private workflows the same, provide the same quality support in both public and private, and make the workflow account for trusted and untrusted contributors. This will help ensure that contributions from external sources and internal sources get the right level of review and approval. However, we're not there yet.

Guidance for new articles contributed by external community

Process

For Azure and other product areas where pull request review is provided by the Aquent vendor teams, the vendors will contact the business approver for the product or service when a new article or significant rewrite is submitted in a public repository.

The FTE content team must work with the external contributor to determine how and whether to proceed. If the decision is made to publish the article, the FTE works with the external contributor to get the article as complete as possible in the public repo. Then, the FTE must pull the commits to the private repository and run the new article through build validation, staging, and Acrolinx. The new article must be merged via the private repo. After the commits are merged to live for publishing, the public pull request will be automatically closed.

Metadata and ownership of articles contributed by external community

There are two metadata attributes that describe who is accountable for an article. Many downstream processes and some automation are dependent on these values:

- `author`: Lists one value, the GitHub account of the author.

  If your repo is covered by current public repo PR automation, this person can sign off on pull requests against the article in the public repo, even if that person is not an employee. (This is a known issue.)

- `ms.author`: Email alias for the author; can accept multiple values.

To ensure that a Microsoft FTE signs off on all updates to the article, an FTE must be listed in the metadata as the author:

- `author`: The GitHub account of the FTE sponsor for the article.
- `ms.author`: The email alias of the FTE sponsor, and the full email address of the community author.

While not optimal, this will ensure an employee reviews any updates to the article that come in on the public side. The community contributor is recognized through their commit history to the article.

Customer etiquette

Be nice to customers who make contributions - use friendly language and remember they have taken time from of their day to contribute to our content! If you have to close a PR, let the contributor know why.

Finding open pull requests

To find open pull requests for your articles, use one of the following methods:

- Where the automation is running:
  - Watch your email for notifications and respond promptly.
  - In the Azure-docs-pr repo, you can filter the PR queue by service. For example, this PR query finds the open PRs for Azure Stack: `is:pr is:open label:azure-stack`. The service label is based on your service metadata.

SLA for responses to public PRs

- PRs that are 1-7 days old are green
- PRs that are 8-14 days old are yellow
- PRs that are 15 days old or more are red

Contributor license agreement (CLA)

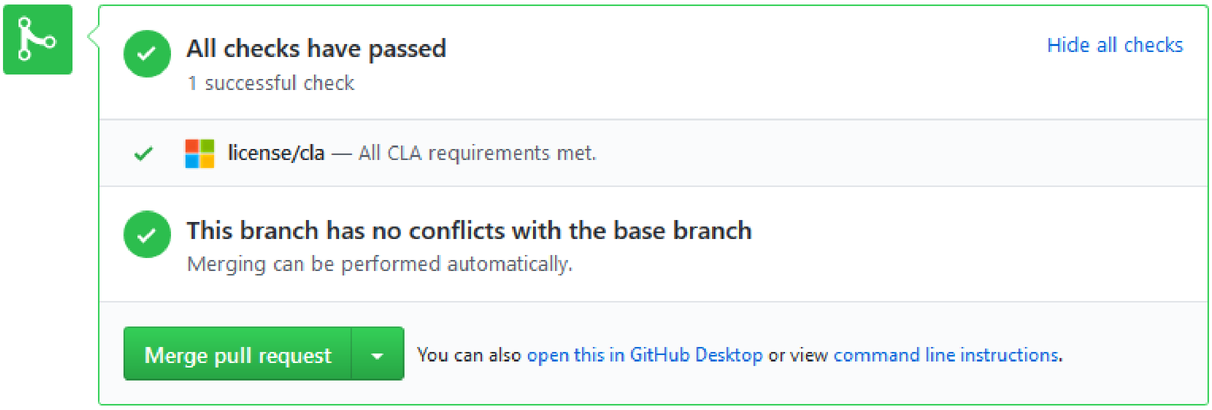
It's crucial that we work correctly with pull requests where a CLA must be signed. Here's the basics:

- Authors must not look at the content of a public PR if a CLA is required and not yet signed due to the legal issues around IP ownership.
- The CLA basically says that the contributor releases ownership of whatever IP the contributor is providing.
- A CLA must be signed when the contribution exceeds the CLA limit. It's about the size/scope of the change.

Here are some screenshots of how the new CLA service makes itself visible in pull requests.

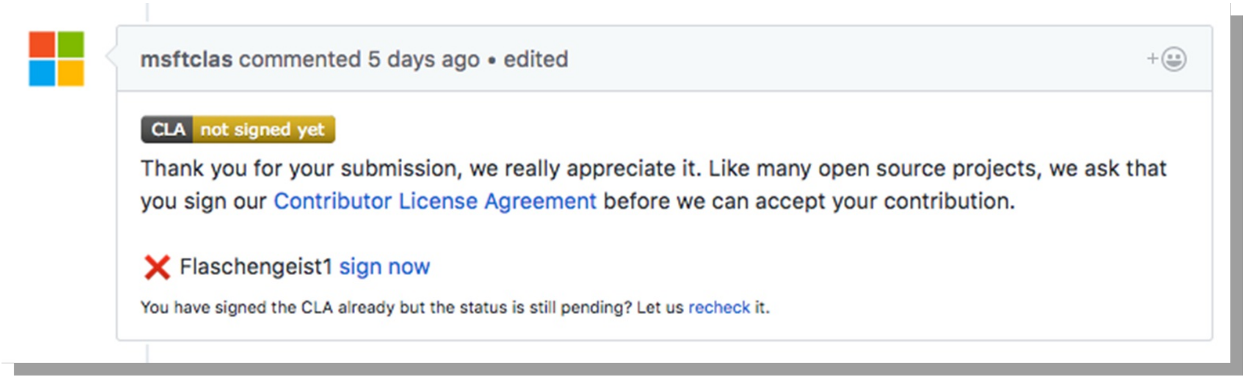
No CLA is required

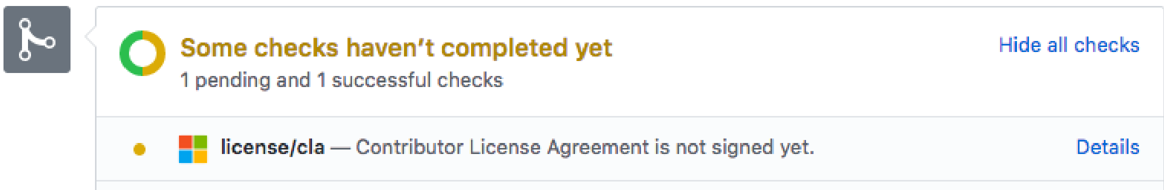
Only a status is provided. This applies when no CLA is required, and it applies when a CLA is required but the contributor already signed the CLA in an earlier PR. Microsoft employees whose identities in GitHub are linked to their MSFT identity also automatically pass the CLA requirement.



CLA is required and not signed yet

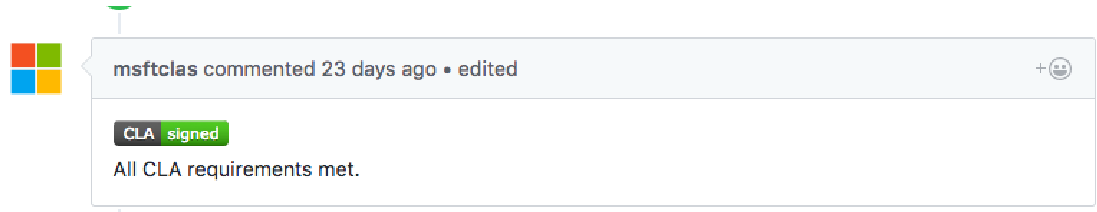
A comment and status are provided.

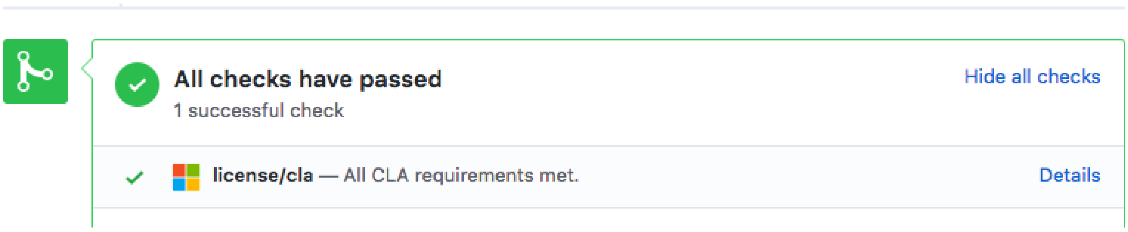




CLA is required and signed by submitter

A comment and a status are provided – this happens only in the PR in which they had to sign the CLA the first time.





Regular, out-of-band, and hotfix publishing

7/7/2021 • 8 minutes to read

This article is for repo admins, business approvers, and the Content & Learning Content Production team. It defines the three types of publishing, and it explains how to perform hotfix publishing. These tasks require write permissions or admin permissions in the repo you're working in. You must also have access to merge to the live branch if the live branch is protected in the repo.

Publishing definitions

Regular publishing: The established daily publishing cycle that happens every work day at approximately the same time for your repo. Here's how the regular publishing process works:

- For a public-only repo, merging the master branch to the live branch automatically triggers a publishing run to docs.microsoft.com.
- Where a public and private repo pairing exists, the private master branch is merged to the public master branch. Then, the public master branch is merged to the private master branch. The final step is to merge the private master branch to the private live branch, which automatically triggers the publishing run to docs.microsoft.com.

Out-of-band publishing: Like regular publishing, out-of-band (OOB) publishing is accomplished by merging the master branch to the live branch. However, an OOB publish is an extra planned publishing that is run outside the standard times. An OOB publish includes all the contents of the master branch. OOB publishing should be pre-planned, pre-announced, and infrequent. The actual publishing process includes all the elements of regular publishing, just at a time and day that is different from the normal schedule.

Early morning publishing runs for large events like Build and Ignite are considered OOB. Why? This is because they happen much earlier in the day than regular publishing, but they're planned and announced well before the date. OOB publishing events should be broadly communicated with content stakeholders. Other content owners may be counting on the regular publishing time. The repo admin must approve OOB publishing and must ensure that stakeholders have been warned/alerted adequately.

Hotfix publishing: Hotfix publishing is unplanned. It's for content that needs to publish outside a standard publish time with little or no prior notice. Unlike Regular and OOB publishing, only content that needs to be published is merged to the live branch. The entire master branch isn't merged. Hotfix publishing prevents content that is already in the master branch from going live before it should.

Hotfix publishing is typically for publish requests that come in after 3:00 PM, the last regular publish of the day for most repositories. You might also use Hotfix publishing during a content freeze for high priority changes that can't wait until the end of the publishing freeze. These publish runs don't involve a full publish cycle where public and private are cross merged.

Hotfix publishing also has some limitations. If you use the process to publish a new article, the **Edit** link on the article becomes unavailable until the next full publish cycle. But, the limited scope of hotfix publishing makes it a lightweight, easy, and fast solution for high priority unplanned publishing requests. Hotfix publishing requires business approver sign-off. Hotfix publishing is also subject to the availability of staff with the required permissions.

Hotfix publishing is reserved for legal, security, privacy, and executive escalations. **Hotfix publishing should be rarely used.** It isn't a replacement for good planning. Don't use the hotfix publishing process to publish large amounts of change. Hotfix publishing assumes you're targeting specific articles for publish.

Why does any of this matter?

If you work in a small repo with only a few contributors, the distinction between publishing types may not matter. But, if people from different teams work in your repo, and if your contributors are many, things can get complicated. The different teams and large number of contributors are constantly planning routine content releases around the planned publishing schedule. A team or contributor may have content in the master branch that is intentionally waiting for the next planned publish - the next morning, or for an event. An urgent unplanned publish to master will also push the other content live, unexpectedly. You can avoid this problem by being disciplined about planning OOB publishing. Use hotfix publishing only to move specific content live, leaving all other content waiting in master for the next planned publishing event.

General processes

General processes for planning your release appear in the Content release planning + process article in this contributor guide.

For Content & Learning, the Content Production team provides publishing support of different types and is typically limited to regular and OOB publishing. The process the Content Production team uses to run regular publishing is documented in the Content Production Service process guide.

How to run hotfix publishing

Prerequisites

To run hotfix publishing, you must meet one of the following requirements:

- You must be an admin of the repo.
- You must have write permission in the repo.

  And, if your live branch is protected, you must be explicitly named as a person who is permitted to merge the live branch.

  Or, you must be a member of a team that is explicitly permitted to merge to the live branch.

> **NOTE**
>
> Usually, if you are doing this work, you will know what level of access you have to the repo where you need to run hotfix publishing. You can tell if you are an admin in a repo if the **Settings** tab is visible to you. The most obvious way to tell if you are a write-level user is if the merge button is available to you in a pull request. Most repos have protected branches, so you may only be authorized to merge to certain branches.

Instructions

Follow these steps when the change you need to push live only includes a few commits:

1. Ensure the changes that you need to push live are merged to the master branch of the repo.
2. In your local clone of the repo, make sure you are in your local master branch:

   ```
   git checkout master
   ```

3. Fetch all upstream branches:

   ```
   git fetch upstream
   ```

4. Depending on your situation:

   - If you already have a copy of the live branch on your local computer, change to your live branch:

     ```
     git checkout live
     ```

   - If you need to create the live branch locally, enter:

     ```
     git checkout -B live upstream/live
     ```

     The latter part of this command causes your local live branch to track the upstream live branch.

5. Update your local live branch from upstream:

   ```
   git pull upstream live
   ```

   > **IMPORTANT**
   >
   > **DO NOT** allow your muscle memory to cause you to enter the pull command against master. Don't do it!
   >
   > If you do, run this command to reset your local live branch to match the upstream live branch:
   >
   > ```
   > git reset --hard upstream/live
   > ```

6. Now, find the pull request that has the changes you need to push live. Use `git cherry-pick` to move each relevant commit in order into the live branch, starting with the oldest:

   ```
   git cherry-pick <commit>
   ```

   Repeat for each commit.

7. After you cherry pick all the relevant commits, run:

   ```
   git status
   ```

   There should be nothing listed - go to step 9. However, if the changes are listed in green in the status message, you need to commit them.

8. Commit your changes:

   ```
   git commit -m "picked commits for hotfix publish"
   ```

9. Push the changes to your fork:

   ```
   git push origin live
   ```

10. Create a pull request from your fork's live branch to the live branch of the upstream repo, and merge it.

    Because the changes you moved to the branch were reviewed and validated in the master branch, it's OK to merge this PR yourself after you verify the changes.

Larger changes

If the changes you need to push live are more extensive than a few commits, you can use the `git checkout` command instead of `git cherry-pick` for hotfix publishing.

For example, say you want to move all the content from the iot-central folder. And, you also want to move a few related changed files to live. But you don't want to publish the rest of the changes from the master branch.

1. Follow steps 1-5 in the preceding section.
2. Then, use `git checkout` to copy the changes you want from your local master branch to your local live branch. Here an example of how to move all the content from one folder along with three other individual files:

   ```
   git checkout master articles/iot-central
   git checkout master bread/toc.yml
   git checkout master docfx.json
   git checkout master includes/iot-central-howto-raspberrypi-selector.md
   ```

3. Add and commit your changes, then push them to the live branch in your GitHub fork. Create a pull request from your fork's live branch to the live branch of the upstream repo.

If the other methods described in this article don't work, you can merge the entire contents of a pull request into your local live branch. You need to be certain about the contents you're bringing to the live branch. The `git cherry-pick` and `git checkout` commands are more precise and safer because they target specific, known files. Check the diff in your pull request with extra care. To pull an entire pull request to your local live branch, follow the steps for cherry picking in the preceding section, but use the following command in step 6:

```
git pull upstream pull/<PR#>/head:<users-branch-name>
```

Caution

Bringing in the changes by targeting a pull request risks merging the contents of the master branch or other release branches. It's *not* the preferred method.

Checking on build status

You can check the status of your build in the Docs Portal. Search for your docset (the docset for azure-docs-pr is azure-documents), and select the **Builds** tab. Under **Build History**, filter on the **live** branch in the **Select a branch** box. Builds may have to wait in the queue for up to 50 minutes during busy times. And, the build may then take from 15 to 30 minutes, depending on usage of the publishing platform. Outside work hours, queues and build times are often faster.

Pull request review and publishing schedules

7/7/2021 • 3 minutes to read

Pull request review schedule

Small, text only changes to existing articles are merged automatically in repositories where PRMerger runs. For more information about which pull requests qualify for automerge, see PRmerger filters.

The vendor PR review team is available six days a week.

Redmond Hours

DAY	PR QUEUE COVERAGE TIMES
Sunday	12:00 AM - 4:00 AM, 2:00 PM - 4:00 PM
Monday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM, 7:00 PM - 9:00 PM
Tuesday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM, 7:00 PM - 9:00 PM
Wednesday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM, 7:00 PM - 9:00 PM
Thursday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM, 7:00 PM - 9:00 PM
Friday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM
Saturday	no coverage

> **NOTE**
> All times are Pacific time (Redmond). Sunday afternoon hours are approximate and may be shorter or longer depending on the number of pull requests in the queue. The Sunday early morning shift begins September 2019.

Israel Hours These are the same times as the Redmond hours listed above, displayed in Israel timezone for the ILDC team's convenience.

DAY	PR QUEUE COVERAGE TIMES
Sunday	10:00 - 14:00
Monday	00:00 - 02:00, 10:00 - 14:00, 17:00 - 00:00
Tuesday	00:00 - 03:00, 05:00 - 07:00, 10:00 - 14:00, 17:00 - 00:00
Wednesday	00:00 - 03:00, 05:00 - 07:00, 10:00 - 14:00, 17:00 - 00:00

DAY	PR QUEUE COVERAGE TIMES
Thursday	00:00 - 03:00, 05:00 - 07:00, 10:00 - 14:00, 17:00 - 00:00
Friday	00:00 - 03:00, 05:00 - 07:00, 10:00 - 14:00, 17:00 - 00:00
Saturday	00:00 - 03:00

Daylight savings time is observed in both Israel and the US, but the effective dates are slightly off - the US goes to DST in early March, Israel in late March. The US reverts to regular time the first Sunday in November while Israel reverts one weekend earlier.

Extra publishing cycles

These publishing cycles, which happen outside of core US hours, are handled by the PR review team.

Publishing schedule - Azure-docs-pr

Redmond Hours The publishing schedule for Azure technical documentation in the azure-docs-pr repository is:

DAY	TIME	TYPE
Sunday	4:00 AM and 4:00 PM	Private repo
Monday - Friday	4:00 AM, 10:00 AM and 3:00 PM	Both public and private repos

Israel Hours These are the same times as the Redmond hours listed above, displayed in Israel timezone for the ILDC team's convenience.

DAY	TIME	TYPE
Sunday	14:00	Private repo
Monday - Friday	14:00 and 20:00	Both public and private repos
Tuesday - Saturday	01:00	Both public and private repos

Publishing schedule - dataexplorer-docs-pr

Redmond Hours The publishing schedule for Azure Data Explorer technical documentation in the dataexplorer-docs-pr repository is:

DAY	TIME	TYPE
Sunday	4:00 PM	Private repo

Israel Hours These are the same times as the Redmond hours listed above, displayed in Israel timezone for the ILDC team's convenience.

DAY	TIME	TYPE
Sunday	14:00	Private repo

Publishing schedule - other repos

Most repos have a morning publishing run between 9:30 and 10:30 PST, and an afternoon run between 2:30 and 3:30 PST. The complete list of repos and schedules is posted on the Content Production Service site.

Publishing team - hours available (PST)

The publishing team is working on requests and available for contact during the hours below. (File requests here. Contact here: pubdesktech@microsoft.com.)

DAY	WORKING HOURS (PST)
Sunday	6:00 PM - 2:00 AM (next day)
Monday	9:00 AM - 2:00 AM (next day)
Tuesday	9:00 AM - 2:00 AM (next day)
Wednesday	9:00 AM - 2:00 AM (next day)
Thursday	9:00 AM - 2:00 AM (next day)
Friday	9:00 AM - 6:00 PM
Saturday	no coverage

Holiday schedule

Limited pull request review and publishing will occur on the following Microsoft US corporate holidays to support authors working outside the US:

DATE	HOLIDAY	PR REVIEW AND PUBLISHING
January 18, 2021	Martin Luther King Jr. Day	Yes - limited hours
February 15, 2021	President's Day	Yes - limited hours
May 31, 2021	Memorial Day	No
July 5, 2021	Independence Day	No
September 6, 2021	Labor Day	No
November 25, 2021	Thanksgiving Day	Yes - limited hours
November 26, 2021	Day after Thanksgiving	Yes - limited hours
December 23, 2021	Christmas Eve	Yes - limited hours
December 24, 2021	Christmas Day	No
December 26-31, 2021	Holiday week	Yes - limited hours

DATE	HOLIDAY	PR REVIEW AND PUBLISHING
January 1, 2022	New Year's Day	No
January 3, 2022	Resume normal schedule	Yes - back to normal schedule

Troubleshooting Git errors

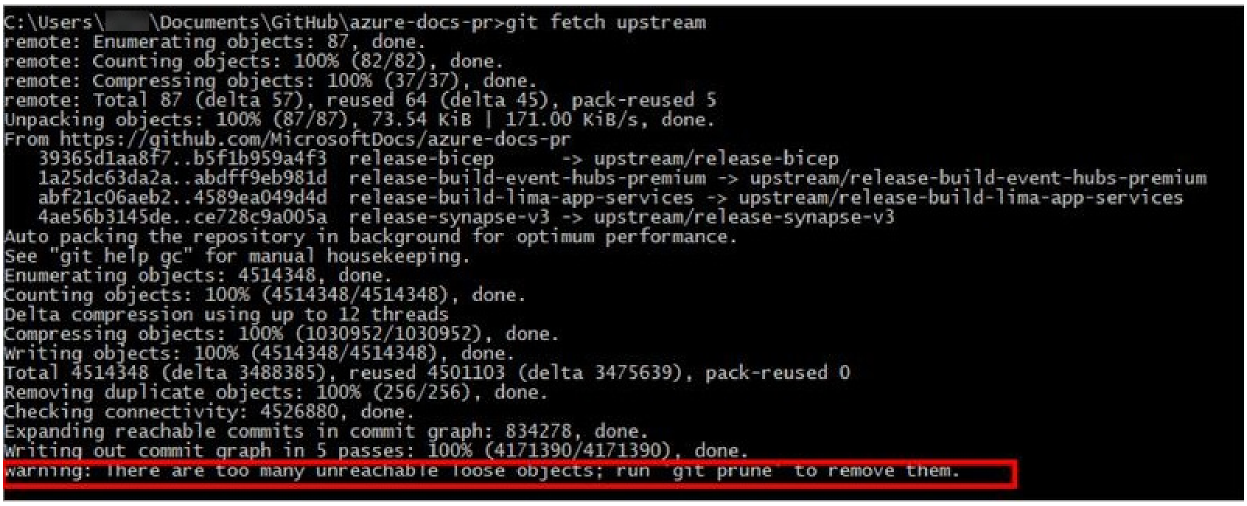
7/7/2021 • 2 minutes to read

This article provides instructions to resolve some Git-specific errors you may get when updating content. For DocsBuild validation errors, see Custom Validation.

Git warning: unreachable loose objects

When you fetch upstream or push a commit in Git, the following error appears:

`warning: There are too many unreachable loose objects; run git prune to remove them`



This warning appears when there are too many orphaned commits, also known as dangling commits. Commits that aren't accessible through a branch or tag is considered dangling.

To resolve this warning, run the *git prune* command:

```
git prune
```

Or a if you prefer, run the *git gc* were gc stands for garbage collection) command, which will not only clean up the dangling commits but will also do a compression on stored Git Objects, freeing up precious disk space.

```
git gc
```

After running the command, try to fetch or push again and your warning will be resolved.

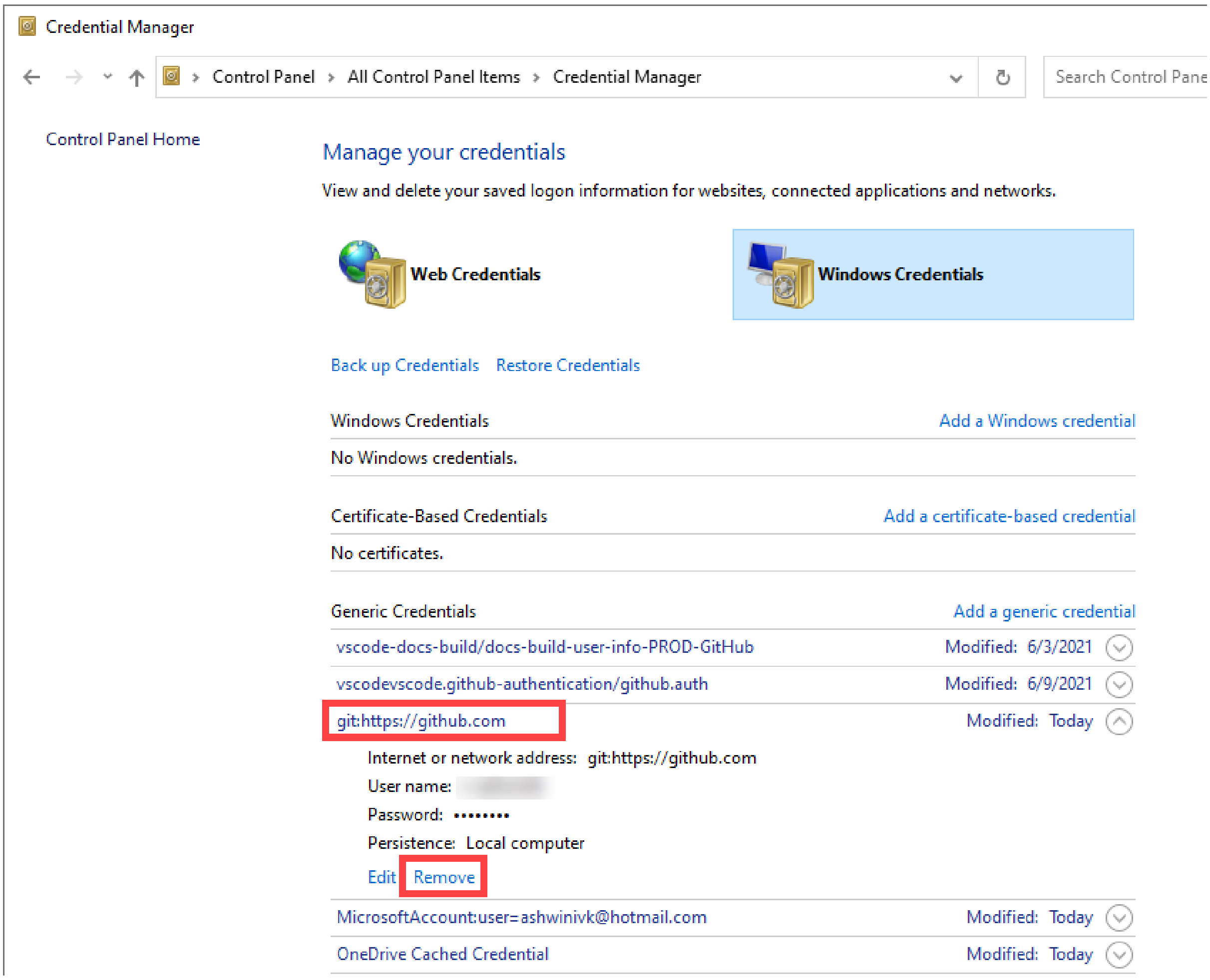
Re-authorize the OAuth Application `Git Credential Manager`

While setting up a new workstation, the user isn't able to clone a repo and receives the following error:

```
remote: The `tamram' organization has enabled or enforced SAML SSO. To access
remote: this repository, you must re-authorize the OAuth Application `Git Credential Manager`.
fatal: unable to access 'https://github.com/tamram/azure-docs-pr.git/':
The requested URL returned error: 403
```

To resolve this issue:

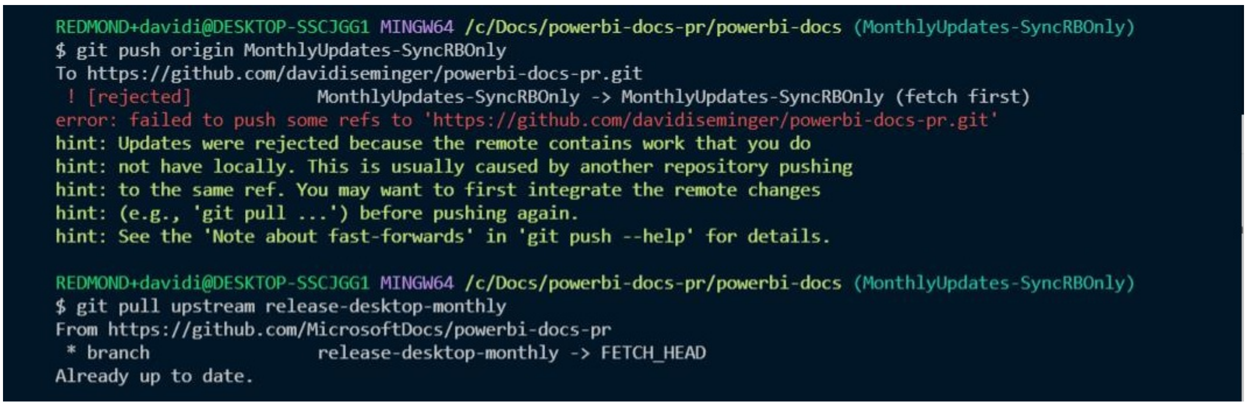
1. Go to Control Panel
2. Select **User Accounts**.
3. From the left side bar select, **Manage your credentials**. The Credential Manager page appears.
4. Select **Windows Credentials**.
5. Locate and expand the **git:** https://github.com entry.
6. Select **Remove**.



7. And then try to clone again.
8. If you are in a GitHub Desktop, sign out and then sign in again before cloning.

error: failed to push some refs to 'https://github.com/...'

You push your commits from your local working branch the following error appears:



```
REDMOND+davidi@DESKTOP-SSCJGG1 MINGW64 /c/Docs/powerbi-docs-pr/powerbi-docs (MonthlyUpdates-SyncRBOnly)
$ git push origin MonthlyUpdates-SyncRBOnly
To https://github.com/davidiseminger/powerbi-docs-pr.git
 ! [rejected]          MonthlyUpdates-SyncRBOnly -> MonthlyUpdates-SyncRBOnly (fetch first)
error: failed to push some refs to 'https://github.com/davidiseminger/powerbi-docs-pr.git'
hint: Updates were rejected because the remote contains work that you do
hint: not have locally. This is usually caused by another repository pushing
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push --help' for details.

REDMOND+davidi@DESKTOP-SSCJGG1 MINGW64 /c/Docs/powerbi-docs-pr/powerbi-docs (MonthlyUpdates-SyncRBOnly)
$ git pull upstream release-desktop-monthly
From https://github.com/MicrosoftDocs/powerbi-docs-pr
 * branch              release-desktop-monthly -> FETCH_HEAD
Already up to date.
```

The above message suggests that your working branch is no longer current with the default/main/master branch.

To resolve this error, run the following command:

> **WARNING**
>
> This could **erase all changes** in your release branch and make it current with the default/main/master branch. We recommend you save all changed files in a different location on your workstation before running this command.

```
git pull origin working-branch-name
```

Then try to push your commit again.

Git hard reset

If you have tried other solutions and can't resolve an error, or your error isn't listed you can hard reset your release branch, as a last resort.

> **WARNING**
>
> A hard reset will **erase all changes** in your release branch and make it current with the default/main/master branch. We recommend you save all changed files in a different location on your workstation before doing a hard reset.

Run the following git commands to reset your release branch:

```
git checkout release-branch-name
```

```
git fetch upstream
```

```
git reset --hard upstream/release-branch-name
```

```
git push origin release-branch-name --force
```

Data, BI, and Reports overview

3/5/2021 • 2 minutes to read

A dashboard of available Content & Learning reports is available at https://aka.ms/clcia.

Data scenarios

In general, the reports and dashboards available provide insights at two different tiers of granularity: at the portfolio level and at the article level.

Portfolio-level metrics

Metrics on a portfolio of article (such as all of the articles published to docs.microsoft.com, the articles in the azure-docs-pr repo, or the articles for a specific service) are typically gathered for partners and stakeholders who need a high-level view of content health.

QUESTION	RELEVANT METRICS	REPORT
What is the overall traffic or CSAT for a content set or section over time?	PageViews, Visits, Unique visitors, CSATHelpfulRate	Documentation and reference report
Are customers engaging with the docs or are they reaching a page and leaving right away?	Bounce rate	Documentation and reference report
How is traffic coming to the doc set?	Referrer types	Documentation and reference report
How many GitHub issues do we have by repo? What types of issues are customers having?	Open and closed GH issues	GitHub issues report, GitHub

Article-level metrics

QUESTION	RELEVANT METRICS	REPORT
What is the overall traffic or CSAT for a specific page or section over time?	PageViews, Visits, Unique visitors, CSATHelpfulRate	Content Performance report
What are customers saying about my article?	Verbatims, GitHub issues	Verbatims report, GitHub issues report, GitHub
How is traffic coming to my section? A specific page?	Referrers, Referrals	Content Performance report, Microsoft Docs Metrics (MDM)
What are customers engaging with on the page? Or are they reaching a page and leaving right away?	Heat map, click data, Bounce rate	Content Performance report, Microsoft Docs Metrics (MDM)
Which keyword searches are driving traffic to my section? Which keywords are driving to a specific page?	Internal and external search terms	Microsoft Docs Metrics (MDM), Google search analytics

Metric definitions

7/14/2021 • 16 minutes to read

Here are the definitions of the metrics most commonly used in our reports.

Traffic metrics

- PageViews
- Visits
- Visitors
- OrganicSearchReferrers

Composite metrics. These take several other metrics as inputs:

- PrimaryKPIScore (and PrimaryKPIScoreRank)
- Percentage of page views to higher performing content
- Freshness

Primary metrics. Used to directly calculate an article's KPI score:

- ClickThroughRate
- CopyTryScrollRate
- BounceRate
- ExitRate
- LearnModuleCompletionRate
- LearningPathCompletionRate

Secondary metrics. Not used to calculate the KPI score, but useful for gaining insights into the content:

- CopyRate
- TryRate
- ScrollRate
- MedianMaxScroll
- DwellRate
- StartTutorialRate
- UserEngagementRate
- AvgCopyTryEngagementRate
- DeployClickRate
- CSATHelpfulRate
- CSATResponseRate
- GitHubOpenIssueCount
- GitHubTotalIssueCount
- GitHubNewIssuesCount
- RatingVerbatim

Referrer metrics. Use to determine the origin of incoming traffic

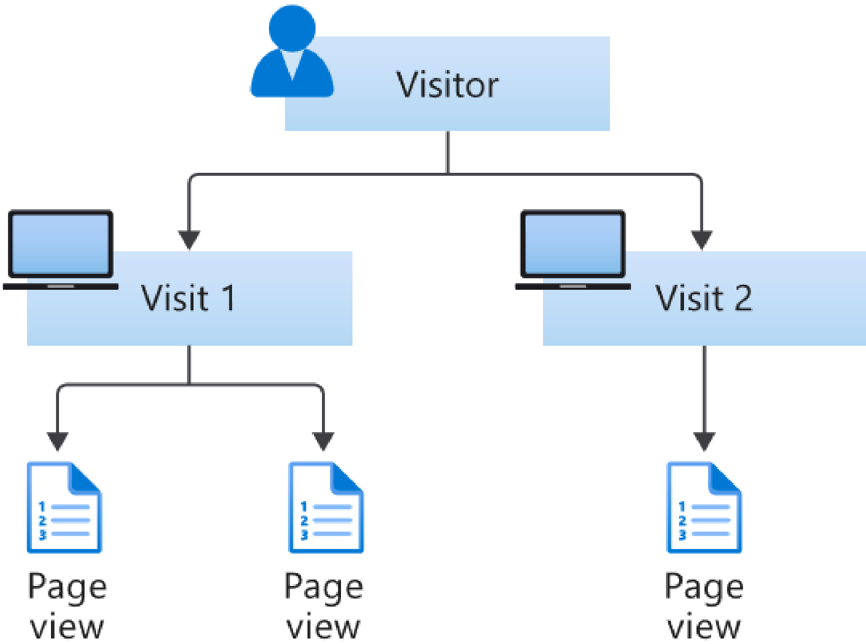
- OrganicSearchRate
- PaidSearchRate

- DirectTrafficRate
- MicrosoftTrafficRate
- ExternalTrafficRate
- SocialMediaTrafficRate

Traffic metrics

Why they're important: Indicates the discovery and awareness of the content. Growth in traffic should normally correspond with adoption and usage of the product or service the content covers.

How to improve them: Two ways: 1) Optimize for SEO to ensure your content is as discoverable through organic search as possible; and 2) Ensure strong engagement and satisfaction with the content to build trust with users and a stronger likelihood they will return to the site and the same area of content the next time they need help or information.



PageViews

Every time a page (article) is loaded.

Visits

A sequence of page views in a sitting. A session starts when the user first arrives on a site and ends after 30 minutes of inactivity.

Visitors (MAU)

An individual user to a site. Unique users are identified via MC1 web cookies, which are created for each user-browser-machine. Therefore, the same user could be counted as two unique users if they visited the same site from different browsers. If a user clears cookies in the browser, it will then be a new MC1 visitor.

Note that visitors are what make up the monthly active user (MAU) metric in organization-level reporting.

Don't sum visitors from multiple pages together or you'll unintentionally over count users who visited more than one of the pages. The data is deduped on the data collection side. Instead use https://aka.ms/docrefdashboard.

OrganicSearchReferrals

The page views from organic search referrals.

Composite metrics

PrimaryKPIScore (and PrimaryKPIScoreRank)

The PrimaryKPIScore (and affiliated metric, PrimaryKPIScoreRank) appear in the Content Performance report.

Definition: This is a score that lets us roll up the primary metrics into a single number we can rank when we compare articles against each other to determine which ones are lower performing. The topics will be ranked by the primaryKPIScore in a type group for us to identify the relative lower performing topics in the set of content by flagging if the rank is below 25%.

Formatting: In the content performance dashboard results, articles with `PrimaryScoreRank` highlighted in light yellow are performing in the bottom 25% when compared to articles of the same topic type. Articles highlighted in dark yellow are performing in the bottom 10% when compared to articles of the same topic type.

Why it's important: We use this score ranking to determine whether an article is lower performing versus other articles of the same topic type. After you sort to find the most-viewed articles in your content area, the score allows you to identify articles where your efforts are most likely to produce impact.

How we calculate the primary KPI value: Here's how the primary KPI rollup score is calculated. In the end, the calculations may seem arbitrary, but they are effective because the resulting primary KPI score lets us compare the relative performance of thousands of articles and identify the lowest performing ones.

Rationale: The desired behavior for most pages is for customers to come to a page, read to learn or understand new concepts, click through to explore more, or copy code snippets. We don't want them to bounce away! Landing pages have one purpose, and that is to help a customer find what they are looking for and click through to an actual technical article. For this reason, the 'good' half of the equation is focused on click-through rate. We don't want users to exit a landing page, and we definitely don't want them to bounce, so we subtract those two bad things from the click-through rate. We multiply the click-through rate by 2 because we are subtracting two negative metrics from the positive metric.

KPI formulas by type group

TYPE GROUP	TOPIC TYPE	LEGACY CALCULATION	NEW CALCULATION
Article	article	DwellRate + ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Conceptual	conceptual	DwellRate + ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Overview	overview	DwellRate + ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Quickstart	quickstart	ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Tutorial	tutorial	ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Sample	sample github-sample	ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate

TYPE GROUP	TOPIC TYPE	LEGACY CALCULATION	NEW CALCULATION
End user help	end-user-help	DwellRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
How to	how-to	DwellRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Troubleshooting	troubleshooting	DwellRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Hub/Landing	hub-page landing-page sample index	2 * ClickThroughRate - (ExitRate + BounceRate)	No change
Reference	reference language-reference managed-reference ioctl structure callback enum enumeration function include interface macro method struct error reference	ClickThroughRate - BounceRate	CopyTryScrollRate - BounceRate
Others	Topic types not on the metadata Allow list or is missing	DwellRate + ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Learn module	interactive-tutorial	LearnModuleCompletionRate	No change
Learning path	interactive-tutorial	LearningPathCompletionRate	No change

NOTE

- If a topic type isn't covered in the content performance model, it's grouped with **others**.
- Approved ms.topic types are listed in the ms.topic allowlist.
- Descriptions for common article types are listed in the metadata attributes article of this Contributor Guide.
- The PnP Topic Type groups example-scenario, reference-architecture, design-guide were deprecated from reporting logic in Feb 2021.

Exception tracking

- **SampleIndex**: Articles where ms.topic=article and the title contains the word "Samples" are treated as SampleIndex.
- **Troubleshooting**: ms.topic starts with 'troubleshooting' or title contains 'troubleshoot' are treated as

TroubleShooting. These customizations were created at a particular point in time; ideally correct metadata would be assigned directly in the articles.

Percentage of page views to higher performing content

The total number of page views (PVs) to pages whose Primary KPI Score metric is above the bottom 25% versus its peers, divided by the total number of page views

- **Caveats**: The bar can be set anywhere. It's currently at the bottom 25%. As we increase the % of PVs to lower performing content, we'll ratchet the bar up.
- **Why it's important**: This is a rollup metric for topic-level performance that helps us steer our efforts across broad sets of content.
- **How to improve it**: Focus on improving high-traffic articles that under perform versus their peers.

Freshness

How many days have passed between the Last Review Date to the end of reporting month, categorized as: 0-90 days (freshest), 91-365 days (fresh), >365 days (stale).

- **Why it's important**: This metric provides context for the primary metrics - for example, an article that has poor performance may just need to be updated to fix outdated screenshots, outdated procedures, broken links, and other things that could be blocking or discouraging users. For more information, see the content maintenance checklist/freshness article.

Primary metrics

ClickThroughRate

The ClickThroughRate for an article is equal to (# of page views with click activity) / (# of total page views). All link click events are counted regardless of the link target (links in the body of the article along with navigation, TOC, in this article).

- **Formatting**: This metric is a higher-is-better metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Caveats**: Clicks on the **Copy** or **Try** buttons for code blocks aren't calculated in this metric.
- **Why it's important**: We use this metric on several page types. If we put links and buttons in front of users, we want them to click those links and buttons.
- **How to improve it**: Use the MDM browser extension or the Click report to drill into clicks on the page. Check to see if some links lower on the page outperform links higher on the page. Moving them up, where appropriate, can reduce the exit rate and increase the click-through rate. Also, consider clearer and more actionable labels on links and buttons to sell the outcome of clicking through. Consider linking to other things of potentially more interest and removing links that aren't of interest.

CopyTryScrollRate

% unique visitors who copy, click try, or scroll on the page / total unique visitors to the page

- **Definition**: Engagement is counted whenever one of the actions (copy, try, scroll) happens no matter which one is first or how many times a user takes the same action
- **Formatting**: This is a higher-is-better metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Why it's important**: We use this metric to measure the amount of overall page engagement. Did customers find something on the page or did they copy code?
- **How to improve it**: If the BounceRate is high, this lowers the CopyTryScrollRate, so address BounceRate

/ SEO issues. For additional guidance, see Tips for low copy-try-scroll rate.

BounceRate

% of page views with a duration less of than 5 seconds and without clicks, scrolling, or copying of content.

- **Formatting**: This is a higher-is-worse metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Caveats**: The industry standard definition is a little different. Typically, it's defined as "% of visits that had only one page view in the session."
- **Why it's important**: We use this metric on all topic types. Leaving a page within 5 seconds without engaging most likely means a poor customer experience.
- **How to improve it**: Do a referrer analysis. Often a particular referrer is to blame for poor performance because they're pointing users to the wrong content to answer their questions. Work with referrers to point to better content or change the content to meet the expectations of that referred traffic. Lastly, write a clearer page title, heading 1, and intro paragraph to set expectations. If users can't tell what's in it for them right away, they'll bounce.

ExitRate

% of page views that are the last page view in a session. Currently ExitRate is only listed for hub and landing pages in the table on the Documentation page of the Content Performance Dashboard report. To find ExitRate for all your articles, go to the Overview page of the Content Performance Dashboard report. Select **Download the Excel Report** and choose either high-performing or low-performing articles. It's a large file, so it'll take a while to download.

- **Formatting**: This is a higher-is-worse metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Why it's important**: We use this metric on navigation pages like hubs, which are designed to get users to other pages. Exiting the site while viewing these pages is sometimes okay, but we don't want high-traffic navigation pages to have an above average exit rate.
- **How to improve it**: Users tend to exit when they can't find what they're looking for. Use the MDM browser extension to drill into clicks on the page to see if some links lower on the page outperform links higher on the page. Moving them up where appropriate can reduce the exit rate and increase the click-through rate.

LearnModuleCompletionRate

% of users who complete the module after starting it

- **Formatting**: This is a higher-is-better metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Why it's important**: We want to know if users are completing the modules.
- **How to improve it**: Consider shorter and more engaging content that creates *Aha* moments sooner.

LearningPathCompletionRate

% of users who complete the learning path after starting it

- **Formatting**: This is a higher-is-better metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Why it's important**: We want to know if users are completing the units.
- **How to improve it**: Consider shorter and more engaging content that creates *Aha* moments sooner.

Secondary metrics

CopyRate

Visitors who have a Copy event / Total visitors on the page

Definition: The percent of visitors that trigger a copy event for a page. Copy events include clicks on Copy button for code snippets, by clicking Copy in the right-click context menu, or key combination CTRL+C)

TryRate

Visitors that have a Try event / Total visitors on the page

Definition: The percent of visitors that trigger a try event for a page.

ScrollRate

% of visitors who have scroll event / Total visitors on the page

- **Why it's important**: Use this metric on any prose-heavy topics to see if users are engaging with the topic. A low percentage could indicate customers stayed longer than 5 seconds but didn't feel compelled scroll down the page and go below the fold.

MedianMaxScroll

Median max scroll depth, based on page views (one PV has one MaxScroll)

- **Why it's important**: Use this metric on any prose-heavy topics to see how far on the page users scrolled. Is there a point where most users leave the page?

DwellRate

Median dwell time/Estimated read time

- **Definition**: Dwell time is how long (in seconds) a user stayed on the page. Estimated read time is calculated during the Build process as 200 words/minute with a minimum of 2 minutes. It includes all text within HTML tags (monikers, code snippets, ALT+text, INCLUDES, tabbed conceptual, and zone pivots).
- **Why it's important**: We use this metric on any prose-heavy content types. Do users stay long enough to read it?
- **How to improve it**: Consider shorter content. Users often scan before they dive in. If it looks TLDR (too long, didn't read), they might leave.

Note: In June 2020, this was changed from **DwellTimeDelta** (MedianDwellTime – Estimated reading time)/Estimated reading time

StartTutorialRate

% of users who start a tutorial from a quickstart page

- **Caveats**: Intended quickstart design includes this tutorial button. Some pages may not include such a button/link, and get penalized by this metric. Docs signed off on continuing to include these in the reporting and not trying to exclude them.
- **Why it's important**: We use this metric on quickstarts. Do users start a tutorial after the quickstart?
- **How to improve it**: If the page doesn't include this button, consider adding one. Helping users learn something is a good thing. If it does include this button, consider a more descriptive call to action on the Tutorial button/link/intro. Use strong verbs and sell the outcome of completing the tutorial: what can you do after learning the next thing... or, try linking to a different tutorial.

UserEngagementRate

% of users who click at least one copy/try link (includes all copy events like right-click, Ctrl+C, and so on)

- **Caveats**: We eventually want to measure % of links on the page clicked, as a proxy for completion rate. This metric is interesting to consider in the meantime.
- **Why it's important**: We use this metric on any article types with steps in a procedure, and samples of code to copy/try. If we include these elements, we want users to click them. What % of users do?
- **How to improve it**: If the page doesn't include this type of UI control or doesn't encourage copying, consider relabeling the article type so we can measure it more appropriately. See roadmap request around "no-code" quickstarts, tutorials.

AvgCopyTryEngagementRate

% average total clicks on copy/try links per user (includes all copy events like right-click, Ctrl+C, and so on)

- **Caveats**: We eventually want to measure % of links on the page clicked, as a proxy for completion rate. This metric is interesting to consider in the meantime.
- **Why it's important**: We use this metric on any article types with steps in a procedure, and samples of code to copy/try. If we include these elements, we want users to click them. How often do users click these elements?
- **How to improve it**: If the page doesn't include this type of UI control or doesn't encourage copying, consider relabeling the article type so we can measure it more appropriately.

DeployClickRate

% clicks on deployable resource per page view

- **Caveats**: Some pages may not include such a button/link, and get penalized by this metric. AAC signed off on continuing to include these in the reporting and not trying to exclude them.
- **Why it's important**: We use this metric on reference architectures. Do users click the deployable resource?
- **How to improve it**: Consider placement on page or stronger calls to action. Consider the relevance of the scenario.

CSATHelpfulRate

% of responses saying the content was helpful divided by total responses

- **Formatting**: Light yellow is below mean. dark yellow is a standard deviation below mean.
- **Caveats**: Null values appear for pages that don't ask for feedback.
- **Why it's important**: This metric is our only quantitative attitudinal metric. It's always worth reading the verbatims, but with such low response rates, we don't include it in primary metrics.
- **How to improve it**: Read verbatims and act on them as much as possible.

CSATResponseRate

(CSATTotalResponses * 1.0 / Visitors) - % users that rated a topic divided by all users

- **Caveats**: Null values appear for pages that don't ask for feedback.

GitHubOpenIssueCount

Total number of issues open for the article as of the end of the reporting period.

GitHubTotalIssueCount

Total number of issues (open + closed) for the article as of the end of the reporting period.

GitHubNewIssuecount

Total number of new issues opened for the article on all locales as of the end of the reporting period.

RatingVerbatim

The number of verbatim created when a user rates the article in the reporting period.

Referrer metrics

OrganicSearchRate

% of page views from organic search (direct referrals only). Organic search comes from Google and Bing and other search engines.

- **Why it's important**: We want to make the most of the traffic we can get from Google and Bing. An increase in this % is a good indicator that the content is generally more discoverable through search. We want the raw volume from search to always grow, but if another channel happens to grow *more* that's OK.
- **How to improve it**: Try SEO best practices.

PaidSearchRate

% of page views from paid search campaigns. Paid Search comes from paid search campaigns on a search engine.

- **Why it's important**: We do not pay for search ads, so this will almost always be close to 0.

DirectTrafficRate

% of page views from direct traffic.

- **Why it's important**: Having links to your topic from within a product provides in-context Help for users.
- **What qualifies as direct traffic**: Direct traffic doesn't have a referring URL, making it difficult to identify the source. Direct traffic sources can be from an email campaign, in-product links (not Web-based), a bookmark, by typing in a URL directly, a server-side redirect, a fwlink or aka.ms link with an outdated target URL, or a bot. Get more details on potential sources
- **How to improve it**: See this article for ways to improve.

MicrosoftTrafficRate

% of page views from any Microsoft domain, including docs.

- **Why it's important**: Having backlinks to your topic helps with SEO.
- **How to improve it**: Do a referrer analysis. Often a particular referrer is to blame for poor performance because they're pointing users to the wrong content to answer their questions. Work with referrers to point to better content or change the content to meet the expectations of that referred traffic. Lastly, consider clearer page titles and intro paragraphs to set expectations. If users can't tell what's in it for them right away, they'll bounce.

ExternalTrafficRate

% of page views from users following links from other domains outside Microsoft.

- **Why it's important**: Having backlinks to your topic helps with SEO.
- **How to improve it**: Because we don't control and can't easily influence these referrers, we may consider ways to meet the expectations they appear to be setting when they link to us: "Looking for X? It's here."

SocialMediaTrafficRate

% of page views from links posted on social media like Facebook and Twitter.

- **Why it's important**: Having backlinks to your topic helps with SEO.

Content Performance report

7/7/2021 • 8 minutes to read

Use this report to get traffic and engagement metrics such as click-through rate (CTR), scroll rate, and bounce rate for individual articles, and track the overall performance for a content set. The report helps us understand how customers engage with our documentation, and whether they are using it as we expect. For example, the purpose of a landing page is to provide navigational links that surface content that helps or interests our customers. If a landing page has a high bounce rate and exit rate, meaning customer's aren't clicking to another topic within the content set, the landing page is not meeting its designed purpose.

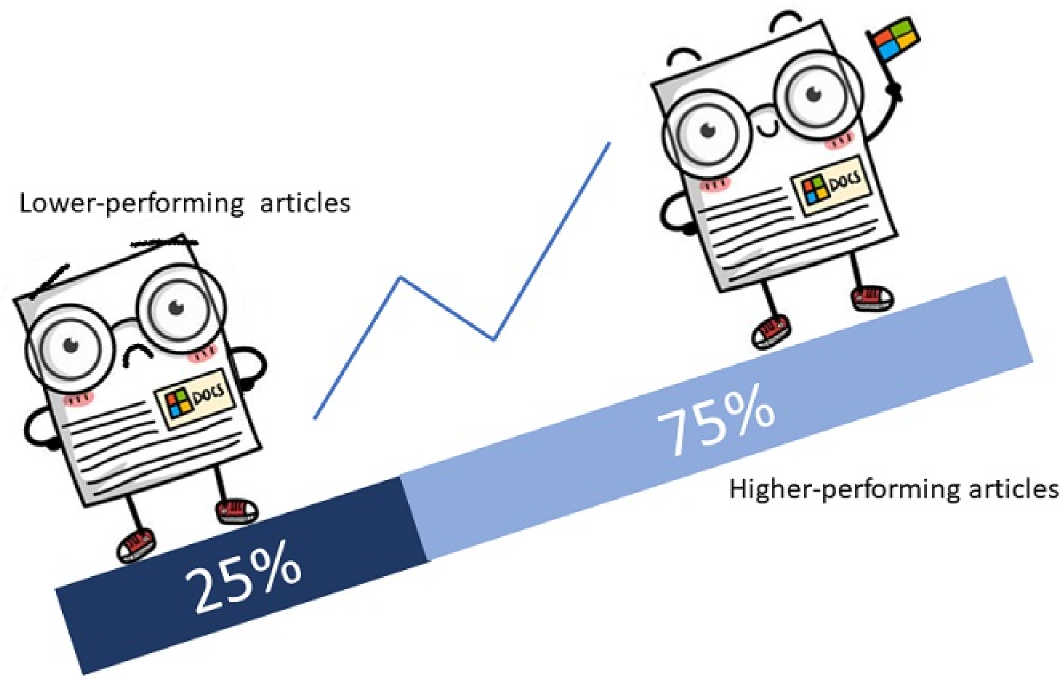
Select the report that maps to your organization. Be sure to bookmark the report by using the aka.ms link to ensure you get the latest version of the report.

- Content & Learning Docs & PnP: https://aka.ms/contentperformancedashboard
- Business Application Group: https://aka.ms/contentperformance-bag
- Microsoft 365: https://aka.ms/contentperformance-m365
- Microsoft Learn: https://aka.ms/mslearnreport

How is engagement measured?

Each topic is given a KPI Score based on the KPI formula assigned to its topic type (ms.topic metadata). The score is then stack ranked with articles of the same topic type to determine which ones are lower performing. For example, the formula for a Quickstart is CopyTryScrollRate + ClickThroughRate - BounceRate, which looks like this when data is filled in (.76 + .18 - .19) = .75. The score is then stack ranked with all the other Quickstarts in the Power BI report.

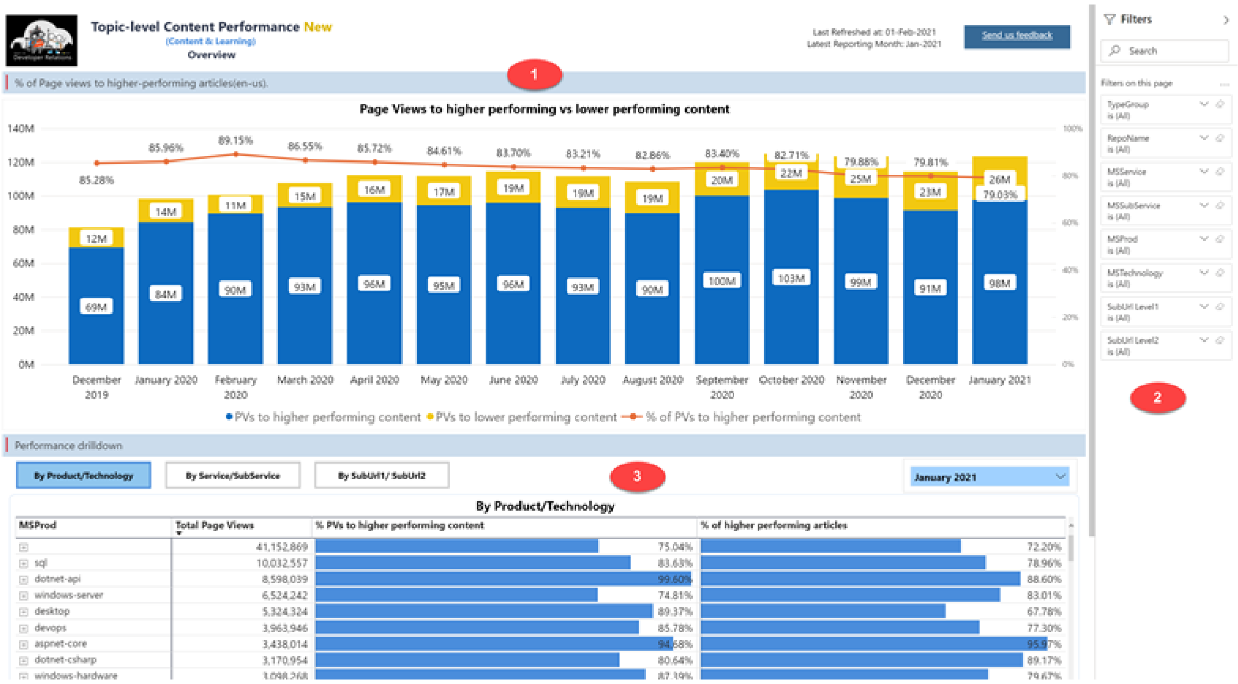
Topics whose primary KPI score ranking falls in the bottom 25% of all articles of that type are considered **lower-performing**. Articles whose performance is in the top 75% are considered **higher-performing**. The report highlights high-traffic lower-performing content so you can target your improvement efforts where they will make the most difference.



This recorded training session covers both the high-level strategy of our reporting and the detailed how-to of using the report and diagnosing performance problems.

Tracking performance for a content set

This page of the report is where you can monitor and track content performance progress across a Topic Type, Service, Product, URL pattern, or Repo. For example, in Fiscal Year 21, the C&L Docs team set an OKR to have 80% of overall page views go to higher performing pages.



Historical performance snapshot (Area 1) - Displays a historical snapshot of the monthly trends of page views of higher performing pages to lower performing pages. Hover over a month in the bar chart to expose a Tooltip that provides the total PVs, % of PVs, MoM PVs, and #articles for each of the low and high-performing groups.

Filter by content set (Area 2) - Use the filters to view the monthly performance data by content type group, repository, service, product, or URL. The low and high-performance ranking is defined by the KPI ranking of the Type Group the topics are a part of and not by the topics within a particular content set.

Drill down of performance by month (Area 3) - Use the **Performance drilldown** section to view the content performance data by Service, Product, or two levels of the URL path for a given month. This view is useful for identifying product or feature areas that might need some investment of effort to improve the customer experience.

Resources and archive (footer) - Download detailed Excel worksheets for previous months (C&L version only), access the Kusto query library, and find key links to the Contributor Guide.

Tracking performance at the page level

Page-level data appears on the **Documentation** and **Reference** pages of the C&L report, and the **Business Applications Group** and **Microsoft 365** pages in their respective reports. Similar data is available in the Learn report on two pages devoted to learning paths.

Two-month filter (Area 1) - Use the buttons to filter by current or previous month. Because of the size of the Docs portfolio, the data is limited to two months.

Filter to your content set (Area 2) - Use the filters to view the monthly data by content type group, repository, service, product, URL, author, and more.

Page-level metrics and KPIScore Rank (Area 3) - Find an article's primary KPI score and primary key metrics and how they perform as compared with articles of the same type. The score rank or metric is highlighted in light yellow if it falls in the bottom 25% and dark yellow if it falls in the bottom 10%. When you sort by highest page views, the list of dark yellow and light yellow results is your list of articles to work on. Focusing on high page view articles ensures you are targeting for impact. The values that appear on a grey background aren't factored into that article's KPI score, but are available for more sleuthing. To identify which metrics a page's KPI score ranking is based on, see Composite Metrics.

View weekly/monthly trends and get Referrer and Click data (Area 4) - Use the **View trends** or **Page drilldown** buttons to view the weekly or monthly trends for a specific article and get drill-down reports in one single view (CTR, Referrers, Verbatim, and GitHub issues).You need Kusto cluster access for both of these dashboards. See how to request access in the next topic and get more details about using these dashboards.

Request access to the Kusto data clusters

To use these resources, you must join the **CL Dashboard User Group** security group through MyAccess. The security group has been granted access to the **followercgadataout** cluster by the CGA team.

- For most users, no action is required. The entire C&L team, as well as any members of the previous Content Learning Dashboard User Group, have been added to the new security group.
- If you don't have access, you should request access to the new CL Dashboard User Group (links below).

Select the links in the table to submit a MyAccess request. A pre-populated form should appear. MyAccess links can be buggy though. If the pre-populated form does not appear, enter the hyperlink text into the MyAccess search (for example, "CL Dashboard User Group").

KUSTO CLUSTER	ROLE	NOTES
CGA Kusto Follower (followercgadataout.westus.kusto.windows.net) through CL Dashboard User Group	Read-only	**Required** for all users. The default cluster.
CloudMine-Data (https://1es.kusto.windows.net)	CloudMine-Data reader	**Required** for all users to view GitHub issue information in the Drill down dashboard

KUSTO CLUSTER	ROLE	NOTES
CGA Kusto Public (cgadatamall.westus.kusto.windows.net)	Read-only	**Optional** for all users. A general purpose public cluster for ad hoc queries

IMPORTANT

The CGA Public (cgadatamall) cluster is intended for exploratory queries. It is not scaled for computationally expensive queries or production workloads.The followercgadataout cluster is higher capacity, but not infinite capacity. There are no guarantees on performance. Regardless of which cluster you use, we recommend limiting the output using constraints like date ranges and content filters to improve performance and avoid memory limits.

Data scope and update frequency

The Content Performance reports are updated around the 2nd of each month and support metrics for all documentation published to docs.microsoft.com, except for documentation under these paths:

- /answers
- /archive
- /contribute
- /dynamics
- /learn - Removed in May 2020 from current performance dashboard and covered in Learn report
- /legal
- /locale
- /previous-versions
- /search
- /teamblog

Data scope for Docs Partners

The content on docs.microsoft.com that's managed by the Microsoft 365 and Business Application Group were separated from the **Content & Learning** scope and reported in another Power BI report.

ORGANIZATION	URL PATTERNS

ORGANIZATION	URL PATTERNS
Microsoft 365 Contacts: John Martin, Daniel Simpson; Serdar Soysal	/compliance /deployedge (contact: Colleen Williams) /deployoffice /education/windows /internet-explorer /exchange /hololens /lync /microsoft-365 /microsoft-desktop-optimization-pack /microsoft-edge (contact: Reeza Ali) /microsoft-store **/microsoftteams** /office /office365 /OfficeUpdates /OfficeUpdates /onedrive /openspecs (moved to this report Jan 2021) /Project /SchoolDataSync /sharepoint /skypeforbusiness /SharePointMigration /surface /surface-hub /windows/application-management /windows/client-management /windows/configuration /windows/console /windows/deployment /windows/privacy /windows/release-information /windows/security /windows/whats-new /windows/windows-10 /windows-insider /workplace-analytics
BAG (Dynamics 365, Power Platform, and Data Integration)	/ai-builder/ /common-data-model/ /connectors/ /data-integration/ /dynamics365/ /forms-pro/ **/industry/** /powerapps/ /power-automate/ **/power-bi/** /power-platform/ /power-query/ /powerquery-m/ /power-virtual-agents/

NOTE

- docs.microsoft.com/windows contains content for both **IT Pro** (owned by the Microsoft365 team) and **Developers** (owned by Content & Learning team). Only **IT Pro** content is listed. The rest of the pages under /windows are part of the Content & Learning scope.
- /microsoftteams (**bold**) is available in both the Microsoft 365 and C&L reports.
- /power-bi and /industry content (**bold**) is in both the C&L and BAG reports.
- Any repos that contain **office**, **kaizale-docs**, or **stream-docs-pr** are counted in the Microsoft365 scope.

FAQ

What does the count in the PowerBI reports refer to?

The number in the Filter Pane is a default feature in Power BI. For example, on the **Documentation** page, the data spans two months, so this number is the total of both months. If you need total articles in a month, the number is displayed at the top of the page and is also available in the Tooltip for each month on the **Overview** page.

If I make an update to the metadata of my topic, when will it show up in the report?

Any updates made before the last day of each month should show up in the monthly report refresh (the second of the month). The new metadata will be applied to the previous month and current month, but only the current month on the **Overview** page. Metadata updates can take 24-48 hours to get into the system.

Why do I see two records of my page with the same URL, but one that has low page views?

Two records of the same page typically means one record is a redirect. If they have two different TopicKeyIds, the redirect will eventually receive no page views as the cache is cleared and any existing links are updated. If you want to combine the data from the redirected page with the new page, then you need to set the `redirect_document_id` in the redirect.json file to `true`. Find more information for redirecting articles.

Why do I get different page view totals based on whether I filter by Repo, URL, or Service metadata?

The page view data might be different because Service and Product values can span multiple repos and URL patterns. If there's any missing metadata in a repo, then you could get higher page views for the repo or URL than for the Service or Product.

Next steps

By using the content performance dashboard to find high-traffic lower-performing content in your content sets, you can optimize the impact you have when you're working to improve published content. Here's some next steps:

- Troubleshoot lower-performing articles
- Understand how the primary KPI is calculated for each article type
- Review report release notes and known issues

Content Health report & score

7/8/2021 • 7 minutes to read

NOTE

Please review the Content & Learning OKRs & key results related to health metrics.

On a quarterly basis, proposed updates to the score and report will be considered, such as including the Acrolinx Clarity score or an SEO score.

In this article, you'll learn what content health is and how use Content Health score to prioritize your investments.

What is Content Health?

Content Health refers to *what we do, as content developers, to ensure quality content*. In response to health signals, we often perform content maintenance. Maintenance is the process of updating content so that it is accurate, free of defects, and up to date.

Content Health is different from content performance, which measures what our customers do, including engagement and feedback.

What is the Content Health Score?

The **monthly Content Health score is a composite of the average of several subscores.** We use this score to monitor the health of our articles and modules. You can see the composite, *Content Health Score*, and each of its subscores in the Content Health report at https://aka.ms/contenthealth.

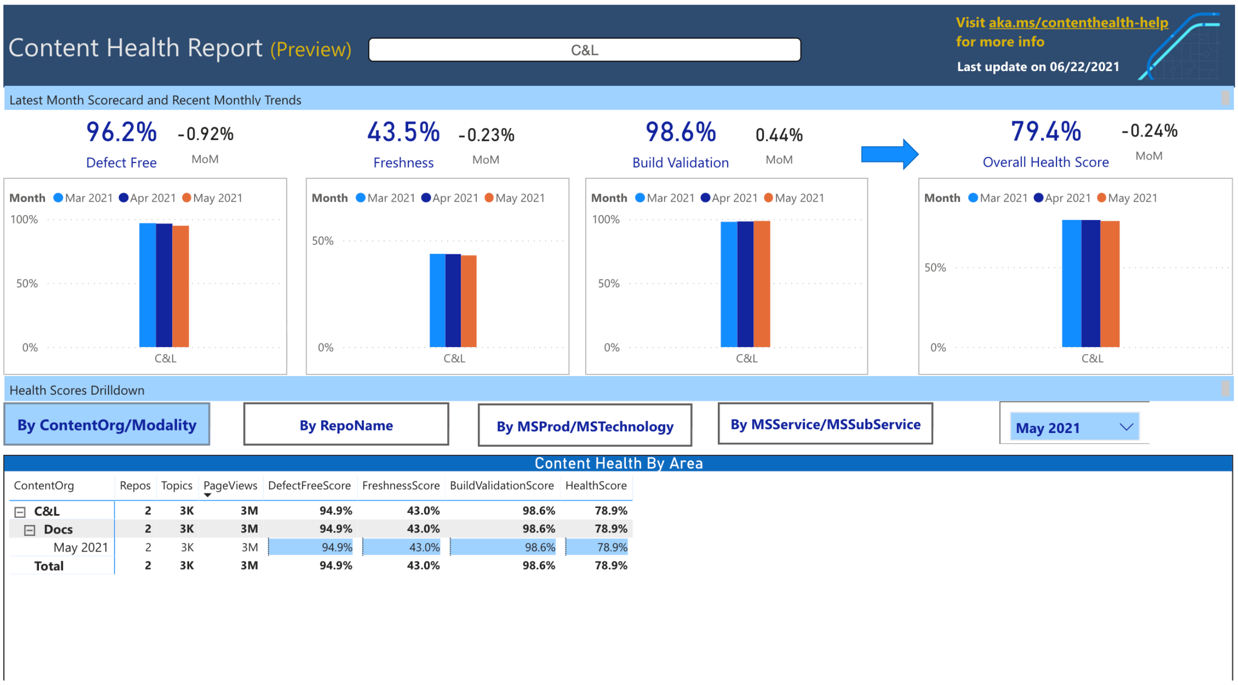
The **audience** for this report are M1/M2 managers and leadership team, to get an overview of health of their content repos. After further analysis using other reports, writers and managers can figure out resourcing needs to improve metrics as required.

At this time, the score is a composite of three subscores of **equal weight**:

- Defect-free page views
- Fresh page views
- Build validations

On a quarterly basis, proposed updates to the score and report will be considered, such as including the Acrolinx Clarity score or an SEO score.

Content Health report



When is it calculated?

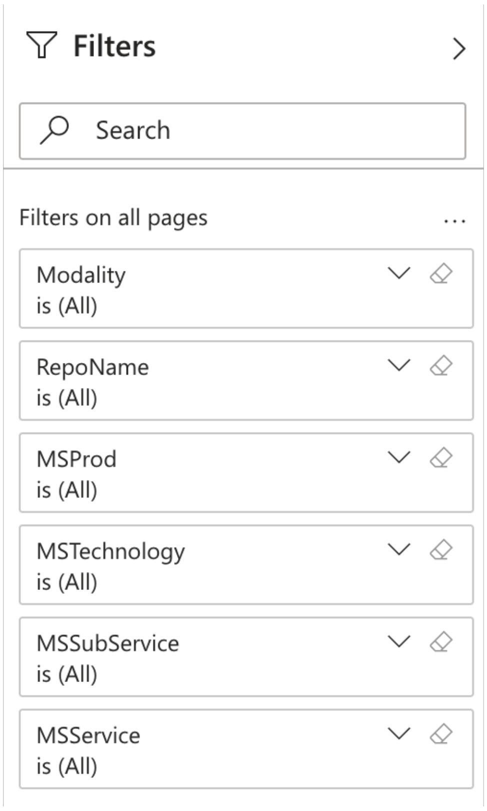
The Content Health composite score is calculated, along with its subscores, at the beginning of each month.

What's covered in the report?

This report covers content from designed repositories managed by the C&L content development teams, Dynamics 365 and Power Platform Docs & Learn, and Microsoft 365 (Commercial and Dev). The list of repositories included in this report can be found in this RepoTracker query (Accessible to C&L FTEs only).

The report includes:

- Only US English (that is, the **en-us** locale) content.
- Purely non-reference repositories. A caveat is for repositories that have both reference and non-reference content; all files will be included for build validations.



What should I do with the score?

> **IMPORTANT**
>
> The Content Health Report and the scores it contains are **not** a report card. They're intended to be used as a signal and to derive insights.

The Content Health score is primarily intended to help managers and content leads understand the health trends in their content spaces. They can use that information to prioritize team projects for content maintenance work, such as addressing build validation messages or doing a freshness pass.

Some individual contributors may find parts of the Content health score useful. Work with your manager to understand your team’s priorities.

Defect Free score

This score calculates the percentage of page views (PVs) to articles/modules with zero defects opened in the public-facing repository. If there is a public/private repo pair, the single repo is used.

The subscore formula

```
1 - (monthly PVs to articles or modules with one or more defects created that month / total monthly PVs)
```

> **NOTE**
>
> The Content Health working group is working closely with the Unified Feedback PM to be sure that when changes to the feedback system roll out, the Content Health score will take those changes into account.

How do I address defects?

From this score, figure out which articles receive defects labeled as "doc-bug" or which modules receive Learning content quality feedback. Then compare page views to determine impact. This may seem reactive as it requires a customer to make a complaint in the first place.

To avoid a negative score, teams should consider proactively auditing their most popular docs and growing a culture of content hygiene.

The score does not measure time to close issues or number of issues closed. In other words, closing issues does not improve the defects score. The only thing that improves the score is if customers don't find and report defects in the first place.

Caveats

- If the GitHub issues feedback control is **enabled** for a repo, the defects score will be calculated. The repo must also use standard labels, including "doc-bug" to get an accurate, comparable score.
- If the GitHub issues feedback control is **disabled**, that content won't include a defects score in the composite.
- Learn has a different feedback mechanism, and since it doesn't have as much granularity, the Defect Free score for Learn is often slightly lower than Products or PnP scores.

Learn how to see the GitHub issues in your content.

Freshness score

The Freshness score calculates the percent of page views reaching content that has an `ms.date` metadata value within the last 365 days. This age range differs from the previous freshness rubrics.

The subscore formula

```
PV count for the articles or modules with a last reviewed date <= 365d / total page monthly PVs
```

How to address freshness?

Find these articles by looking at the date on a live page, or by sorting the LastReviewed date column or the Freshness column in aka.ms/contentperformancedashboard, then compare page views to prioritize the most impactful stale articles. Then refresh these articles using this guidance.

Consider prioritizing your investments if you have lots of older pages. We recommend that you look at how many page views go to an article as compared to others in your content set, and adoption/acquisition content.

Caveats

- If `ms.date` is missing for an article/module, the page date is automatically considered to be > 365 days.
- Freshness data collection began in June 2021, a month before the Content Health Score goes live for all repos. In July, a copy of the data for June will be replicated backward for April and May so that you can see how the reports will work.

Build Validation score

This score represents a *point in time*. The count is calculated at the end of each month from the warnings and suggestions generated during **the most recent build** from the latest pull request on the `live` branch (most common), or whatever the actual published branch name is (less common).

These are builds on our private, internal repositories.

Warnings have twice the weight of **Suggestions** since they are considered a higher severity and they block PRMerger.

The subscore formula

```
1 - ((2 * warning count) + suggestion count ) / total number of articles or units
```

How do I address build validations?

Review the build report for your pull request to see details about the warnings and suggestions in the repo. For more info, see Docs Build validation overview.

Caveats

- **Changes to validation rulesets can greatly affect the build validation score.** (Remember, it isn't a report card.) Over time, new validation rules are introduced or suggestions are converted to warnings as part of our business. These build validation rule changes are made toward the beginning of the month and the Health score data is calculated at the end of the month. This gap leaves a few weeks each month in which to catch and correct validation messages.
- Builds are never forced by this scoring process to avoid unintended consequences.
- For any repo that contains both reference and conceptual content types, reference content will be included in the build validation score. The calculation does not differentiate by ms.topic type. (Reference only repos are not otherwise included in Content Health scoring). We can't calculate the number of articles or units built at a service level when we build. Instead, we use the number of articles per service in our calculation, which includes reference content.
- Build validation scores are set to stay at or above 0, to avoid negative values that may occur because of an article triggering more than 1 warning or suggestion.
- In the special case of `learn-pr` repo mapping to multiple public repos, all scores are from `learn-pr`.

Other FAQs

What if I want to see a subset of a repository?

You can filter by service or subservice for Docs and Patterns and Practice content. Learn isn't tagged with service/subvice, so those filters aren't available.

What do the spaces in the tables mean?

Sometimes due to unavailability of data, a particular score cannot be calculated for a content set. In that case, a space will appear for that content set in the corresponding table. This differentiates from entries that have literal zero, which occurs when the score can be actually calculated and is found to be zero.

For build validation, why are we multiplying the warning count by 2?

Warnings have a higher severity than suggestions and they block PRMerger, so they are a higher priority to fix.

Are include files considered in the topic keys?

No, only actual published articles or Learn units are considered as topic keys. Include files are a part of actual published *topics*. TBD: Build validations?

How do we account for evergreen content in the freshness score?

It depends on the individual team resources and business needs. Teams can either strive for a healthy freshness score by periodically reviewing evergreen content and updating the ms.date, OR they can accept a lower score and understand why it is low. Both are reasonable choices.

Any perspective on CSAs (Customer Solution Areas)?

The first version of this report does not have filters for CSA, however, we are evaluating if a future iteration of this report can have that capacity.

Documentation and reference report

6/24/2021 • 5 minutes to read

The Documentation and Reference report provides metrics at the portfolio-level (as opposed to the Content Performance report, which provides metrics on individual articles). It can be found at https://aka.ms/docrefdashboard.

The report has five, high-level filters on the left-hand sidebar (**Area 1** in the image above):

- **By Repo**: Providing KPIs for individual repos, along with a view for all repositories(`AllRepo`) combined.
- **By SubUrl**: Providing KPIs for articles grouped by `SubUrlLevel1` and `SubUrlLevel2`, taken from an article's URL: `https://docs.microsoft.com/locale/` `suburllevel1` `/` `suburllevel2`. For the URL `https://docs.microsoft.com/en-us/azure/security/benchmarks`, for instance, `SubUrlLevel1` would be "azure" and `SubUrlLevel2` would be "security". This report also provides a data view for a `SubUrlLevel1` that includes all the `SubUrlLevel2` paths.
- **By Product**: Providing KPIs for the `Product` and `Technology` metadata elements. It also provides a data view for all `Technology` for a certain `Product`.
- **By Services**: Providing KPIs for the `Service`, and `SubService` metadata elements. It also provides a data view for all `SubService` for a certain `Service`.
- **By Doc Subset**: The Doc Subset Report (preview) shows the most-often-used KPIs at a content set level, either by URL, repo, prod/service, or custom. This is not an article level report. For more information, see Doc Subset Dashboard.

Area 2: Use the filters to view the monthly data by a dimension (`Repo`, `SubUrl`, `Service`, `SubService`, `Product` and `Technology`).

Area 3: Select the reporting month.

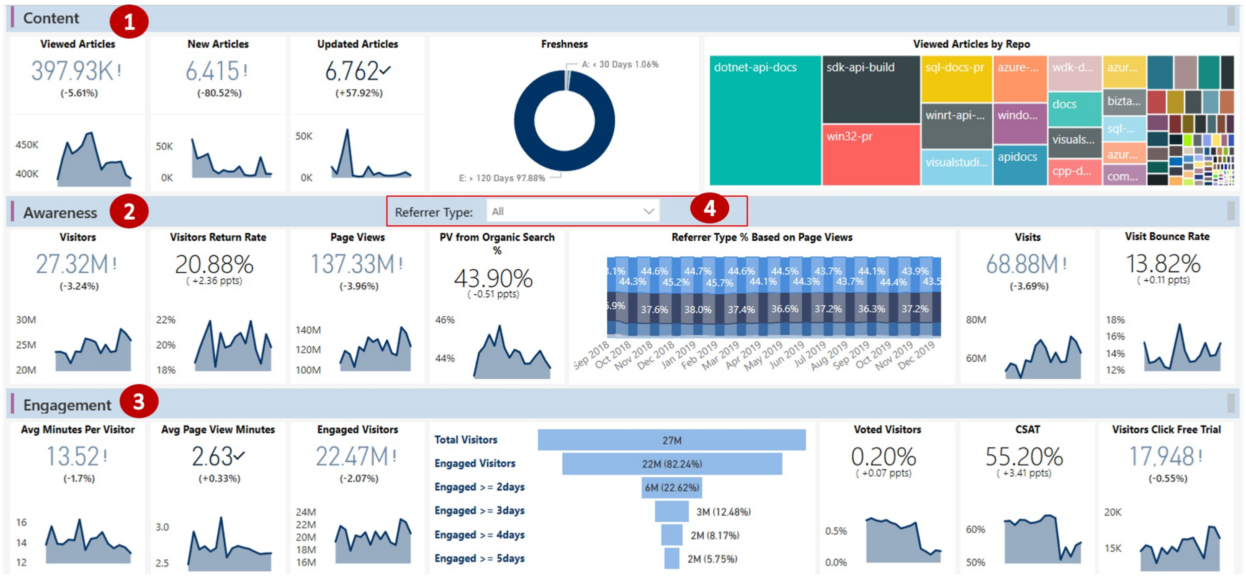
Area 4: you can pivot by `Repo Scope` (**Azure** means only the report the KPI for the repo marked `Product=Azure`, **Azure&NotAzure** means all the repo in repo tracker), and `Area` (**Reference** only reports the KPI for articles which has topic type of `reference`, `managed-reference` and `language-reference`, **Documentation** reports for all other articles, and **Documentation&Reference** is for all articles defined in the repo tracker).

Area 5: you can provide the feedback, clicking the button will launch outlook and create a DevOps task for CIA team triage.

Area 6: you can view the KPI for selected scope, including the **Content**, **Awareness** and **Engagement**.

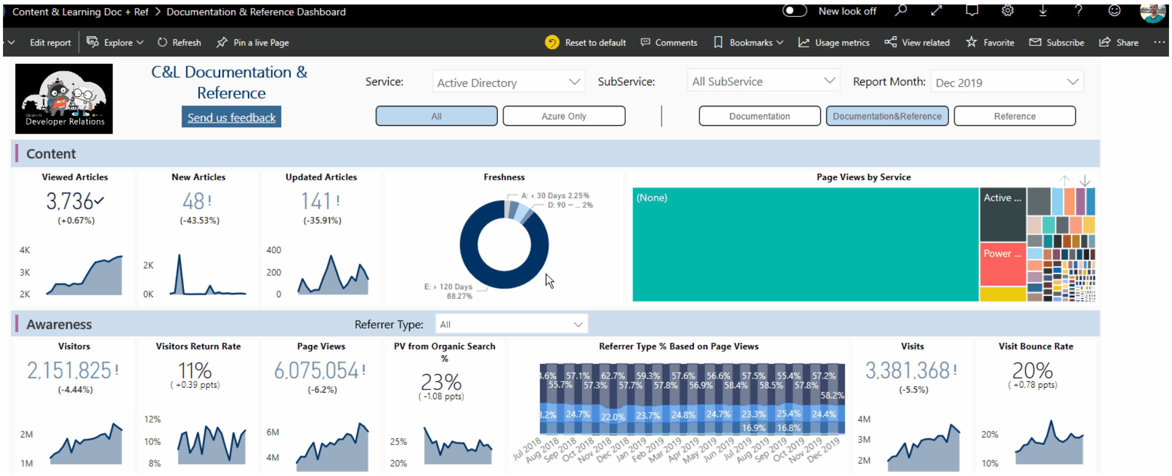
What's the definition for each KPI?

The report provides the Content, Awareness, and Engagement KPIs. For more metric definitions, see Metrics definitions.



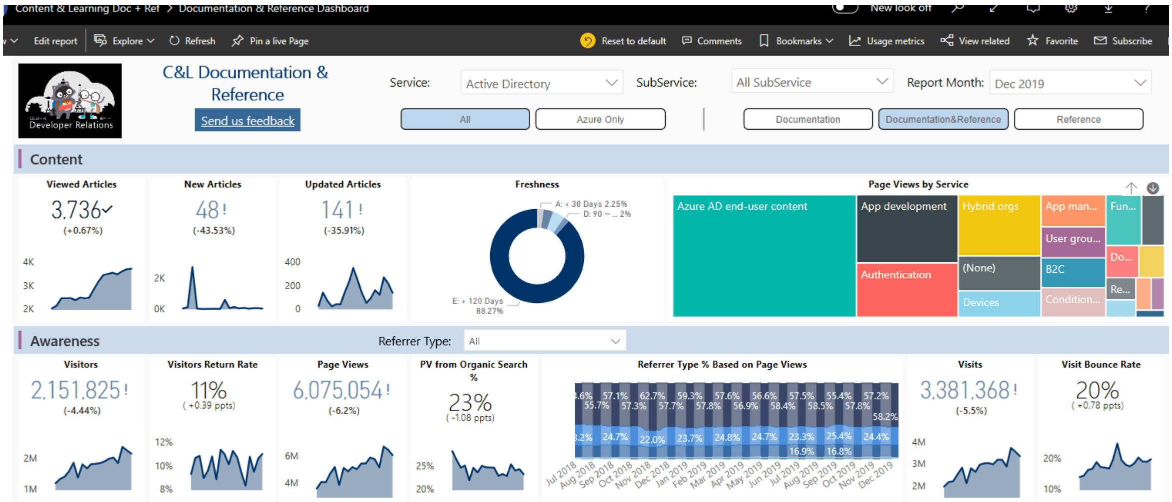
Area 1 (Content)

- **Viewed Articles**: # of articles viewed in the reporting month based on `ContentID`. Non-en-us locales are included with en-us.
- **New Articles**: # of articles first published in the reporting month, based on `ContentID`. Non-en-us locales are included with en-us.
- **Updated Articles**: # of articles updated published in the reporting month, based on `ContentID`, `Last Reviewed Date`. Non-en-us locales are included with en-us.
- **Freshness**: Distribution of the articles based on last updated articles, we count the days gap from `Last Review Date` to the end of reporting month, and categorize them to A,B,C,D,E based on days gap of 0-30, 31-60, 61-90, 91-120, >120 days. Metrics definitions: Freshness.
- **Page View by Dimensions**: A treemap by `PageViews` for level 1 dimension and Level2. For example, in the image below, we can find out the `Active-Directory` is the top 1 `Service` with most Page Views in Dec 2019, and we can Drill down `SubService` and find `Azure Ad end-user content` is the top 1 subservice under `Active-Directory`. The same treemap also shows on other 3 pages.



Area 2 (Awareness)

- **Visitors**: # of unique visitors, based on `VisitorsId`. You can also pivot by `Referrer Type` in **Area 4**. Also see Unique visitors.
- **Visitor Return Rate**: % of visitors in last month returned this month. # of visitors who show in both previous and current reporting month / # of visitors in previous month.
- **Page Views**: # of Page Views, based on `PageViewId`. Also see Metrics definitions: PageViews.
- **PV from Organic Search %**: # of Page View from `Organic Search` / total Page Views. Also see Unique visitors. Also see Metrics definitions: OrganicSearchRate
- **Visits**: # of visit sessions, based on `enrich_session_id`. Metrics definitions: PageViews.
- **Visit Bounce Rate**: we define the visits with only one Page view and `active page view length` < 5 seconds as bounce visits, and we calculate the visit bounce rate as `# of bounce visit/total visits`. Also see Metrics definitions: PageViews.
- **Referrer Type % Based on Page Views**: This is to show the Page View trend for each Referrer type. In the image below, `Microsoft` is the number 1 referrer channel, and `Organic Search` is the second. And there are no big changes in the last 12 months for their Referrer Type %.



Area 3 (Engagement)

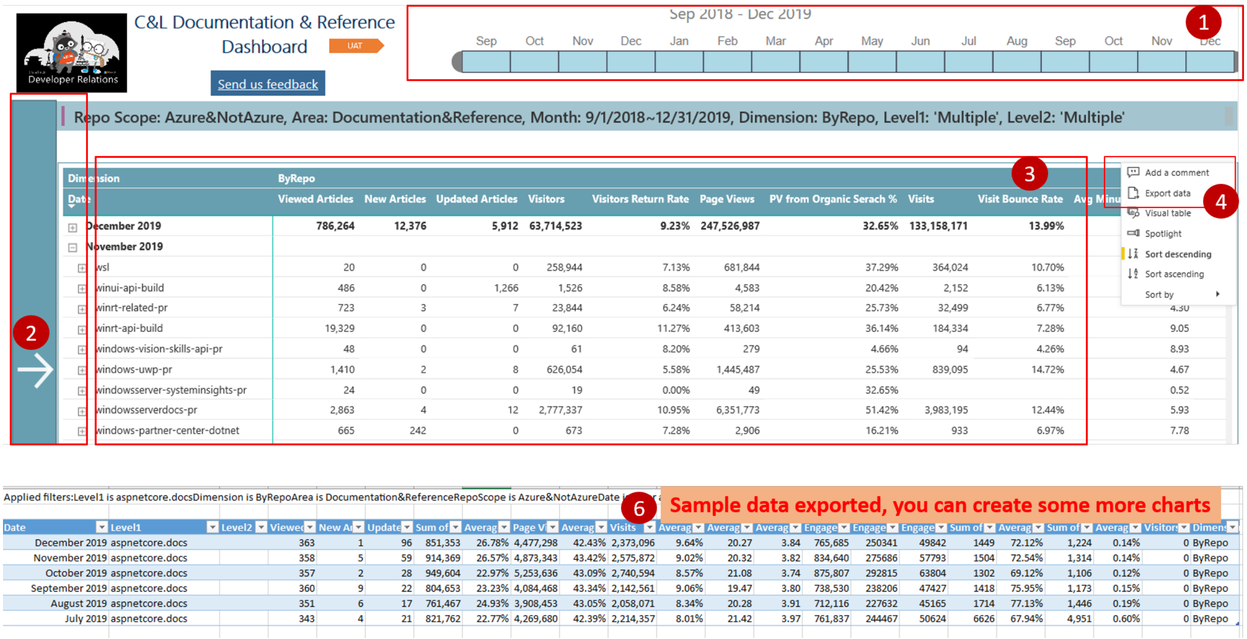
- **Avg Minutes Per Visitors**: The Avg Active Page View Length(minutes) per visitors, we sum the `Active Page View Length` by visitors.
- **Avg Page View Minutes**: The Avg Active Page View Length(minutes) per page views, we sum the `Active Page View Length` by Page View.
- **Engaged Visitors**: We defined the visitors who have valid page view as engaged visitors, the visitors who

only have bounce visits will not be counted. We call the visitors who come to docs in different `N` days on a certain area as a `N Days Engaged Visitors` in reporting month.

- **Engagement Funnel**: To show the funnel for `Total Visitors`, `Engaged Visitors`, `Engaged Visitors >=2 Days`, `Engaged Visitors >=3 Days`, `Engaged Visitors >=4 Days`, `Engaged Visitors >=5 Days`.
- **Voted Visitors rate**: The # of visitors who rated / total visitors.
- **CSAT**: The # of Helpful ratings / total Ratings
- **Visitors Click Free Trial**: # of visitors who clicked `Azure Free Trial`.

How to export data for customized reporting?

The dashboard provides a way to export the same data we are using the Dashboard from DB, you can choose time range in **Area 1**, and choose **Repo Scope**, **Areas** and **Dimensions** in **Area 2**. After reviewing the data in your selected scope in **Area 3**, you can **Export Data** in **Area 4** after clicking `More Options` (... icon).



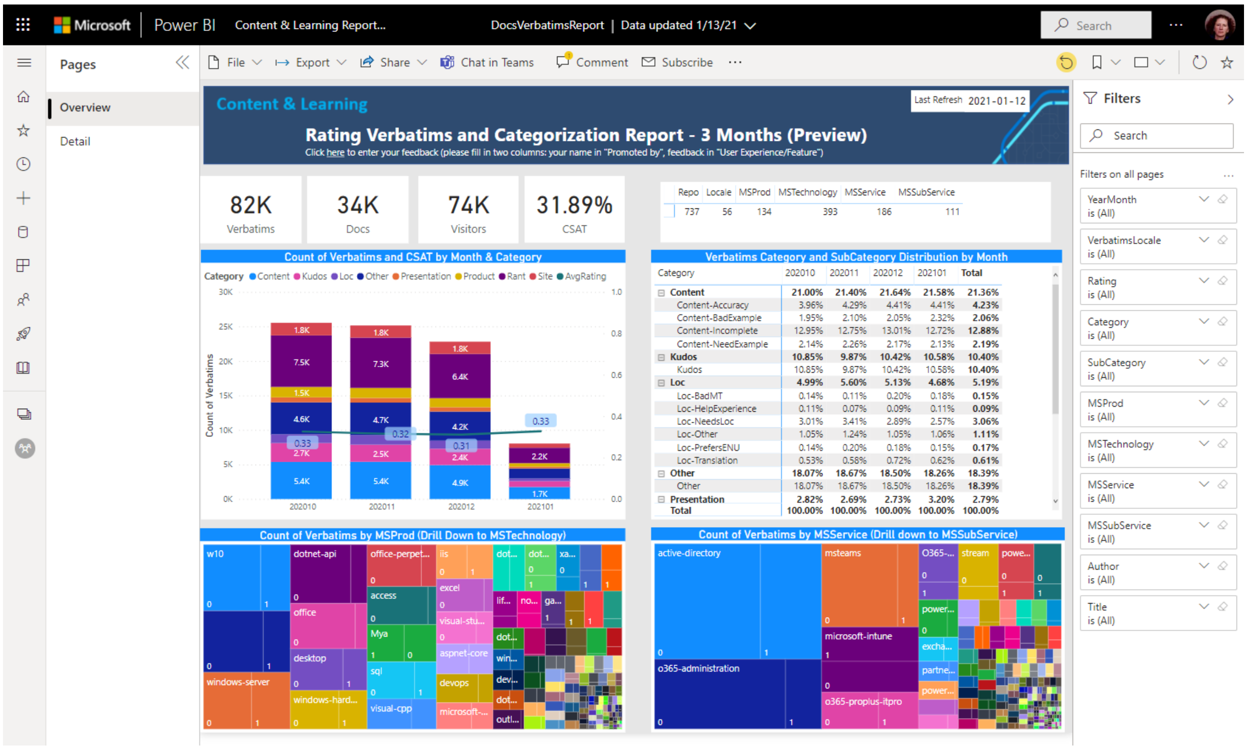
Area 6 provides a sample of data that was exported. You can create additional charts based on the CSV file.

> **WARNING**
>
> If you filter the Dimension field by URL, Product, or Service, and if you do not want to further filter by a Level2 filter (SubURL, Technology, or SubService), then we recommend that you select the "All" filter in Level2 (AllSubUrlLevel2, All Technology, or AllSubService).
>
> If you don't select the "All" filter in Level2, then if you collapse the rows of the table, the roll-up row at the URL, Product, or Service level will be inaccurate. It will sum the "All" Level2 row with all the Level2 rows, effectively doubling the counts.

Verbatims report (Preview)

3/5/2021 • 2 minutes to read

We're rolling out a preview version of a new Verbatims report to replace the previous report. The data in the previous version hadn't updated since September 2020, and we don't own it.



The new Verbatims report https://aka.ms/DocsVerbatimsReport has two pages:

- **Overview** displays summary data for the past three full months, plus the current month to date. It shows total verbatims, number of docs, and the average rating (the 0 or 100 "Is this page helpful?" rating). It segments count of verbatims in a few different ways:
  - By MSProd or MSService.
  - By category and subcategory. These subcategories are assigned using a machine learning text classification tool. They may or may not be accurate. We're planning more training for the tools.
- **Detail** shows the actual comments left by customers via the rating control. You can use these comments to improve your docset. See How to view and act on customer verbatim comments.

Overview page

KPIs

The KPI numbers across the top:

- **Verbatims**: Total number of verbatims in the filtered or unfiltered page.
- **Docs**: Total number of articles with verbatims in the filtered or unfiltered page.
- **Visitors**: Total number of visitors (MAU) with verbatims in the filtered or unfiltered page.
- **CSAT**: The average rating that these articles have received. Note that:
  - Customers can't provide a verbatim without selecting a rating.
  - This average is lower than the overall rating average. Possible explanation: Customers who add verbatims are on average less satisfied.

Column + line chart: "Count of Verbatims & Average Rating by Month & Area"

- Each column in this chart represents the total number of verbatims for a month: three full months, plus the fraction of the current month to date. Month names are year plus month: yyyymm. For example, 202011 is November 2020.
- The legend divides the verbatims by category. The eight areas are a roll-up of 20+ subcategories. The Content category is of most interest to authors.
- The line shows the average rating by month.

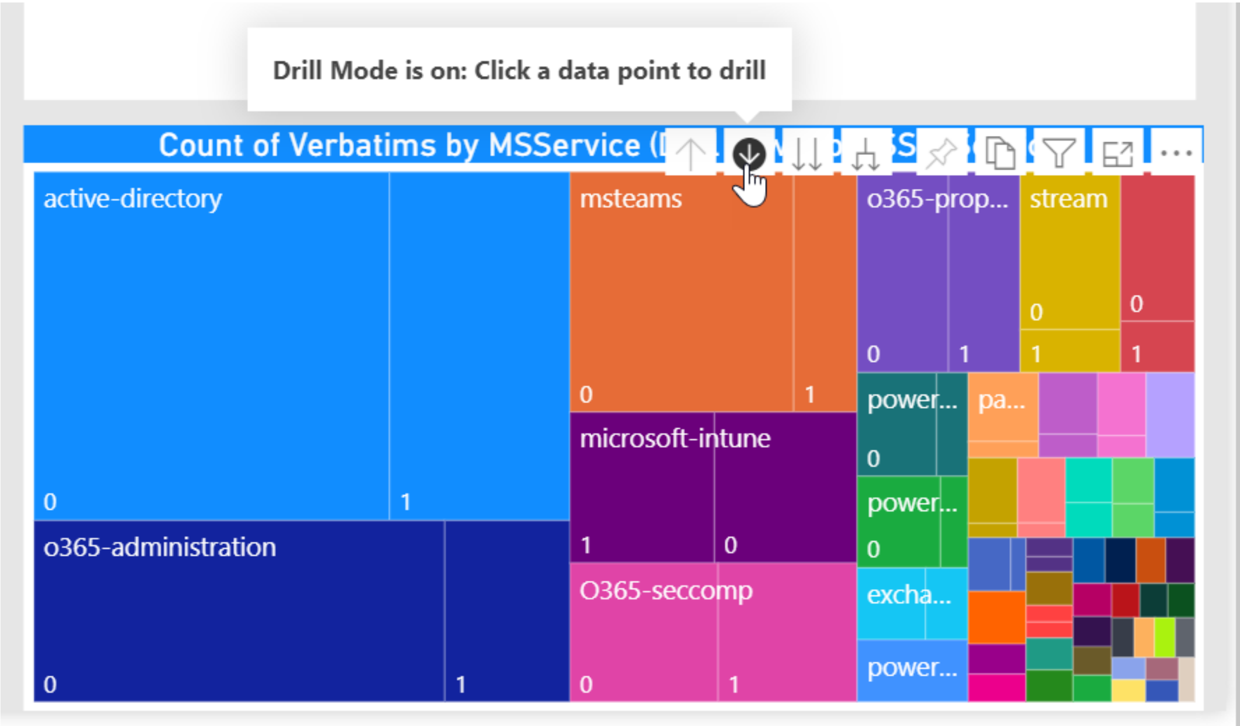
Matrix: "Verbatims Area & Category Distribution by Month"

This matrix shows the same data as the column + line chart, but it lets you drill down the the subcategories in each category.

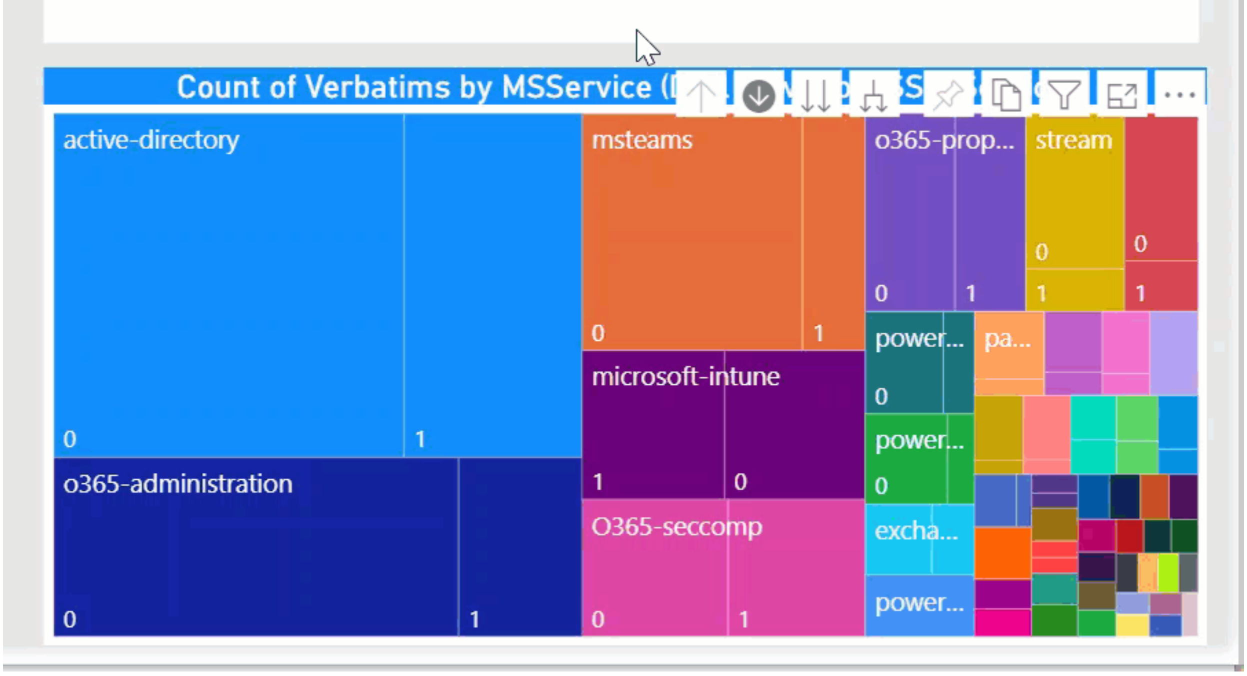
Treemaps for MSProd and MSService

The two treemaps at the bottom break the verbatims out by product on the left or service on the right. You can drill down in either of them to see the technologies/subservices that make up a product/service.

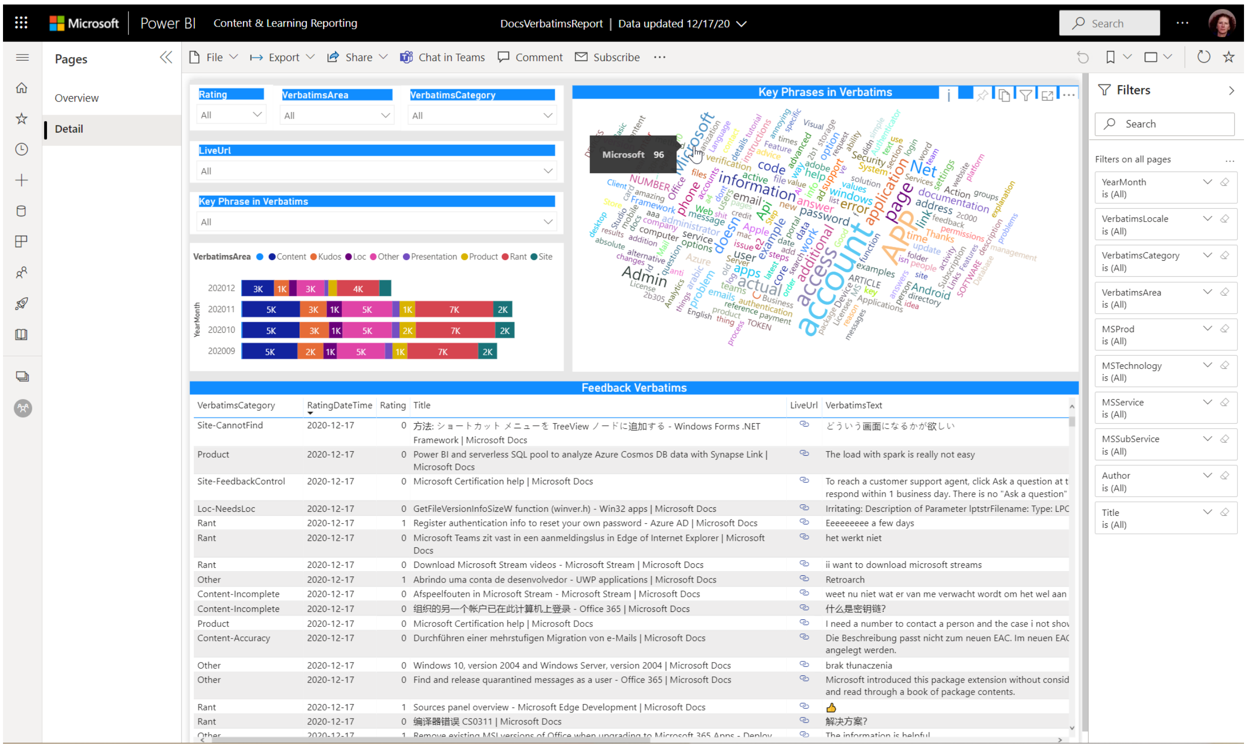
Hover over the treemap and select the down arrow to **turn on Drill down**.



Now select one of the products/services to see its technologies/subservices.



Detail page



Word cloud

The word cloud pulls out key phrases from the verbatims. Size indicates the number of times the word comes up -- the larger, the more frequent. Hover over a word to see the actual number. Selecting a word filters the other visuals on the page, including the table.

Feedback verbatims table

The table lists all the verbatims for the articles that match the filters on the page. If the verbatim isn't in English, scroll right in the table to see the autogenerated English translation.

Next steps

This Verbatims report is in preview. If you have feedback, please enter it in this Verbatims Report Feedback Excel file.

- Learn How to view and act on customer verbatim comments

GitHub issue and pull-request report

3/5/2021 • 3 minutes to read

The GitHub issues and pull-request report provides both portfolio-level metrics - on a repo, department, product, or service - and details on specific Issues.

Pages

- **ReadMe**: A collection of important details and known issues for the current version of the report.
- **GitHubIssueSummary**: A collection of high-level power metrics and visualizations representing the health of GitHub issues within the scope defined by filter settings. This page is currently scoped to public repos within DevRel: Content & Learning that have GitHub issues enabled.
- **GitHubIssueDetails**: Provides issue-level details within the scope defined by filter settings. This page is currently scoped to public repos within DevRel: Content & Learning that have GitHub issues enabled.
- **GitHubPRSummary**: A collection of high-level power metrics and visualizations representing the health of pull-requests within the scope defined by filter settings. This page is currently scoped to all repos within DevRel: Content & Learning.
- **GitHubPRDetails**: Provides PR-level details within the scope defined by filter settings. This page is currently scoped to all repos within DevRel: Content & Learning.
- **WorkArea_**: These pages are testbeds for new visualizations and report functionality.

Filters

Pages: GitHubIssueSummary, GitHubIssueDetails

FILTER	REPORT LOCATION	DESCRIPTION
State	Summary/Details	Open and/or Closed issue states
IssueType	Summary/Details	Type of issue. Possible values are defined by labels applied during the triage process.
RepositoryName	Summary/Details	Name of one or more GitHub repositories
YearMonth	Summary/Details	Years and Months included in the measures.
Priority	Summary/Details	Priority of issues. Possible values are defined by labels applied by the GitHub automation app and are based on page view for the associated topic.
OKRRepo?	Summary/Details	Repositories included in the aggregate Content Health KR measure surfaced in the Monthly business review.
OrganizationLogin	Summary/Details	GitHub Organization (for example, MicrosoftDocs, DotNet)

FILTER	REPORT LOCATION	DESCRIPTION
MSProd	Summary/Details	ms.product value of the topic the issue was filed against
MSTechnology	Summary/Details	ms.technology value of the topic the issue was filed against
MSService	Summary/Details	ms.service value of the topic the issue was filed against
MSSubService	Summary/Details	ms.subservice value of the topic the issue was filed against
AssigneeOrg	Summary/Details	Assigned issues by Organization.
AssigneeDepartment	Summary/Details	Assigned Issues by Department
AssigneeManagerAlias	Summary/Details	Assigned issues by manager assignee(s) report to
AssigneeMSAlias	Summary/Details	Assigned issues by MS alias
GitAlias	Summary/Details	Assigned issues by Git alias
ClosedMonth	Details	Issues closed in each month(s)
SubmitterType	Details	Internal or External (non-MSFT) submitter
Submitter	Details	Individual that created the issue
MSFTSubmitterOrg	Details	MSFT submissions by Organization.
MSFTSubmitterDepartment	Details	MSFT submissions by Department
MSFTSubmitterManager	Details	MSFT manager submitter(s) report to
MSFTSubmitterAlias	Details	Submissions by MS alias
Labels	Details	By issue label(s)
ContentId	Details	By the ContentId defined in a topic's metadata

Pages: GitHubPRSummary, GitHubPRDetails

FILTER	LOCATION	DESCRIPTION
YearMonth	Summary/Details	Years and Months included in the measures.

FILTER	LOCATION	DESCRIPTION
OKRRepo?	Summary/Details	Repositories included in the aggregate Content Health KR measure surfaced in the Monthly business review.
OrganizationLogin	Summary/Details	GitHub Organization (for example, MicrosoftDocs, DotNet)
RepositoryName	Summary/Details	Name of one or more GitHub repositories
MSProd	Summary/Details	ms.product value of the topic the PR was opened against
MSTechnology	Summary/Details	ms.technology value of the topic the PR was opened against
MSService	Summary/Details	ms.service value of the topic the PR was opened against
MSSubService	Summary/Details	ms.subservice value of the topic the PR was opened against
AssigneeOrg	Summary/Details	Assigned pull-request by Organization.
AssigneeDepartment	Summary/Details	Assigned pull-request by Department
AssigneeManagerAlias	Summary/Details	Assigned pull-request by manager assignee(s) report to
AssigneeMSAlias	Summary/Details	Assigned pull-request by MS alias
GitAlias	Summary/Details	Assigned pull-request by Git alias
ClosedMonth	Details	Issues closed in each month(s)
OpenDays	Details	How long the PR has been, or was, open
SubmitterType	Details	Internal or External (non-MSFT) submitter
SubmitterGitAlias	Details	Submission(s) by GitHub alias
SubmitterDepartment	Details	MSFT submissions by Department
SubmitterManagerAlias	Details	MSFT manager submitter(s) report to
SubmitterMSAlias	Details	MSFT submissions by alias

Providing feedback

Similar to other released content reports, feedback can be provided via the Content Data Advocacy Group dashboard.

Next steps

- Learn the GitHub issues process for Azure authors.
- Receive GitHub issue and public pull request reports.

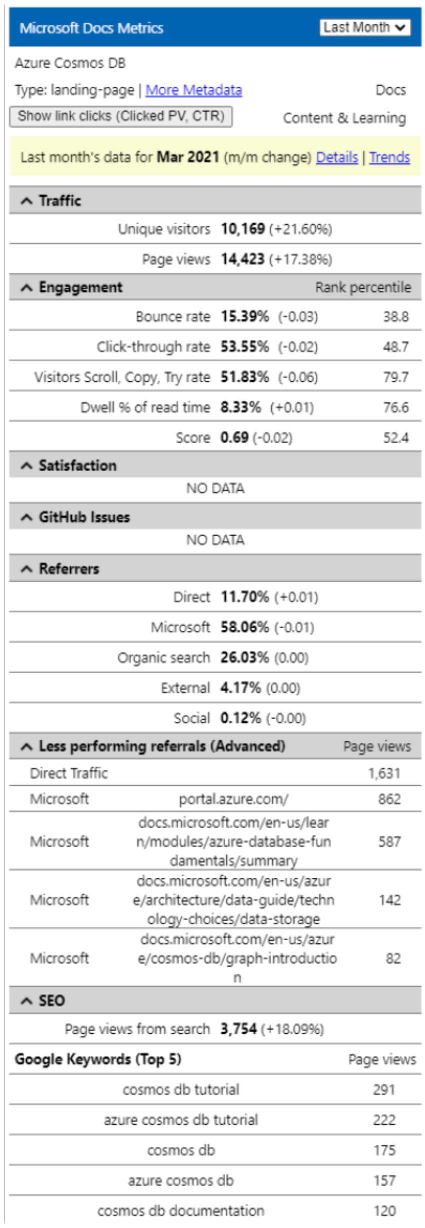
Get page-level metrics in your browser: Microsoft Docs Metrics

7/7/2021 • 12 minutes to read

The Microsoft Docs Metrics (MDM) tool that allows you to get high-level metrics for a page on docs.microsoft.com right from your browser. It's convenient and useful when you want easy access to traffic, engagement, and referrer data, or you want to see verbatim comments for a specific page.

> **WARNING**
>
> SkyRay is no longer supported and has reached end-of-life. MDM versions installed prior to 4/21 are no longer working. Reinstall latest version.

For content not on docs.microsoft.com, but on approved tech sites, the MDM tool opens directly in Azure Data Explorer instead. Those sites include: azure.microsoft.com, blogs.msdn.microsoft.com, blogs.technet.microsoft.com, cloudblogs.microsoft.com , channel9.msdn.com , code.msdn.microsoft.com , code.visualstudio.com , devblogs.microsoft.com , developer.microsoft.com, gallery.technet.microsoft.com, msdn.microsoft.com , social.msdn.microsoft.com, social.technet.microsoft.com, techcommunity.microsoft.com, technet.microsoft.com, techprofile.microsoft.com , www.azure.cn, docs.azure.cn , and https://www.microsoft.com



Edge: install & login

Microsoft Docs Metrics on Edge is GA (offering auto-updates).

Install the tool:

1. Get the latest extension from version: 3.0.2
   - You must be **on corpnet** and **signed into the browser with your MSFT alias** to see the **Extensions lab** page.
   - You must scroll down to locate the MDM extension because the **Extensions lab** page is not searchable.
2. Enable auto-update from extensions lab:
   a. In your browser, open the Extension Management tab.
   b. Turn on **Auto install extensions from Extension Lab** to automatically get newer versions of the extension when they are released.

Log in:

1. In your browser's settings, **allow pop-ups** from `https://skyray-api-public.azurefd.net:443` so you can log in using these Edge steps.
2. Open the page on https://docs.microsoft.com for which you want data.

3. In the browswer toolbar, click the **Microsoft Doc Metrics** extension icon in the top-right corner. The extension opens on the right side of the page.
   Tip: Pin the extension if it isn't already in the browser toolbar for easy access.

Chrome: install & login

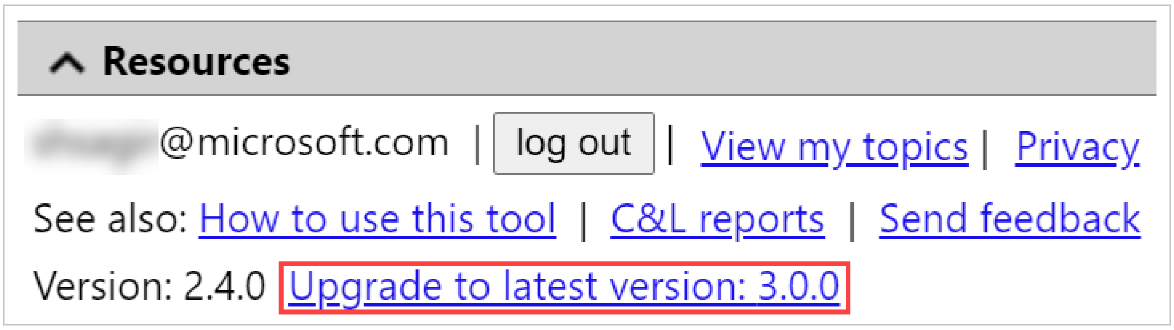
> **WARNING**
> The MDM extension does not auto-update on Chrome. Chrome users are responsible for keeping MDM up to date. The latest version is shown at the bottom of the tool window in your browser at all times.

Install the tool:

1. If upgrading from earlier MDM version, uninstall the older version first.
   - On your computer, open **Chrome**.
   - In the address bar, enter: `chrome://extensions/`.
   - On the Microsoft Docs Metrics card, click **Remove**.
   - Confirm by clicking **Remove**.
2. Download the MDM extension zip file.
3. Unzip the file locally.
4. Back in your browser, return to the Extensions tab. (In the address bar, enter: `chrome://extensions/`.)
5. Turn on **Developer mode** slider in the upper right.
6. Select **Load unpacked**.
7. In the dialog box, select the folder for the unzipped extension.
8. Once loaded, make sure the blue toggle shows that the extension is enabled. *The extension is installed and appears in the list.*

Log in:

1. In your browser's settings, **allow pop-ups** from `https://skyray-api-public.azurefd.net:443` so you can log in using these Chrome instructions.
2. Open the page on https://docs.microsoft.com for which you want data.
3. In the browswer toolbar, click the **Microsoft Doc Metrics** extension icon in the top-right corner. The extension opens on the right side of the page.
   Tip: Pin the extension if it isn't already in the browser toolbar for easy access.



Troubleshooting sign-ons

If you aren't automatically signed in, try the following fixes:

- If the window doesn't appear, check your pop-up blocker. Allow pop-ups from `https://skyray-api-public.azurefd.net:443`.
- Close your browser completely and restart.

- Make sure the latest Windows updates are installed and reboot your machine.
- Manually enter your Microsoft domain account in the Login window.
- If you are a **Google Chrome user** and signed into your browser, go to **Chrome > Preferences** in the menu. In the **You and Google** section, log out.

If these don't unblock you, email Kelly Pittman -AND- Yiwen Zhang.

Permissions

1. Ensure you have permission to view MDM metrics in the Azure Data Explorer dashboard. If not, request **Read-only** permission to the CL Dashboard User Group. If you cannot get access, please email Kelly Pittman or Yiwen Zhang.

2. In the Operations section, you'll find a link to the source page in GitHub. To see source files in GitHub, you'll need to give your GitHub account access to the private MicrosoftDocs organization. Certain internal staff may not be authorized for certain Kusto clusters at this time. If getting the above listed permissions doesn't suffice, please reach out using the 'Report Feedback' button at the bottom of the tool.

What data is in MDM?

Time range choices

Use the drop-down in the upper right of the tool to choose the time period for the data.

> **TIP**
> It can take up to a week for the previous month's data to become available. For example, you might not see December data until after January 5th.

OPTION	DESCRIPTION
Last month	The latest complete month of data
Last week	The latest complete week of data

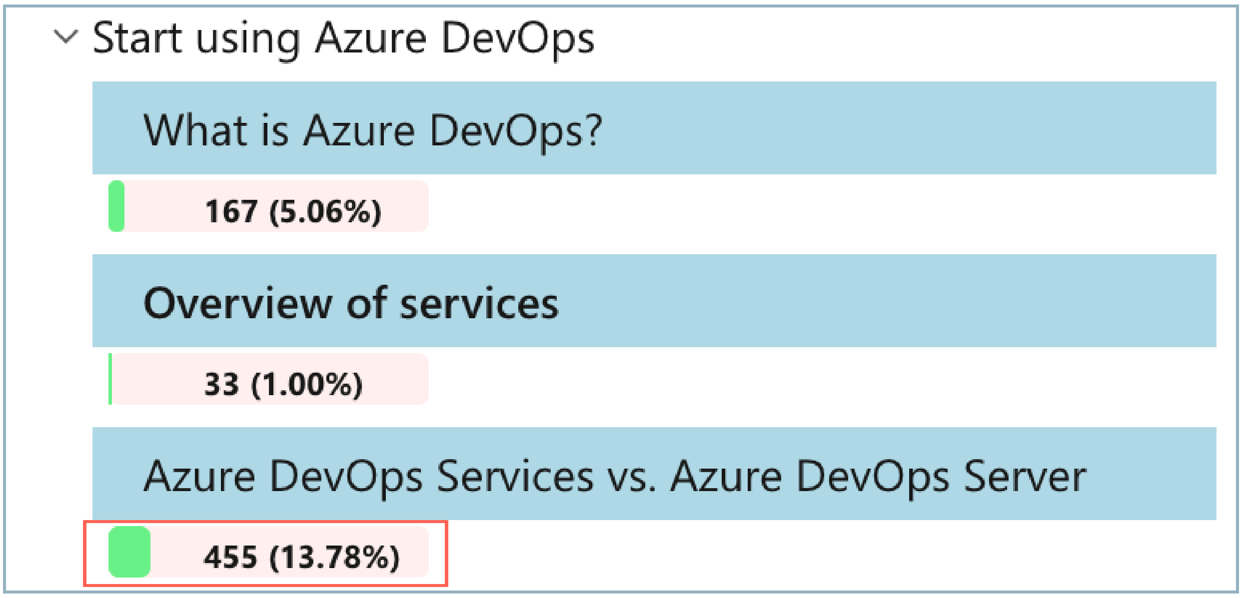
Outgoing click counts

To see what outgoing links were clicked (click-through), select the button **Show link clicks (Clicked PV, CTR)**. When you select this button, all the links clicked during the selected time range appear on the page.

The data appears on the page with this syntax `Clicked PV (CTR %)`:

VALUE	DESCRIPTION
Clicked PV	This value shows the # of page views with outgoing traffic to the URL behind the link, and **not the actual clicks on the link**. If you have 3 links on that page pointing to the same URL, their counts will be aggregated.
CTR	The percentage of page views that visited the linked URL (click-through rate).

You'll also see how many page views a URL got by the visual bar.



TIP

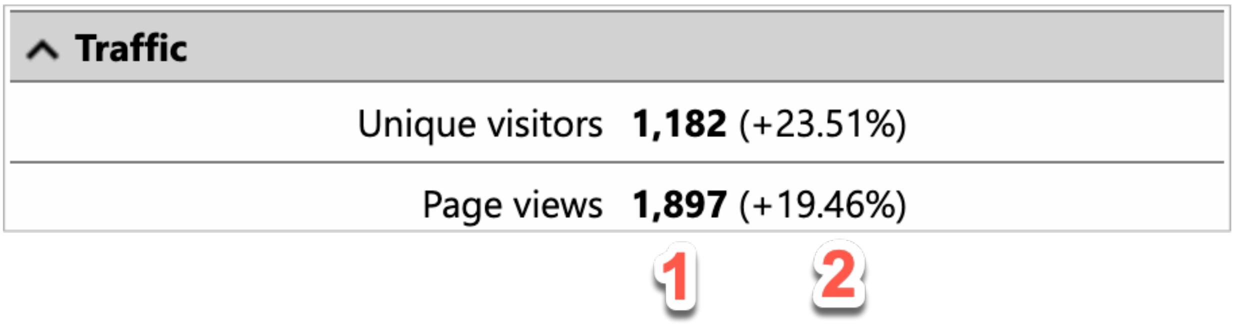
If you can't see your page well, click the MDM icon in the toolbar again to hide the extension while keeping the link counts.

Traffic metrics

This section provides page traffic information for the selected period.

For each metric, you can review:

1. The overall volume of **page views** and **unique visitors**
2. The **growth rate** (month-over-month or week-over-week)



Engagement metrics

This section provides content metrics on how readers engaged with this page. Use these metrics to understand:

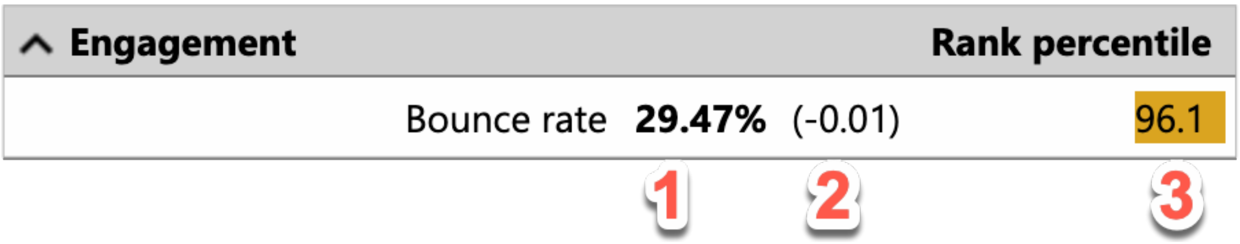
VALUE	GROUP	DESCRIPTION	RANK PERCENTILE
Bounce rate	All	Left too quickly & bounced in <5 seconds	The lower, the better
Click-through rate	All	Clicked through to another page on our site	The higher, the better
Visitors Scroll, Copy, Try rate	Docs	Engaged (copy, try, scroll) with page content	The higher, the better

VALUE	GROUP	DESCRIPTION	RANK PERCENTILE
Dwell % of read time	All	Dwelt on the page for a while	The higher, the better
Monthly completion rate	Learn	Completed the learning path	The higher, the better
Score	Docs	Roll-up of the engagement metrics into a single number	The higher, the better

In the **Rank percentile** column, you can see the relative performance of this page against other pages of the same type group across all of https://docs.microsoft.com. For example, all tutorials use the same primary metrics. If you see multiple ranks, it may mean that this page appears for two groups, such as in a C&L doc set *and* a BAG doc set.

The **Score** value is a roll-up of the primary engagement metrics for a given page type (see formulas) into a single number. Identifying pages that are performing lower than others can help you spot potential issues. and troubleshoot your page's performance.

> **TIP**
> If **Bounce** is high, check the **Underperforming referrers** and run the **Referral Analysis query**. See the Referrer section.



As shown in the graphic, the data is presented as:

1. Rate of PVs exhibiting this behavior.
2. The growth rate month-over-month or week-over-week.
3. The percentile ranking of this page against other pages like it, with highlighting wherever the article is performing significantly worse that its peers in this area.

Satisfaction (CSAT) & Verbatim feedback

Not applicable to **Learn** content.

Customer satisfaction (**CSAT**), is measured by the % of responses with helpful votes divided by total votes. This section is particularly helpful when the page has received many votes. If you only get a few, it might not be as insightful. It shows "NO DATA" if there aren't any CSAT responses on the page.

You can also select the **Details** link to view an Azure Data Explorer dashboard filtered to the article you were looking at, and the time period you had selected: one week or one month. You can update the dates in the query to see more verbatim.

Use the verbatim widgets to view see anonymous feedback that readers submit after voting on the "Is this page helpful?" widget. Learn how to handle verbatim feedback.

GitHub Issues

Not applicable to **Learn** content.

When readers find a typo, doc issue, or have a question, they can use their GitHub account to open GitHub issues on pages on docs.microsoft.com. In the tool, you can see how many issues were opened during the selected time period plus how many issues are open at the moment. Addressing open issues is a content priority.

Referrers: Share of traffic

Referral traffic is the incoming traffic to your page. This section helps you understand:

- High direct traffic is an indication of a problem, especially when bounce is also high.
- Low search traffic is an indication of potential discoverability and relevance issues. Look at keywords if this happens and review SEO best practices.

REFERRER	DESCRIPTION
Direct	% of PV without a referring URL. Page view came directly via a link in the product, a bookmark, an aka.ms referral, contextual help/F1 links, typing in a URL directly, or an email campaign. Tips to lower direct traffic
Microsoft	PVs from any Microsoft domain, including docs. Having backlinks to your page helps with SEO.
Organic search	PVs directly from Google, Bing, and other search engines. The higher the better as it is an indicator of discoverability. This should represent the largest share of referrals. SEO best practices
External	PVs coming through links from other domains outside Microsoft.
Social	PVs from links on social media (Facebook, Twitter, and other social networks). Backlinks to your content boosts SEO.

You can also select the **Details** link to view an Azure Data Explorer dashboard filtered to the article you were looking at, and the time period you had selected: one week or one month. You can update the dates in the query to see more referrals.

Use the referrals widgets to dive into identify **underperforming** and **top performing** referrals. Understand how your incoming traffic is leading to highly engaged visitors or poorly engaged visitors can help your page strategy. You want low bounce. Identifying incoming referrers that need to be improved can help you achieve lower bounce.

Referrers: Underperformers

Most simply put, **underperforming referrals** are the biggest sources of less-engaged page views (PVs). Fixing popular sources that bring poorly engaged visitors can pay off. If your page is higher performing overall (see the `Score` rank in the **Engagement** section), the information here might not provide as much value to you. Prioritizing your efforts on high volume referrers that bring in lower engagement gives you the best bang for your buck.

Each referrer receives its own engagement score (PrimaryKPIScore) using the formula for the topic type. Of the referrers with the lowest engagement scores (bottom half), the five with the highest PVs are displayed as potential opportunities. To better understand what might be contributing to a score, you can click the **Referral**

analysis query (described previously). Learn more about how to handle referral issues.

Example: You might find, for example, that 1500 page views a month come from a single referrer. You can then run the **referral analysis** query where you notice that most page views end in bounce. Now, you can fix the link or remove the link from the referrer to this page.

> **WARNING**
> **KNOWN ISSUE**: Underperforming referrer data may not load if the page has a big volume of PVs.

SEO: Keywords

In the SEO section, you see the **number of page views from search** and the growth rate from last month or week.

You also see some **Google keyword data**. 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. The keywords in the tool were used to reach the page, but there are likely more opportunities to optimize your page for search.

You only get a handful of keywords with this tool. Get more with these resources:

- For a more complete list, read Find keywords for your page.
- Learn how to research, target, and track keywords.

> **WARNING**
> **KNOWN ISSUE**: There may be a 24-hour window where Google keyword data isn't available during the monthly refresh job.

Operations metrics

In this operations section, you can open the private version of this file in GitHub using the **View GitHub Content** link. You'll need Docs publishing access from Github](contribute-get-started-setup-github.md) to open the file.

You can also see:

- Freshness: "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.
- **Last Reviewed** date and **Last Commit** in GitHub.
- Score of last GitHub change as **lines changed**.

MDM change log

3.0.2

Version 3.0.2, released on **2021-07-01**, and contains the following major fixes

- Update **Send feedback** link
- Remove unused endpoint URLs

3.0.1

Version 3.0.1, released on **2021-04-22**, and contains the following major fixes:

- Fix a potential security issue for token exposure.

- Update backend API service endpoint which is secured by PME tenant.
- Remove '(beta)' characters for new release version.
- Add privacy statement url in extension UI.
- Signed extension for Edge version (not signed yet for Chrome).

2.202102

Version 2.202102, released on **2021-02-28**, and contains the following major fixes:

- Switched to content performance V3 data to align the ranking and latest engagement metrics (**CopyTryScroll rate**) used in the Content Performance Dashboard
- Added support for *docs.microsoft.com/archive* (*Other Docs Page*) not in the scope of the Content Performance Dashboard for all metrics except ranking
- Added support for Tech sites to view common page KPIs in a new ADX dashboard tab
- Removed drilldown links to **RatingVerbatim** (under Satisfaction) and **ReferralDrilldown** (under Referrers) links as the data is available in the dashboard opened using the **Details** link
- Typos and label clarifications fixed

2.202012

Version 2.202012, released on **2020-12-10**, and contains the following major fixes:

- Improved the outgoing links match % with some work arounds the dynamic links on tabs and cards links
- Improved the least performing referral results, only add the enrich_url_page_query_string to Direct Traffic referrals
- Added notification feature to inform users on MDM if new version available, and users can also find the local version and latest version on server, follow the links to install the latest version
- Replaced Trend and details in MDM to match content perf dashboard.
- Query data from Kusto when hitting empty cache, this is to fix the Google search data shows empty when it's refreshed on the 4rd day while other content performance data ready on 2nd day
- Improved the CSS style when embedding the MDM to work around the TOC overlayer problem
- Typos and label clarifications fixed

Note: **Version 1.9** will stop working in late December due to CGA changes. Updates as soon as possible.

1.9

Version 1.9, released on **2020-10-22**, and contains unspecified UAT fixes.

Contributor Insights

6/24/2021 • 5 minutes to read

The Contributor Insights Dashboard helps you understand the types of contributors who are publishing their contributions to the docs.microsoft.com platform. To see the Contributor Insights Dashboard, visit https://aka.ms/clcontributordash.

Our active community across Microsoft is a diverse group of contributors working together across the company and uses input coming from outside Microsoft as well. This dashboard categorizes our community as members into one of three groups:

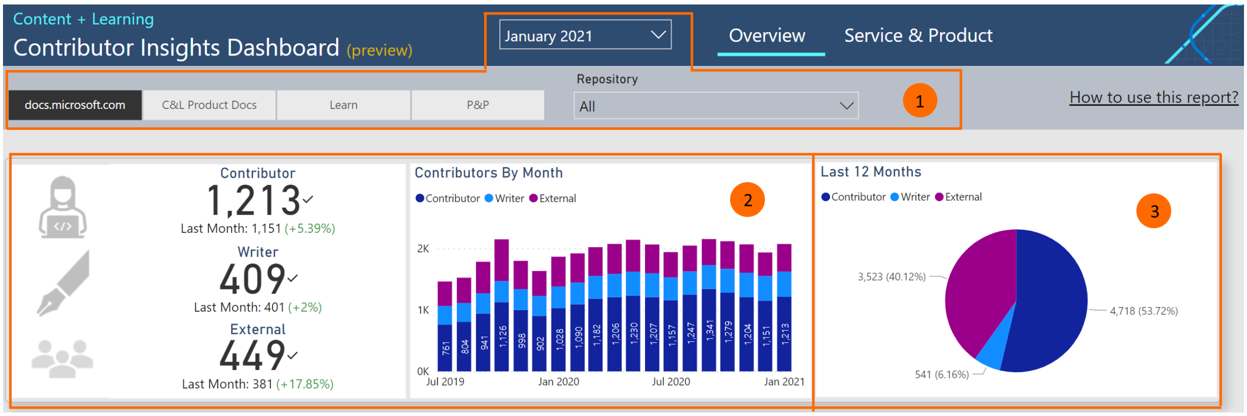
- **Writer** - a community member for whom content development is their primary job responsibility (including vendors and publishing staff)
- **Part Time Contributor** - a community member for whom content development is not their main job role (for example product team Program Managers, Engineers, etc.; includes vendors)
- **External** - a community member who is outside Microsoft, possibly a customer, a partner, or a developer

> **IMPORTANT**
> A **Contributor** is defined as anyone who has authored a commit in the live branch of a docs.microsoft.com repository, e.g. Contributors have had their work *published*.

Understand the overview page

The Overview page shows contributor information for all content repositories for the most recent completed month. You can change the month of the report, or use the filters to narrow down contributor information to only certain repositories.

For more information on the areas highlighted in this image, read the corresponding sections below.



Choose your scope - Area 1

The overview page offers a few selections for how to scope the rest of the page.

Month - Change the month of the report to show contributor participation in previous months.

> **IMPORTANT**
>
> A **Contributor** is counted in a given month if their commits joined the live branch of a docs.microsoft.com repository in that month, regardless of when the commits were authored.

Repository selection - use the buttons or the Repository menu to set the scope of the report. You can either use the buttons to select groups of repositories, or the item selection to choose a single repo.

The four buttons filter the report to a view of groups of repositories.

- `docs.microsoft.com` - Choose to see information for all repositories for docs.microsoft.com
- `C&L Product Docs` - Choose to see information for only technical docs for C&L Products
- `Learn` - Choose to see information for only Learn repositories
- `P&P` - Choose to see information for only P&P Repositories

Repositories are categorized into one of these groups using Repo Tracker, which categorizes and tracks active repositories. Please contact Ryan Thompson with questions about repo tracker at this time.

Mirrored Repositories - For repos in a private-public pair, *only the private repo is listed* in the drop-down list of repos. The public repo is not listed because content is not published from public repositories. Commits from external users are mirrored to the private repo before they're published, so they appear in the dashboard when the private repo is selected.

> **TIP**
>
> The `Repository` selection is affected by the buttons, so if you're looking for a repository but can't find it, try changing your selection back to `docs.microsoft.com`.

Contributor count by Month - Area 2

The contributor counts shows the number of contributors of each type for the selected filter.

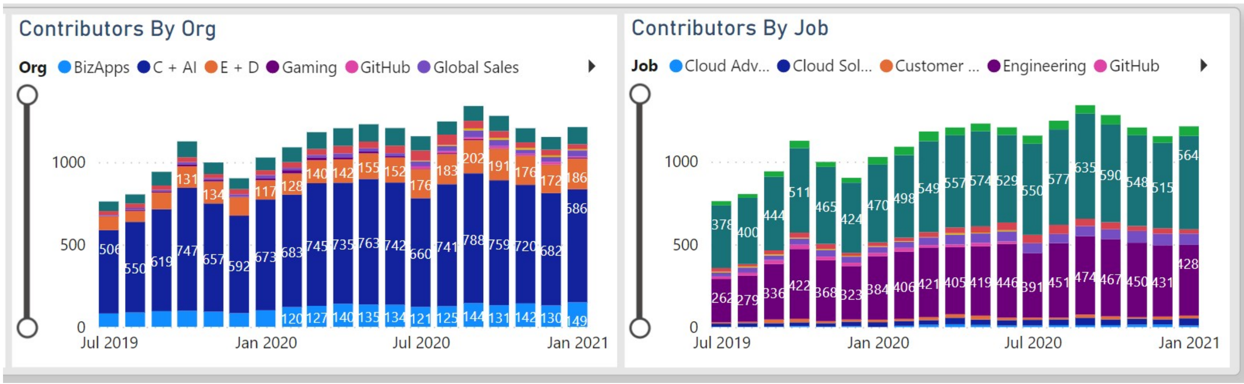
Contributor count over 12 months - Area 3

The contributor engagement window shows the count of contributors that have published content on docs.microsoft.com in the previous 12 months.

In this screenshot, 1900 part-time contributors contributed in only one month of the previous 12 to all repositories, and around 300 full-time contributors have contributed in 11 or 12 months in the past 12.

Digging deeper into contributors

The Contributor Insights dashboard answers basic profiling questions for the selected scope over the past 12 months.



Contributors by job

The contributor insights dashboard categorizes by contributors using their job title in their Microsoft dashboard.

The categories it shows are: - Product PM - Engineer - IT & Service Operations - Cloud Solution Architects & Sales & Marketing - Support - Customer Engineer - Cloud Advocates - Others

Contributors by organization

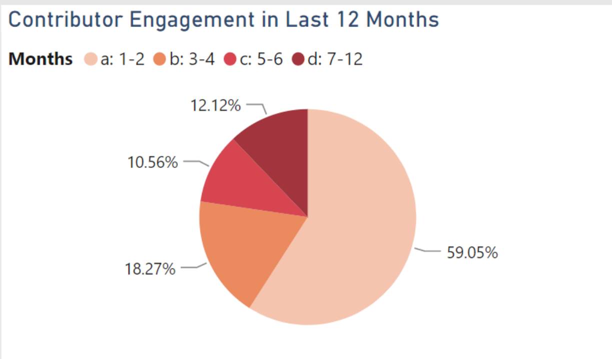
The contributor insights dashboard shows a view of which organization within Microsoft a contributor belongs to. We ask this question by asking which senior leader within Microsoft an individual contributor reports up to.

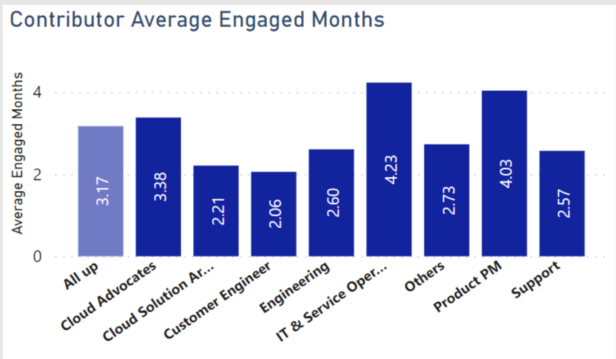
This table shows the designation of organization and who the contributor reports up to for this categorization:

ORGANIZATION	CONTRIBUTOR REPORTS UP TO
C + AI	Scott Guthrie
E + D	Rajesh Ja
WW Commercial Business	Judson Althoff
Global Sales	Jean-Philippe Courtois
Gaming	Phil Spencer
BizApps	James Phillips
Linkedin	Ryan Rolansky
Github	Nat Friedman

How often are part-time contributors contributing

Two visualizations exist to tell the story about how often part time contributors are contributing. The first looks how many of the past 12 months each Part Time contributor showed up in. Since they do not contribute every month, this view shows how active our community is. The second shows the average number of months for all part-timers, but then splits out that average to isolate the activity of the different jobs.





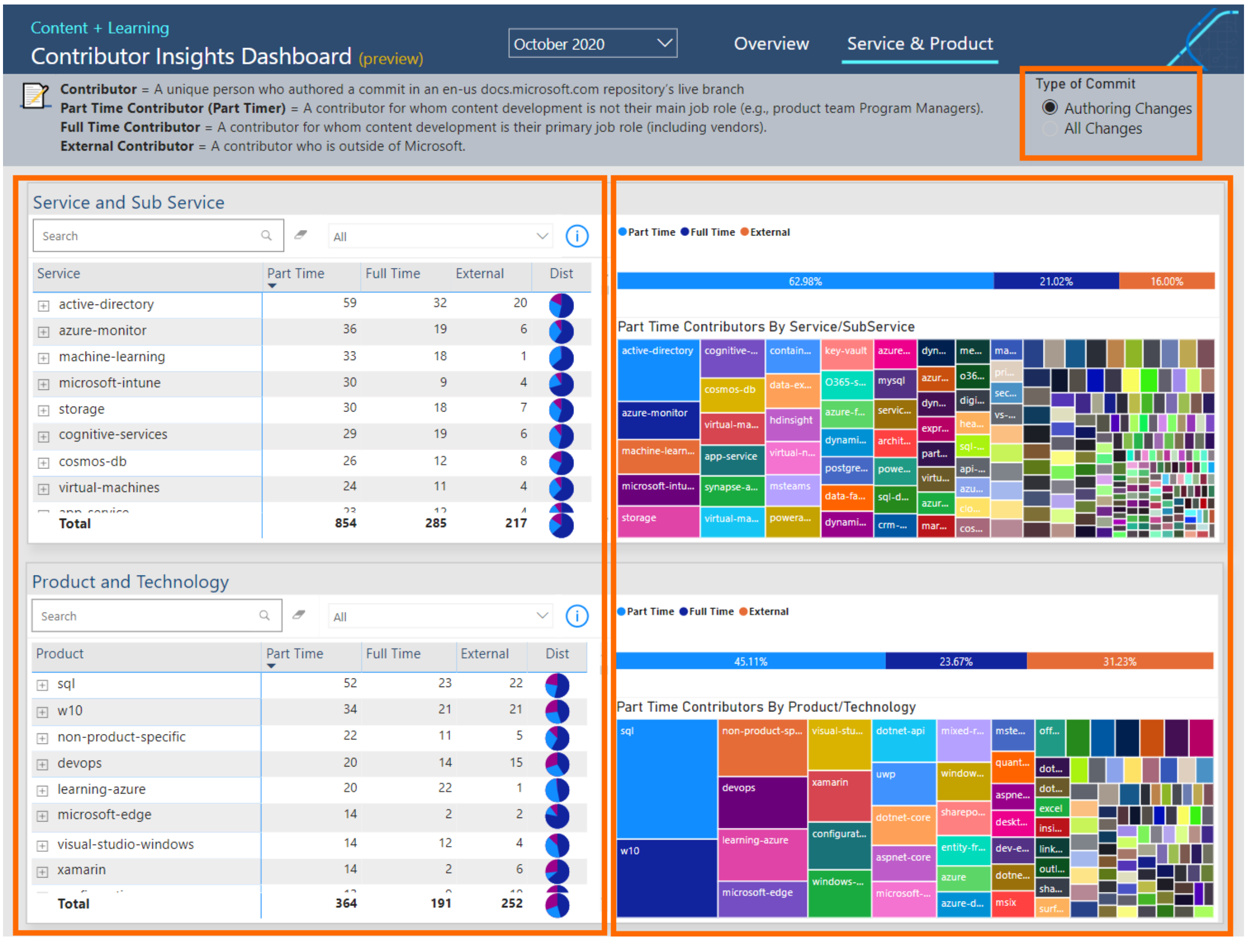
How many contributors this month also contributed last month

A final visualization asks "of the contributors in the current month, how many also contributed last month?" to understand our month over month *return rate* for contributors. It shows both the number of part-timers who returned from the previous month, and the % value of Part Time contributors who returned.

Contributors by product or service

On the Service & Product tab of the report, the contributor insights dashboard shows the count of contributors that committed to individual service and product tags.

> **IMPORTANT**
>
> Use the **Type of Commit** control to toggle between a view that shows *Authoring Changes* only, or *All Changes* which also includes merge commits. The default is *Authoring Changes*.

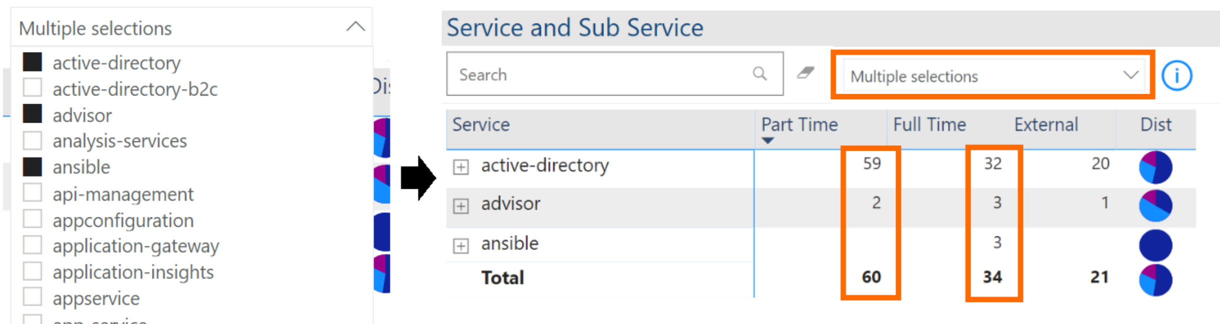


Contributor Counts - This window shows each service/subservice tag and product/technology tag in use on docs and shows the counts of contributors to each service.

Heat Map - The heat map shows a graphical view of the selected tags, and the contributor count of the selection at the top.

Choosing multiple services or products

The Service and Product page allows the selection of multiple entries using the control button.



When multi-selecting services, a count of unique individual contributors is calculated and shown in the **Total** row.

In this screenshot, 59 part-time contributors contributed to Active Directory content and 2 part-time contributors contributed to advisor content. In total, 60 part-time individuals contributed across these two services, because one of the individuals contributed to both services.

Under the hood: Contributor Insights Dashboard data sources

The contributor insights dashboard relies on the following data sources:

- Github commit history to identify authors of all commits that are published to docs.microsoft.com
- 1ES engineering system to correlate github authors with internal Microsoft users, learn their title, and then find which organization they belong to
- HR Information that categorizes Job role into a set of disciplines

How to use the page-level trends & drill-down dashboards

6/25/2021 • 2 minutes to read

You can use the Trends and Drill-down dashboards to find weekly or monthly data trends or to get referrer, verbatim, GitHub issues, or click-through data for a single page.

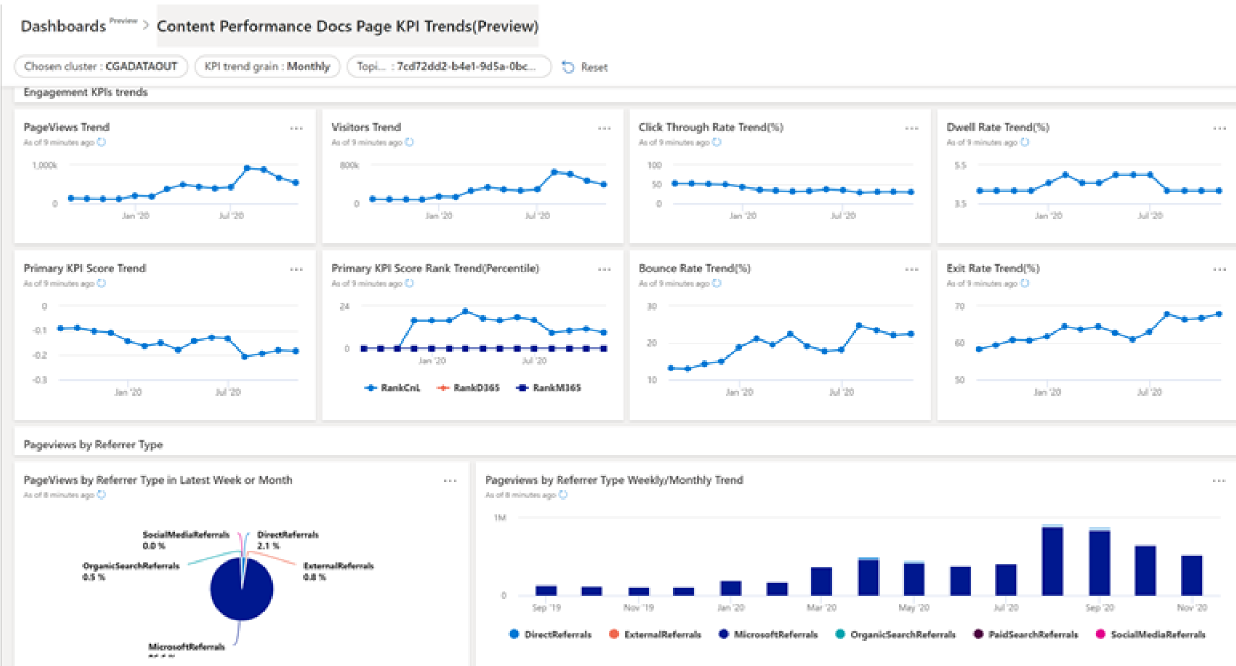
Weekly & monthly trends dashboard

Purpose: Check trends for a single article (up to 14 months or 14 weeks)

How to access: Follow these steps to access from both the Content Performance report and the Microsoft Docs Metrics.

Direct report link: https://aka.ms/contentperfkpitrends

Resources: Check out ways to improve your topics with troubleshooting tips or find definitions for these metrics.



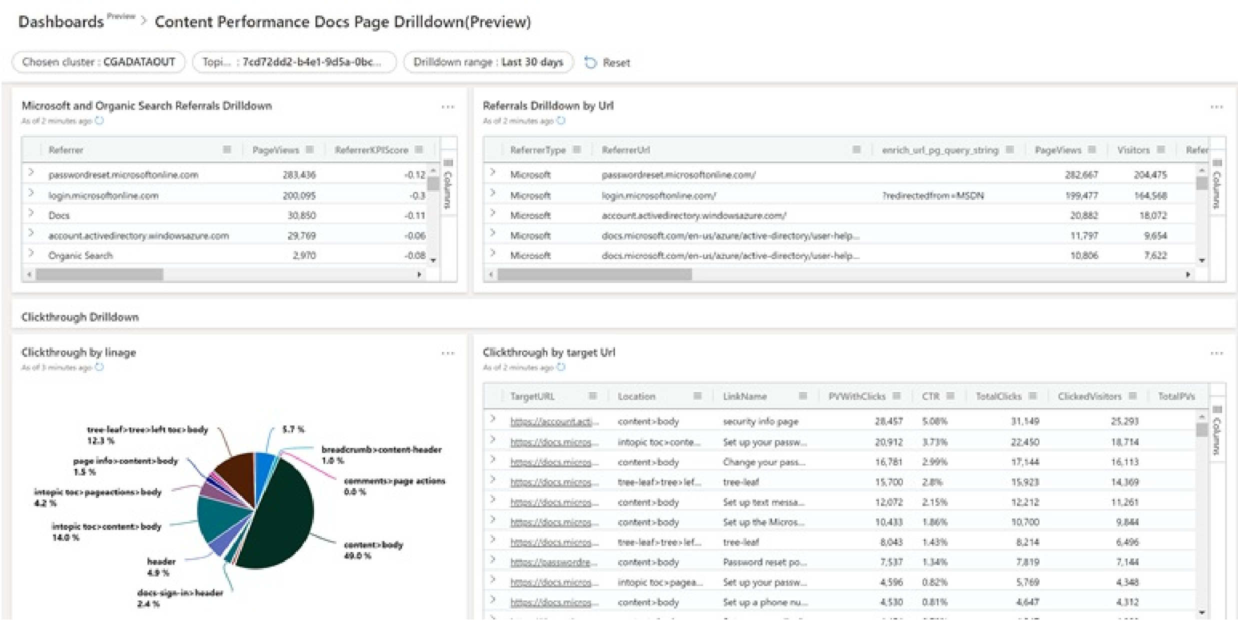
Topic drill-down dashboard

Purpose: Get all the drill-down reports for your topic in one single view (CTR, Referrers, Verbatim, and GitHub issues). You can change the time range to a preconfigured time like LastWeek, Last30Days, or choose Start and End to customize the date range by clicking the Drill-down range filter.

How to access: Follow these steps to access from both the Content Performance report and the Microsoft Docs Metrics.

Direct report link: https://aka.ms/contentperfpagedrilldown

Resources: Check out ways to improve your topics with troubleshooting tips or find definitions for these metrics.



How to access the Trends & Drill-down dashboards

To use these resources, you must join the **CL Dashboard User Group** security group through MyAccess. The security group has been granted access to the **followercgadataout** cluster by the CGA team.

- For most users, no action is required. The entire C&L team, as well as any members of the previous Content Learning Dashboard User Group, have been added to the new security group.
- If you don't have access, you should request access to the new CL Dashboard User Group (links below).

Select the links in the table to submit a MyAccess request. A pre-populated form should appear. MyAccess links can be buggy though. If the pre-populated form does not appear, enter the hyperlink text into the MyAccess search (for example, "CL Dashboard User Group").

KUSTO CLUSTER	ROLE	NOTES
CGA Kusto Follower (followercgadataout.westus.kusto.windows.net) through CL Dashboard User Group	Read-only	**Required** for all users. The default cluster.
CloudMine-Data (https://1es.kusto.windows.net)	CloudMine-Data reader	**Required** for all users to view GitHub issue information in the Drill down dashboard
CGA Kusto Public (cgadatamall.westus.kusto.windows.net)	Read-only	**Optional** for all users. A general purpose public cluster for ad hoc queries

From the Content Performance report

1. Select a topic in either the **Documentation** or **Reference** pages of the C&L report or the BAG or Microsoft 365 page.
2. Click the **View Trends** or **Page drill-down** button. The Kusto query runs in Azure Data Explorer.

From the Microsoft Docs Metrics tool (MDM)

1. From the MDM tool window, click the **Trends** or **Details** link. The Kusto query runs in Azure Data Explorer.

Purpose of zero and low page views reports

3/31/2021 • 2 minutes to read

You can use zero and low page views (PVs) reports to help you decide to retire or archive low-performing articles. A large volume of zero and low PVs articles reside on the Docs site. These docs are of little or no value to customers. They consume resources, bring down the quality metrics for the site, and compete for search visibility.

Why use this report?

Zero and low PVs reports help writers, contributors, and stakeholders:

- Understand where they should or shouldn't invest time.
- See top and bottom articles for PVs.
- Make archiving or unpublishing decisions easier.

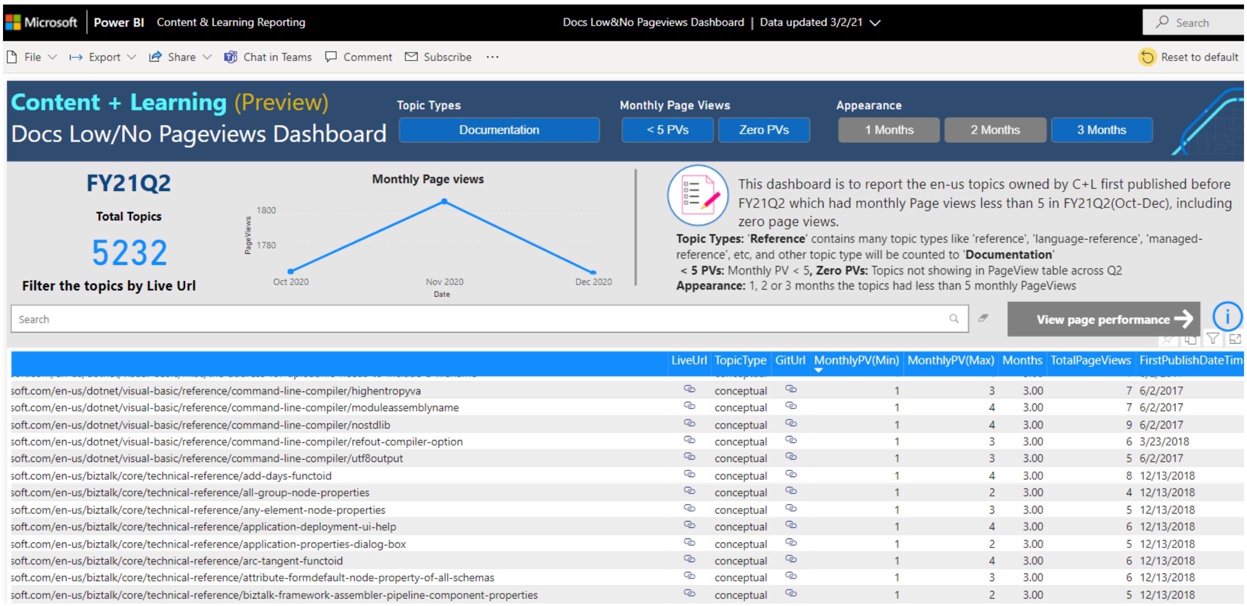
Removing content with zero and low PVs:

- Improves the SEO of the Docs site.
- Reduces costs related to OPS, such as localization, storage, and compute time for builds.

Use zero and low page views reports

The following image of a dashboard shows a list of zero and low PVs reports:

- The report only includes all C+L repos and has the data for both conceptual and reference content. By default, conceptual content is selected. On the report UI, you can select the **Reference** button or cancel the selection to see the data for reference articles.



- Under **Appearance,** three buttons are available. Selecting **1 Month** provides the data for zero and less than five PVs for at least one month in a quarter. Selecting **2 Months** provides the data for zero and less than five PVs in two months of the quarter. Selecting **3 Months** provides the data for zero and less than five PVs for all three months. If you want to get the zero and less than five PVs data for the quarter, select all three buttons. Data for an entire quarter is also the default for this report.
- The report allows filtering by repo name, topic type, ms.service, ms.subservice, ms.prod, ms.technology,

author, GitUrl, liveUrl, and last published date.

- To drill down on a particular URL, enter the URL in the **Search** field and select **View page performance**. Azure Data Explorer displays the performance of the article over a period of time.

FAQ

Why does content that was already redirected appear in this report?

This situation can happen if an article is redirected but the document ID wasn't preserved. In situations where content is moved within a repo or to another repo and the file name is changed, set `redirect_document_id` to true instead of false.

```
      "source_path_from_root": "/articles/multi-factor-authentication/multi-factor-authentication-microsoft-authenticator.md",
      "redirect_url": "./end-user/microsoft-authenticator-app-how-to",
      "redirect_document_id": false
```

For more information about redirection and how it works, see the contributor guide article How to redirect obsolete articles

Why am I seeing reference API content even though I haven't selected the Reference button?

This situation can happen if reference content is missing Topic_type metadata.

Why am I seeing unexpected or duplicate results for versioned content?

When you find an article that you didn't expect to see in a report, select **View page performance**. The article with zero and low PVs that appears in Azure Data Explorer is probably based on a specific version. Content for each version gets its own topic key, and the report is based on the topic key metadata.

Latest Metadata

LiveUrl	TopicType	Monikers	MSService	MSSubService	MSProd	MSTechnology
https://docs.microsoft.com/en-us/visualstudio/mac/open-multiple-solutions	how-to	["vsmac-2017"]			visual-studio-mac	["vs-ide-general"]

In this example, the vsmac-2017 version had less than five PVs, but not the other version of the same article.

The Doc Subset dashboard

6/25/2021 • 2 minutes to read

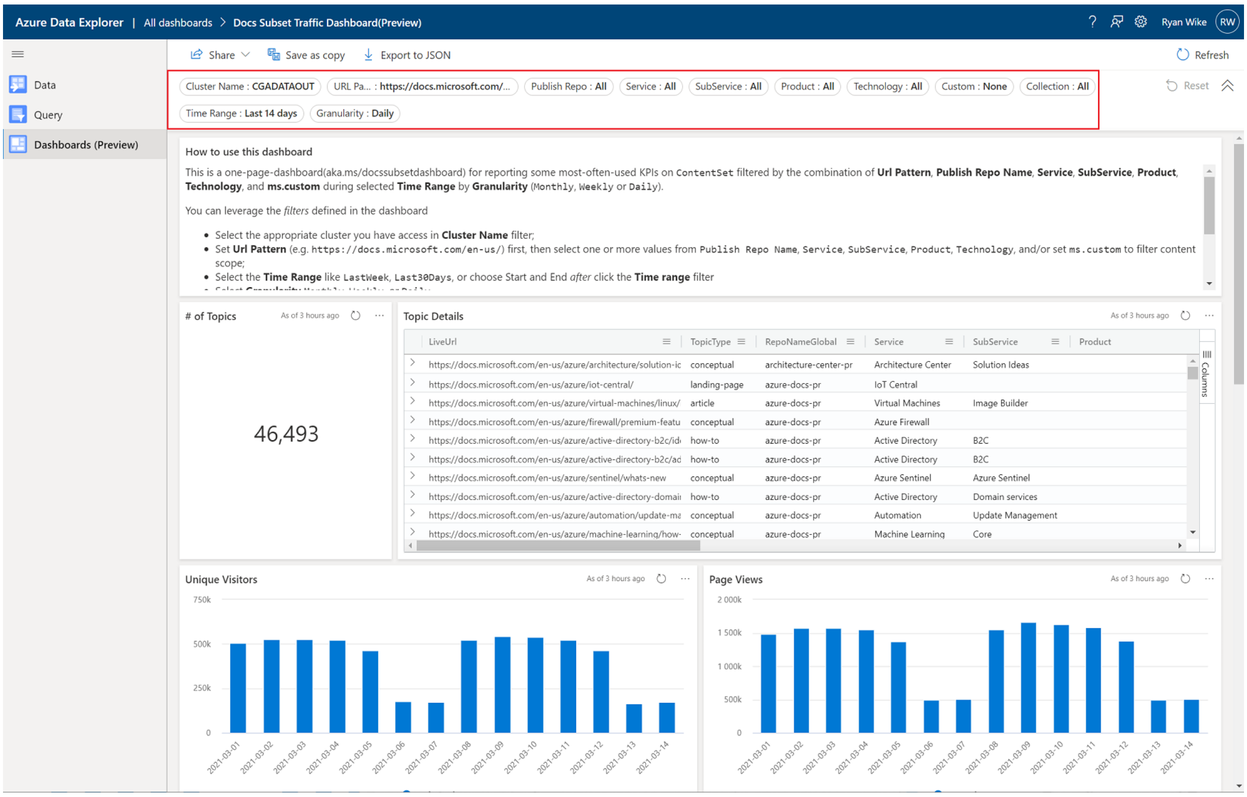
This report shows the most-often-used KPIs for a content set or set of articles (not for individual articles).

Why use this report?

- Useful for identifying trends in your content.
- Provides fine-grained time and granularity controls.
- Search more than two months into the past.
- Easy to filter on lang-local in the URL.

How to use the Doc Subset dashboard

Direct report link: https://aka.ms/docssubsetdashboard. For info on getting access to the dashboard, see the prerequisites.



Keep the default Kusto cluster (FOLLOWERCGADATAOUT) in **Cluster Name** filter.

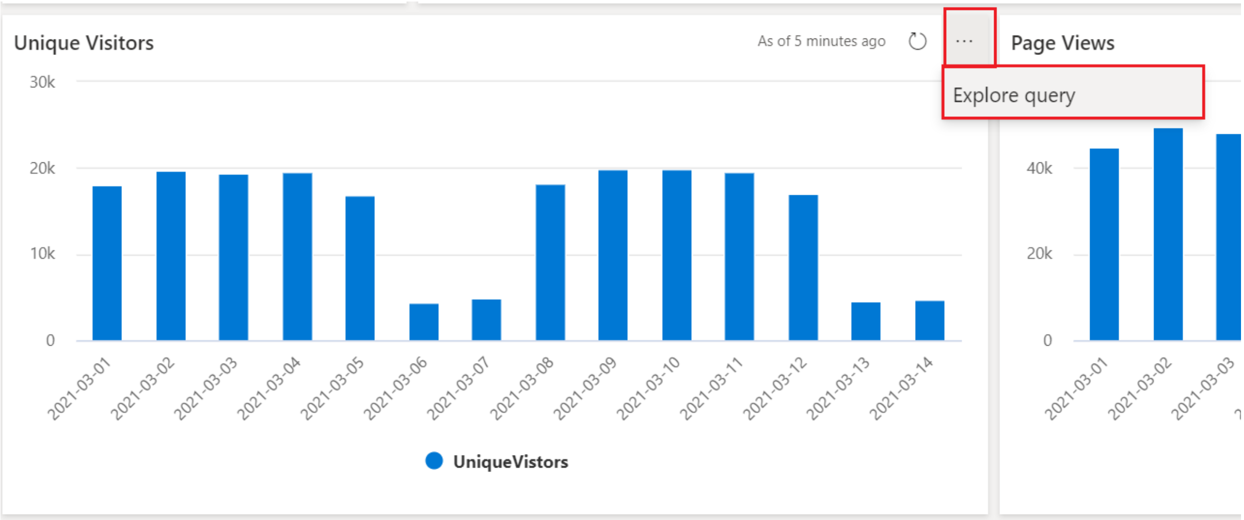
1. Set **Url Pattern** with language-locale (for example, `https://docs.microsoft.com/en-us/`) first, then select one or more values from **Publish Repo Name**, **Service**, **SubService**, **Product**, **Technology**, and/or set **Custom** (ms.custom metadata value) to filter content scope.
2. Select the **Time Range** (for example, last 14 days, last 60 days, or a custom date range).
3. Select **Granularity** (Monthly, Weekly, or Daily).

> **NOTE**
>
> When querying a large content scope, it may take some time to refresh the dashboard due to Kusto performance limitations.

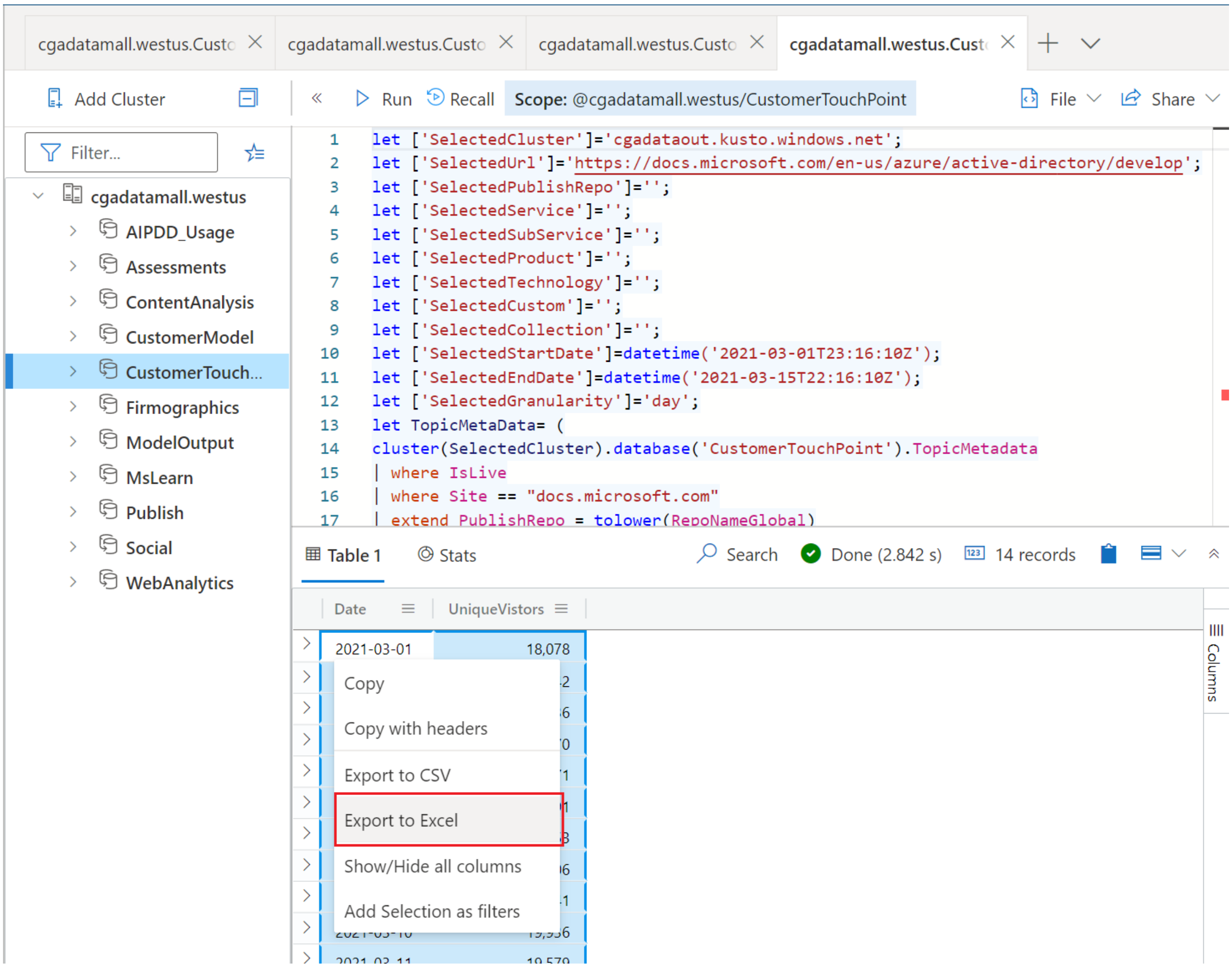
How to export data

You can export data from the charts in the Doc Subset dashboard, for example **PageViews** or **Unique Visitors**.

Select the ellipses (...) in the upper right corner of the chart and select **Explore query**.



A query window opens in Azure Data Explorer. In the results window select the data, right click, and select an export option.



How to get support or submit feedback

You can get support or route feedback through the analytics teams channel: Analytics teams channel.

How to access the DocSubset dashboard

You can access this report from within the Doc & Ref Dashboard, or directly at https://aka.ms/docssubsetdashboard.

To use these resources, you must join the **CL Dashboard User Group** security group through MyAccess. The security group has been granted access to the **followercgadataout** cluster by the CGA team.

- For most users, no action is required. The entire C&L team, as well as any members of the previous Content Learning Dashboard User Group, have been added to the new security group.
- If you don't have access, you should request access to the new CL Dashboard User Group (links below).

Select the links in the table to submit a MyAccess request. A pre-populated form should appear. MyAccess links can be buggy though. If the pre-populated form does not appear, enter the hyperlink text into the MyAccess search (for example, "CL Dashboard User Group").

KUSTO CLUSTER	ROLE	NOTES
CGA Kusto Follower (followercgadataout.westus.kusto.windows.net) through CL Dashboard User Group	Read-only	**Required** for all users. The default cluster.
CloudMine-Data (https://1es.kusto.windows.net)	CloudMine-Data reader	**Required** for all users to view GitHub issue information in the Drill down dashboard
CGA Kusto Public (cgadatamall.westus.kusto.windows.net)	Read-only	**Optional** for all users. A general purpose public cluster for ad hoc queries

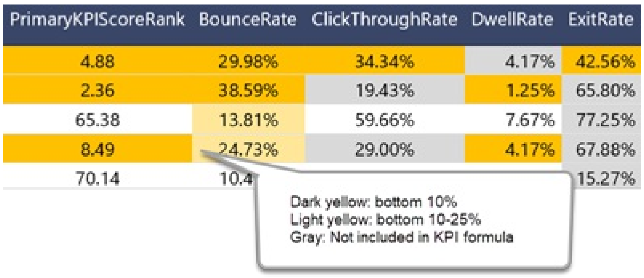
How to troubleshoot lower-performing articles

7/8/2021 • 15 minutes to read

After you've identified a lower-performing article, you can use the steps in this article to make improvements to it, depending on which metrics might be causing the performance issue.

To identify lower-performing articles, follow these steps:

1. In the C&L Content Performance report, navigate to the **Documentation** or **Reference** page. In the BAG report, navigate to the **Business Applications Group** page, and in the Microsoft 365 report, use the **Microsoft365** page.
2. In the far-right of the Power BI report, use the **Filters** pane to scope to a content set based on your preference; for example, by URL, author, repo.
3. Look for articles with a KPI score rank of 25% or less, highlighted in light or dark yellow. Or in the **Filter** pane, in the **PrimaryKPIScoreRank** filter, select **Advanced filtering**, select **is less than or equal to**, enter **25**, and then click **Apply filter**.
4. Sort the table by highest page views.



PrimaryKPIScoreRank	BounceRate	ClickThroughRate	DwellRate	ExitRate
4.88	29.98%	34.34%	4.17%	42.56%
2.36	38.59%	19.43%	1.25%	65.80%
65.38	13.81%	59.66%	7.67%	77.25%
8.49	24.73%	29.00%	4.17%	67.88%
70.14	10.4			15.27%

Identify the correct topic type and corresponding formula

Make sure you assign the correct ms.topic value for the article. Because the KPI Score is based on the topic type, it's important to correctly label your topics so it is compared to similar topic types. Review the primaryKPIScore calculations in the Primary KPI metric article.

For example, if the title of the article is "How to troubleshoot 2-factor authentication problems in Windows Intune" and it contains troubleshooting procedures, and the article is tagged as **conceptual**, then **Troubleshooting** would be more appropriate so that troubleshooting articles are compared to other troubleshooting articles.

If you change the ms.topic type, the KPI Score Rank or score (if topic type has different formula) will be recalculated in the next monthly report. Correct classification could change your score, so you may want to change only the ms.topic type before you invest more effort.

Tips for high bounce rate

If BounceRate is flagged, it means that the topic didn't meet customer expectations and they left within 5 seconds without engaging with anything on the page. They didn't click a link, and they didn't scroll. Often a particular referrer can cause poor performance. How is BounceRate calculated?

How to influence bounce rate:

- Check referrer traffic. Are there any referrers that have high bounce rates, signaling that the link to your topic isn't meeting customer intent. If possible, work with referrer owners (for example, product or website owners) to point to better content or change the content to meet the expectations of that referred traffic.

- Review customer feedback. Are there any verbatim or GitHub issues that provide hints to customer issues?
- Consider clearer headings and intro paragraphs to set expectations. If customers can't tell what's in it for them right away, they'll bounce.

Tips for low click-through rate

If the ClickThroughRate is flagged, it might indicate that there aren't links in the body of the topic or the links that are present aren't visible or relevant. How is ClickThroughRate calculated?

How to influence click-through rate:

- Improve bounce rates. If you can improve your bounce rate by decreasing it, overall page engagement should also improve.
- Cross linking. Ensure the topic has relevant cross links to related topics or next steps. Review the click patterns and verbatim for possible clues.
- Reposition links. On pages designed for navigation, like hub and landing pages, it's useful to identify links/buttons lower on the page that get more clicks and move them up. Remove links that customers aren't clicking to simplify choice. For longer articles, consider moving high value links earlier in the topic.
- Consider clearer link text and calls to action so users know what to expect before they click. Link text should be short and descriptive.

Check what and where users are clicking

For poor performance on click-through, it's useful to look at what customers are clicking on a page. You can check click activity for a page in two ways:

- Use the Microsoft Docs Metrics extension for a quick visual of where users are clicking on the page. Note that the click totals are aggregated by target URL, so if the same link appears multiple times on the page, the total all of the clicks for that URL, not the specific link.
- Use the click-through drill-down report that's in the Drill down dashboard. On the **Documentation** or **Reference** pages, select a single article from the table, and then click the **Page drilldown** button in the top-right corner. (Requires Kusto permissions)

Click report details

The Page drilldown report includes a pie chart of click activity by area and a table of each link with relevant metrics.

COLUMN HEADING	DEFINITION
Location	Where the link is located on the page See Link location
LinkName	Text of link
Target URL	URL associated with URL
Clicks	All clicks on the particular link
ClicksPVs	All clicks on a particular link by page view. Will be less than "clicks" as the links are deduped
ClicksVisitors	All clicks on a particular link by unique visitor. Will be less than clicks or clicks by PV as this value is deduped by user
TotalPVs	Number of times the selected page was loaded

COLUMN HEADING	DEFINITION
TotalVisitors	Number of individual users to the page identified by a web cookie
CTRbyPV	Number of clicks by page view as a percentage of total page views for this page (pageClicksPageViews/TotalVisitors)
CTRbyVisitor	Number of clicks by visitor as a percentage of total visitors (pageClicksVisitors/TotalVisitors)

Link location

Here's a key for where links are located on the page. If you right-click the page and view source you can search for "data-bi-name" and that typically provides the location information.

LOCATION	AREA
1	content>body
2	header
3	content-header (breadcrumb and comments)
4	TOC: **tree-leaf>tree>left toc** (mobile version) and if includes >**body** (tablet resolution or larger)
5	In this article: intopic toc
6	moniker-picker
7	search (blank, use target URL for docs.microsoft.com/search to identify)
8	footer (terms of use, language selector)
9	recommendations

Tips for low copy-try-scroll rate

If CopyTryScrollRate is flagged, it means that customers didn't frequently scroll, copy code, or use the Try button. How is CopyTryScrollRate calculated?

How to influence copy-try-scroll rate:

- Check your BounceRate. If BounceRate is high, address BounceRate and SEO issues.
- Check your ClickThroughRate. If ClickThroughRate is high, and CopyTryScroll is low, this is usually expected (customers may click a link instead of scrolling). In this scenario, generally Primary KPI is good, so this is a good result.
- Check your topic length. Very short topics have low CopyTryScrollRate, because for some window sizes there isn't opportunity to scroll. Evaluate whether the topic should be combined with other topics, or is a candidate for removal.
- Check the customer focus. Is the customer finding what they need before scrolling? You can use MedianMaxScroll to help investigate.

Tips for low dwell rate

If DwellRate is flagged, it means that while customers aren't bouncing, they aren't staying on the page as long as expected either. How is DwellRate calculated? DwellRate is no longer a calculation in the new performance report.

How to influence dwell rate:

- Review customer feedback. Are there any verbatim or GitHub issues that provide any hints to customer issues?
- Run the referrer drill-down query. Are there any referrers that have high bounce rates, signaling that the link to your topic isn't meeting customer intent. If possible, work with referrer owners (for example, product or website owners) to point to better content or change the content to meet the expectations of that referred traffic.
- Consider content structure. Customers often scan webpages to find information.
  - Can you tighten the topic to focus on the customer intent? Can you break up large paragraphs into bullets or use visuals to explain concepts?
  - Is the article visually appealing? Does it have an image "above the fold," or is it a "wall of words"?
  - If there are distinct topics that won't compete in search, consider breaking a larger article into smaller articles with more focused intents.

Tips for high exit rate

If ExitRate is flagged as high for a landing, hub, or sample index page, it means that customers aren't clicking through, so the page is not serving its purpose, which is to drive customers to other pages within the content set. How is ExitRate calculated?

How to influence exit rate:

- Optimize for Search. Review the Search-related metadata to make sure your title, description, and H1 are optimized for the right audience.
- Review click patterns. Review your link selection to find areas to improve and consider clearer link text and calls to action so customers know what to expect before they click. Link text should be short and descriptive.

Check referrer traffic

A common problem with content is a mismatch between the expectation set by the referrer and the content on the page. It's often useful to look at a breakdown of all the referring traffic sources, and how performance metrics vary across each one. You may find a referrer that is driving many page views and causing high bounce rates compared to other referrers.

1. To run the referrer report from the Content Performance report, on the **Documentation** or **Reference** page, select a single article from the table, and then click the **Page drilldown** button in the top-right corner. To run the referrer report from the MDM extension, click the **Details** link.
2. An Azure Data Explorer dashboard runs for your page (based on TopicKeyID). If you don't get results, check your cluster permission or change the **Chosen cluster** at top of report.
3. To identify which referrers are causing the low page performance, sort the **PageView** column by the highest traffic, and then look at the bounce rate column to see which referrers have the highest bounce rates. Alternatively, you can use the Referrer KPIScore. Each referral is assessed by using the same Primary KPI Score formula as the topic.

Search referrer traffic with high bounce rate

If the high bounce rate is coming from search traffic, such as Google, then optimizing your page for Search is the best bet.

- Use the steps outlined in SEO: Basic techniques for good web writing.
- Does your topic share common search queries with a different audience type; for example, could consumers be landing on it because of how it displays in search results? Can you update the metadata title or the opening paragraph to match the article's intent better or add a Note that directs them to a relevant Support topic?

Direct referrer traffic with high bounce rate

High bounce rate caused by direct traffic is tough to pinpoint since direct traffic doesn't have a known referring URL.

POTENTIAL SOURCES OF DIRECT TRAFFIC	MORE INFO
In-product link (non Web products)	Ask your product team for lists or ideas. Context gets lost and cannot be determined. Consider adding a campaign ID to your in product links, which will add a parameter you can see in the Referrer drill-down report. More details about this approach are outlined in the topic Links to articles from the UI.
Outdated URLs causing double hops	Double hops (A->B, B->C) lose referrer context: • **Server-side redirects**: The original URLs that hits server-side redirects. Legacy blogs, forums, or Support sites might contain links to your topic that are now 301 redirects; for example, with "msdn.microsoft.com" in them. If possible, reach out to the site owner to get those links updated or removed. • **FWlinks** and **AKA.ms** who have outdated URLs that then get redirected. **Tip:** Reach out to the FWlink or AKA link team for links that include an old URL base. • Multiple redirections in your `.openpublishing.redirection.json` file. Beware of daisy-chained redirects. **Tip:** If your topic changed URL structures, be sure to look for those variations in the fwlink tool and update the target URL. You might need to change the page the link is pointing to or work with the product team to remove the link.

POTENTIAL SOURCES OF DIRECT TRAFFIC	MORE INFO
Bot or other fraudulent traffic	High levels of direct traffic may also be an indication of fraudulent bot traffic that needs to be reported. Some testing systems are also known to generate large amounts of direct traffic, particularly for landing pages. If you suspect this might be the case, reach out to the C&L Insights and Analytics team so they can investigate and add to the exclude list.
In-browser bookmark	Users may bookmark a page and come to your site that way which hides context. There is no referrer in this case.
URL entered in browser	There is no referrer in this case.
Email-based messages	In email, newsletters, messages, and campaigns cannot provide referral context.

External or Microsoft referrer traffic with high bounce rate

If external or Microsoft-owned web pages are responsible for the high bounce rates compared to other traffic sources, then you should check the context of the backlink on those pages. Is it misleading? If so, modify the link text or remove or replace the link so the correct context is set.

Check verbatim comments and GitHub issues

Incorporating customer feedback from feedback sources is always a good thing to do when you are troubleshooting article performance.

- **Verbatim in Content Performance**. In the Content Performance report, check the **Verbatims** column to see if there are any for the month. If there are, click the topic to highlight it, and then click the **Page drilldown** button in the top-right of the page to open the Drilldown dashboard for the topic. (Quick tip: In the Drilldown dashboard, if you click the ellipses in the right side of the tile, and then click **Explore query**, you can manually change the date range in the Kusto query to include previous months).
- **Verbatim in the MDM tool**. In the Microsoft Docs Metrics tool, check the **Satisfaction** section. To access the verbatim, click the **Details** link to open the Drilldown report.
- **Verbatims for multiple pages or months**. To see customer feedback across a content set or just one page for the past 90 days, use the Verbatims report.
- **GitHub issues**. Check the GitHubOpenIssueCount column, highlight the topic, and then click the **Page drilldown** button in the top-right of the page to open the Drilldown dashboard for the topic. This requires access to the **CloudMine-Data reader role** in the 1es CloudMine cluster (https://1es.kusto.windows.net/).

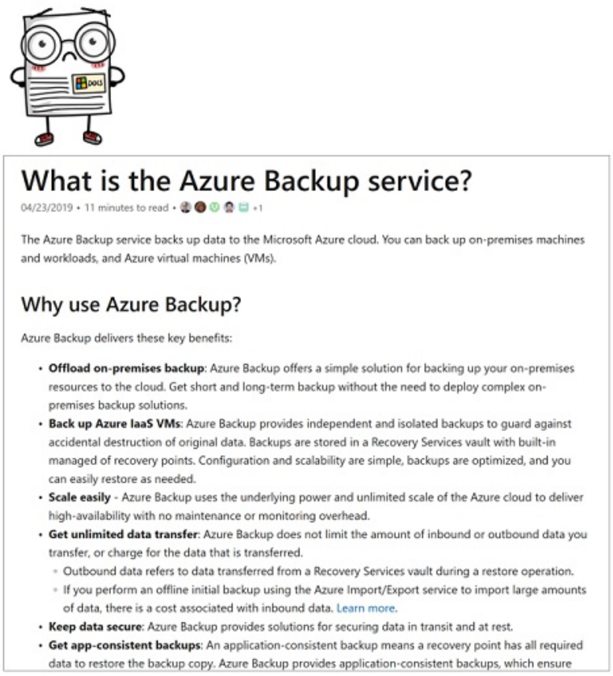
Verify content follows best practices

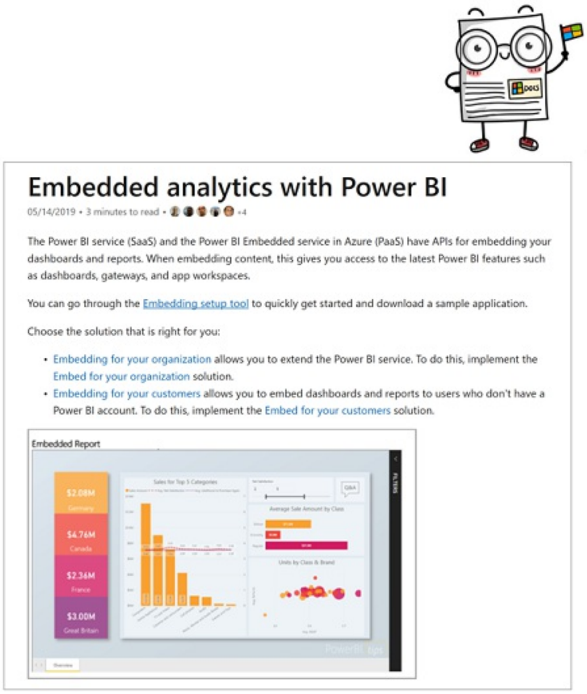
Frequently, low performing articles simply don't comply with our core content authoring, voice or SEO guidance, and that lack of compliance drives low performance. You may need to update the article to better address a single customer intent, to include images, or to reduce extraneous text. If the article is long, you may need to break it into smaller articles.

- Voice and writing guidance is available, and it makes a difference!
- Request a developmental edit to obtain feedback on voice and writing guidance.
- SEO: Basic techniques for good web writing.
- Consult the writer's SEO cheat sheet. Pay particular attention to: titles, meta descriptions, H1s
- Freshness + maintenance checklist. Check the Freshness column to see if it's been awhile since the content was updated. Screenshots, links, or outdated information might be causing the underlying issues with the

topic.

Looking at a higher performing article (not flagged in yellow) of the same type may provide insight. For example, let's compare stats for the following articles.





We see that the articles, while of the same type and with similar monthly page views, were different from one another. The higher performing article follows our modern voice guidance and content guidelines for well-written content. The lower performing article does not.

ARTICLE 1	ARTICLE 2
Appx page views 10,000/month, page type = overview	Appx page views 10,000/month, page type=overview
11 minute estimated read time	3 minute estimated read time
Wall of text	Attractive, even unusual graphics, one of which is above the fold
Little white space, not easy to scan	Noticeable white space, easier to scan
Tons of tables with a lot of minutiae	
Next steps section uses text-based links	Next steps section uses large clickable buttons
Low levels of click through	High levels of click through
Seems to lack a single customer intent	Clearly focused customer intent

In this case, the recommendations to fix Article 1 included:

- Identify clear customer intents for the different parts of the article and break it up into multiple, smaller, more focused articles.
- Add some visuals as appropriate – you need them to draw the user in.
- Add relevant links in more prominent places in the article to encourage click through

Bubble up product pain points

You may be using the steps in this article as part of trying to raise CSAT for a technical article on docs.microsoft.com. After analyzing the article's performance, you may conclude that the right users are finding the article but they are very unhappy because the document covers a product feature or area that generates a lot of customer pain. That pain is then expressed through negative verbatim, angry GitHub issues, and low CSAT that cannot be addressed by changes in the article. In these cases, use the negative data around page views, low CSAT, and customer feedback to try to drive a product fix. We can fix bad documentation, but we cannot fix a broken product experience with a doc.

Article assessment: Measure the impact of your changes

The Article Assessment PowerApp that was a separate page in the Content Performance report was deprecated in June 2021. To compare the performance of a single article, before and after you make changes to it, you can use the View Trends dashboard (accessed through the Documentation page in the same dashboard, as well as a direct aka.ms link) by week or by month.

While a minimum of 3 to 4 weeks is needed before SEO improvements can be discerned, the core bounce, scroll rate, click-through, and exit rate metrics typically will show a response during the first full week after the publication week.

As you make more changes within a content set, month over month, you should monitor that content set for overall improvement of the percent of page views that go to higher performing content. On the **Overview** tab of the content performance dashboard, filter to the content set you care about and note the monthly changes overall.

Resources

- Content performance report
- Learn how to use the Content performance report

Content performance reporting principles

7/7/2021 • 3 minutes to read

The most common concern about the content performance reporting is that a particular metric is unfair or inappropriate. We might imagine scenarios when the metric's high or low score isn't necessarily a good or bad thing. This article describes the principles that inform our approach to content performance.

We compare articles to their peers and look for outliers

You may think it's unfair to measure, say, the ClickThroughRate on Overviews. You could believe that Overviews are just walls of prose with few links. Keep in mind that when we're determining under- or over-performance, we're comparing that rate against the rate of other Overviews. Those imagined scenarios where the metrics don't seem "true" to the experience apply to all Overviews, and so factor out. In the end, we still have to wonder and investigate why some Overviews have such anomalously low rates among their peers. It's why we compare articles to their peers and then look for outliers.

Engagement is a good thing to see in aggregate data

To continue the example, ClickThroughRate is a good indicator of a good experience in nearly all situations. So are the many other metrics we use:

- CopyTryEngagementRate
- StartTutorialRate
- DeployClickRate
- and so on

They show users engaging further with our content. In the ClickThroughRate example, it's hard to argue that it's a bad thing when users choose to click and view another page (or copy code, or start a tutorial...) after they read an Overview that introduces them to a service. It's especially true when we're measuring whether users dwelt long enough (DwellTimeDelta) to have read the Overview and combining that into the performance score. These behaviors are good things to see, and they're easily and reliably measured. Yes, it's possible lots of clicking is people lost on docs.microsoft.com. We have other metrics and other user research methods employed to easily detect people who are clearly lost on docs.microsoft.com.

Reporting is only as good as your metadata

We certainly do have limited metadata accuracy in some cases. For example, there are lots of different content types that we give an `ms.topic` value of `article` and `conceptual`. Generalization prevents us from measuring some of them appropriately or "fairly". However, the solution here isn't "hold off on measuring content performance until we can fix the metadata." The solution is to fix the metadata. We need good metadata for lots of reasons beyond reporting: personalization, searching, browsing, SEO. and so on.

We can change the metrics, but only if we've done the work to know we should

That said, we do routinely look for opportunities to try new metrics, or stop using some metrics that don't prove to be actionable. However, proving a metric isn't actionable requires us to actually try to improve it on many articles, and be ready to show our work. Then we can really know when a metric isn't worth our while. If we're willing to put in the work, we'll definitely find some duds down the road. Ultimately, that's not the point of putting in the work.

Content performance isn't personal performance

We need to remember that with content performance, we're not measuring ourselves nor our work. We're measuring engagement and using it as a tool to help us do our work. Content performance isn't tied to our Connects. If anything, what's tied to our Connects should be what we learn together from using a common framework for evaluating our users' experience with our content. Authors should have individual goals around using analytics and iteration to identify generalizable insights so that we can discover tips and tricks that drive performance when applied in different contexts and educate the team about it.

Release notes for Content Performance and other reports

7/8/2021 • 18 minutes to read

The release notes are organized by the month that the report is refreshed with the previous month's data, for example, June data is released in July.

July 2021

- **CL Dashboard User Group renewal notices**. You might receive a renewal prompt if you changed managers during the recent Content & Learning reorg. No action needed unless you truly lost access. Use this report Azure Data Explorer to test. Send an email to Yiwen Zhang (yiwzhan) if you no longer have access to this report.
- **June data for page views to higher or lower performance**. The page views and articles to higher vs. lower performing content that's included on the Overview page of the Content Performance report wasn't ready until 7/7 because of the retirement of the Employee table on CGA cluster for increased security concerns. The page-level data wasn't impacted. The Analytics team has put in a temporary fix and all the reports impacted above have been updated. A long-term solution will be implemented at a later time without disruption to users.
- **Content Freshness column**. This column was updated to reflect the latest freshness categories and a filter was added to the Filter pane. Learn more about content freshness.
  - 0-90 days (Freshest). No conditional formatting
  - 91-365 days (Fresh). Light yellow
  - 365 days (Stale). Dark yellow
- **Article assessment PowerApp**. Due to increased security standard on PowerApp, the Article Assessment was depreciated. The View Trends dashboard (accessed through the Documentation page in the same dashboard, as well as a direct aka.ms link) was implemented to accomplish similar report/analysis although not in the exact same format/capability. We'll be assessing whether we need to make updates to the ADE if desirable and viable. Feel free to reach out to kellypi or yiwzhan.
- **June 2021 reporting insights**. The overall page views were up by 8% and the percentage of higher performing increased by ~4% pts to 80.42% for C&L Content Performance. The increase is related to the Windows 11 announcement and Documentation release, with huge PV increases to the Windows Hardware commercialization-docs-pr repo.

June 2021

- **Azure Data Explore Reports changed to Follower cluster**. CGA has scaled up the follower (followercgadataout) cluster to be more powerful than the public (cgadatamall) cluster, and positioned follower cluster to handle queries from non-ad hoc process/system (including ADE dashboards). Thus we have switched to followercgadataout as the default cluster for ADE reports. Details about accessing these reports are documented in the Page-level trends & drill-down dashboards and Doc Subset dashboard topics.
- **Cluster access request through security group**. The process to request Cluster access has been streamlined. Instead of requesting access to the cluster directly, access is granted through the CL Dashboard User Group security group through MyAccess. The security group has been granted

followercgadataout access by the CGA team. **No action is required for most users**:

- All of the C&L team as well as any members of the previous **Content Learning Dashboard User Group** were added to CL Dashboard User Group.
- If for some reason you don't have access to the reports, then request access to the new CL Dashboard User Group.
- Access to cgadatamall is not required to access the reports, and the approval is handled directly by CGA team through MyAccess.

- **May 2021 reporting insights**.
  - The general drop is page views in May follows the annual trend.
  - The % of overall page views dropped by 6.5 percentage points, likely the cause of updates to ms.topic metadata, which decreased the number of conceptual and article topics in the overall pool and changed the bottom 25%.
- **Verbatim report not updated since June 2021**. Note that this report hasn't been updated since early June. Verbatim categorization powered by the CGA data science team stopped updating in early June in the Verbatims report, and likely will no longer be available due to product deprecation. After we confirm the report (without categorization) is still needed, we will switch the data source, and keep the verbatim report but with no categorization. Verbatim categorization work will be planned out in FY22.

April 2021

- **Kusto cluster update**. The endpoint for **DevRel read-only** users of the CGA Kusto cluster **cgadataout** changed on April 20. This includes the CustomerTouchpoint, WebAnalytics, MsLearn, and Publish databases. The new location is **CGA KustoFollow DevRel** (followercgadataout.westus.kusto.windows.net). If you already had permissions to **cgadataout**, your permissions should have been automatically migrated to the new cluster.
- **Added Page Views (PVs) month over month (MoM)**. Available in a couple of places in the report:
  - On the **Overview** page, in the bar chart as a Tooltip. Hover over each month to expose the Tooltip or export the monthly data by clicking the ellipses in top-right corner, and then **Export data**.
  - On the **Overview** page, in the Performance drilldown section, as a new column titled **PVs MoM**.
  - On the **Documentation**, **Reference**, and Partner-specific pages, as a new column titled **PVs MOM**.
- **Page rank change**. To show whether a page changed its rank from the previous month, look for the new column **PrimaryKPIScoreRank Change** in the **Documentation**, **Reference**, and Partner-specific pages.
- **Data alerts**. Added notices to highlight any months with data loss on the **Overview** page. Any month with a data issue will have info in the Tooltip for that month and an alert in the Performance drilldown section of the page.

February 2021

- **High bounce rates** - In February, some engagement events related to dwell on pages were under-counted, resulting in over-counted bounce rates, which will show a spike in this metric. This data gap impacts all of docs.microsoft.com and related reports.
  - Period of data impacted: 2/9-3/3
  - Background: A frontend docs bug introduced on 2/9 prevented the following custom telemetry events from firing: page-focus-changed, page-visibility-changed, select-value-changed, print, secondary-content-scroll, and unload.
- **Filter pane**. Added Title as a filter on the Documentation, Reference, BAG, and Microsoft 365 pages
- **BAG version of report**. Filtered out nine low-priority repos for the BAG report
- **Exporting data**. Fixed the ability to export underlying data so that service and product columns are

included

January 2021

New formula for most pages. Replaced DwellRate with a new metric CopyTryScrollRate. See more details.

- Non-nav pages: (CopyTryScrollRate + ClickThroughRate) – BounceRate
- Reference: CopyTryScrollRate – BounceRate
- No change to hub/landing formula: 2 * ClickThroughRate - (ExitRate + BounceRate)

New version of reports. The new formulas required us to rerun the data for all pages, which updated our baseline trends, and we've created a new version of the report so there's an archive of the legacy report for comparison. As a result, any updates to metadata or page redirection might impact your historical page view data. The new version uses the same aka.ms links and the legacy report can be accessed through these links:

- https://aka.ms/contentperf-legacy (Content & Learning)
- https://aka.ms/contentperf-bag-legacy (Business Application Group)
- https://aka.ms/contentperf-m365-legacy (Microsoft 365)

Overview page updates. A few notable updates:

- Trend chart now shows 13 months instead of 7.
- New filter for URL (level 1 and level 2).
- The Performance drill-down report is now on this page rather than a standalone page and includes a column for % of high performing articles.

Documentation and Reference pages

- CopyTryScrollRate is part of the KPI formulas and other metrics are for available for diagnostics:
  - CopyTryScrollRate
  - CopyRate
  - TryRate
  - MedianMaxScrollRate
  - OrganicSearchReferrers
- Weekly and Monthly trend data is available through the **View Trends** button in the top-right corner.
- Referrer, Verbatim, Click-through data, and GitHub issues are available through the **Drill-down** button in the top-right corner rather than in individual columns in the table.

Report scope changes. Several URL paths were either excluded from all reports or added to a different report based on the organization the content belongs to. The overall page views will differ between the new and legacy reports.

ORGANIZATION	EXCLUDED	ADDED
Content & Learning	/search (3.6M PVs) /locale (173K PVs) /answers (171K PVs) /compliance (155K PVs) /ai-builder (33K PVs) /teamblog (29K PVs) /contribute (23K PVs)	No change

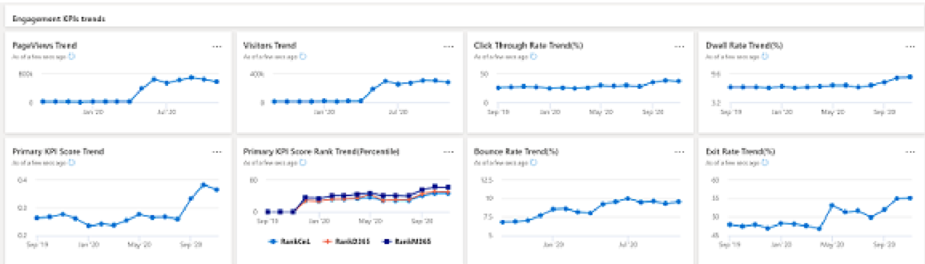
ORGANIZATION	EXCLUDED	ADDED
Microsoft 365	No change	/kaizala /stream /openspecs (400K PVs) /compliance (155K PVs)
BAG	/dynamics	/ai-builder (33K PVs) /industry (6K PVs)

November 2020

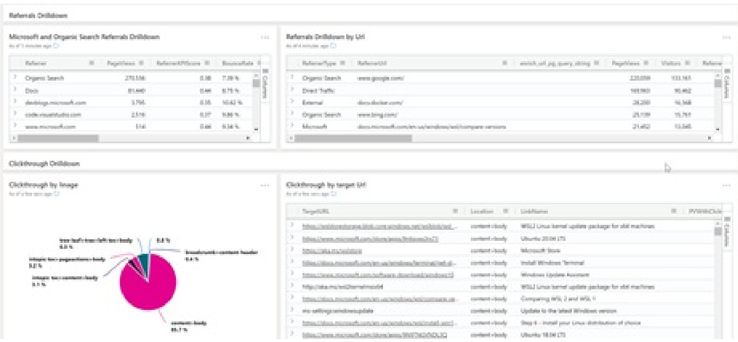
Redirected pages. Service and Product metadata values weren't preserved when redirect_document_id was set to false in the redirect.json file prior to November 2020. The code bug in the Metadata Store was fixed so these values are preserved moving forward.

New Trend and Drill-down dashboards. Try out these two dashboards available in the Content Performance report for a quick snapshot of your topic.

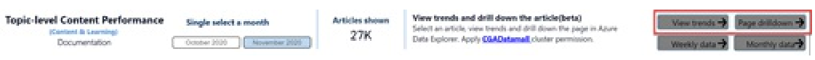
- **Trend dashboard**. Get weekly + monthly topic trends for latest 14 weeks or months.



- **Drill-down dashboard**. Get all the drill-down reports for your topic in one single view (CTR, Referrers, Verbatims, and GitHub issues). You can Customize **Time Range** like LastWeek, Last30Days, or choose Start and End by clicking the **Drilldown range** filter.



- **How to use**. Select a topic in either the Documentation or Reference pages, and then click one of the buttons in the top-right of the page.

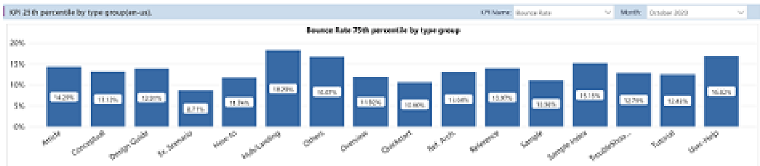


KPI benchmark charts. Located on a new KPI Benchmark page, these charts and table show the 25th and 10th percentile by Topic Types for all topics in the report-not by content set. Use the 25th or 10th percentile buttons and Month drop-downs to flip between core metrics across topic type groups to see the monthly trends.

Change to KPI score baseline. ~36K articles were moved from C&L reporting to BAG/Microsoft 365 (openspecs and Power platform). With fewer articles in C&L, the new competitive landscape changed and inevitably impacted article ranking. The impact mostly occurred with *Article*, *Conceptual*, and *Others Type* groups. We saw close to a 6-point decrease in the all-up KPI score for C&L (72.41% to 66.69%).

October 2020

- New benchmark chart added to the Overview page. It shows the 25th percentile by Topic Types for all topics in the report-not by content set. Use the KPI name and Month drop-downs to flip between core metrics across topic type groups to see the monthly trends. Note that this chart will likely move to a separate page in November. You can use this chart to:
    - Identify the bottom 25th percentile for a particular metric by type group and what you need to achieve to move out of the bottom.
    - Get a quick idea of how one topic type compares to another. It can help set expectations when changing your ms.topic to the appropriate ms.topic value.

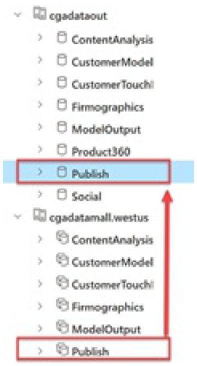
- Article assessment page. A Cluster Name drop-down was added, allowing you to switch between two Kusto clusters to find the one that resolves faster.

- Kusto cluster permissions. Many features within the Content Performance report require you to have cluster permissions for them to run. The CGA team owns the management of these permissions.
    - If you are in the Dev Relations organization, you can request access to both the cgadataout (cgadataout.kusto.windows.net) and the CGA Kusto Public (cgadatamall.westus.kusto.windows.net) clusters.
    - If you aren't a member of Dev Relations, you can request access to the CGA Kusto Public (cgadatamall.westus.kusto.windows.net) cluster.
- Excel reports. By request, the downloadable versions of the Docs monthly report were separated by repo (azure, win32-pr, windows-driver-docs, and open-specs_windows, and other) instead of high and low page views to make them faster to download.

September 2020

- **Referrer drill-down query**. This query is currently failing. As a workaround, you can cancel the query and instead run it by using the cgadataout cluster. Request access to the cgadataout cluster if you don't already have it (`cgadataout.kusto.windows.net`). There's an issue with the underlying function in the cgadatamall cluster and an active bug with cgasupport.



- **New colors to signal low performers**. We heard your feedback about the red and pink conditional formatting being too harsh and have changed to yellow to match the low performing category on the Overview page. Ranking within the bottom 10% is now dark yellow and ranking in the bottom 10- 25% is light yellow.

- **Additional metrics available**. You can now see metrics, such as click-through and exit rates, even when they aren't used to calculate the KPI score. In those cases, the value appears on a grey background. The extra data points can help you with sleuthing.

PrimaryKPIScoreRank	BounceRate	ClickThroughRate	DwellRate	ExitRate
4.88	29.98%	34.34%	4.17%	42.56%
2.36	38.59%	19.43%	1.25%	65.80%
65.38	13.81%	59.66%	7.67%	77.25%
8.49	24.73%	29.00%	4.17%	67.88%
70.14	10.4			15.27%

Dark yellow: bottom 10%
Light yellow: bottom 10-25%
Gray: Not included in KPI formula

- **New direct traffic details exposed**. A new column in the referral drill-down titled "enrich_url_pg_query_string" can help identify traffic coming from MSDN redirects or F1 Help.

ReferrerType ↑	enrich_url_pg_query_string	PageViews
Direct Traffic	?redirectedfrom=MSDN	175
Direct Traffic		6,350
Direct Traffic	?OCID=WinClient_Ver1703_Settings_DevMode	81,272

August 2020

***Historical data on the Overview page (Legacy report)**

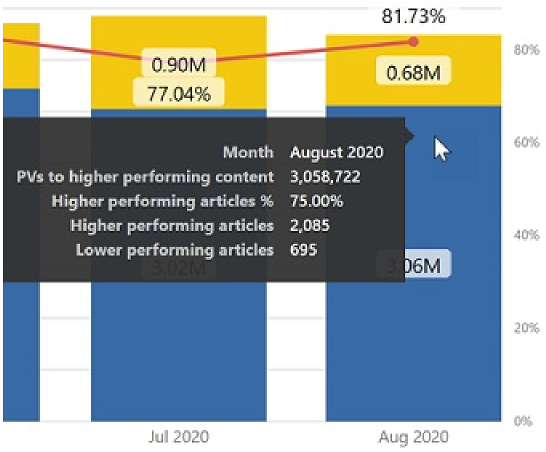
We needed to refresh the historical data for both June and July 2020 based on the following reasons:

- Root cause: The .md version of hub and landing pages, which are no longer live, were being calculated as part of the historical snapshot query for the Overview page (details are in the Docs Build v3 migration changes. The same page is using two topic IDs (one for .yml version and one for .md version). Both topic IDs are showing traffic, and the expected behavior is that the not-live version (.md file) shouldn't have any traffic. Although the traffic is small, it impacted the overall ranking.
- Impact: Overview tab-June and July only. Type group most impacted is Hub & landing. This update didn't impact the monthly data on Docs and Reference report pages.

- Short-term fix: Update snapshot query and refresh June and July data only on the Overview page.
- Long-term fix: CGA looking into root cause in the Kusto page view table.

Total # of articles

- On the Overview page, you can find the total number of articles for the low and high-performing groups and the % of articles that are high performing. The totals include both the docs and ref articles. To access, hover over the column to view the tooltip. This update helps align the article totals with the totals in the "Articles shown" total at the top of the Documentation and Reference pages.



Scope changes

- Moved the Windows Console topics from Microsoft 365 to the C+L report.
- Added Microsoft Teams topics to the C&L report in addition to keeping it in the Microsoft 365 report.

July 2020

- Added the Acrolinx clarity score next to the Acrolinx score. If the Clarity score is low, content owners can use recommendations in the report to improve their article.
- Added the MSAuthor column on the Documentation and Reference pages for easier content owner identification.
- Permissions update. To use many of the features in the content performance report, you need the right permissions. Based on CGA guidance, the permissions have changed to the cgadataoutmall cluster (public). Links to the permissions form are updated in the report and all the drill-down Kusto reports now point to this cluster (previously: cgadataout). This update should fix any "permissions denied" issues.
- Downloadable Excel reports available from Overview page. Added all primary metrics data for articles such as clickthrough and exist rates to the spreadsheets and added grey color formatting when the metric aren't used in score calculation. The added metrics might help content owners with more sleuthing. Also added columns that were recently added to the Power BI report that weren't in exported reports.
- Removed timestamp in the LastReviewed column to tighten up space.
- Article assessment page now has a link to how to use it.
- Expanded the scope for the GitHub OpenIssues link. Originally, we only provided the link for the articles that had open issues in the last 365 days.

June 2020

New and updated functions:

- Click-through report. Clicking the link in the **ClickThroughsDrilldown** column launches Azure Data Explorer and provides the number of clicks for each link by link name or target URL and its location on the page. See the Troubleshooting article
- Referrers report. Added CurrentPageKPIScore, ReferrerKPIScore, and KPIScoreDelta to the ReferralsDrilldown

function. Both scores are using the same formula according to the current topic type, and a positive KPIScoreDelta means the referral is good. You can leverage the Page Views and KPIScoreDelta in the ReferralDrilldown when you analyze the referrals. See the Troubleshooting article.

UI changes:

- Increased the font size by 1 pt in the table to make it more readable
- Removed "|Microsoft Docs" from Title column
- Moved OrganicSearchRate after "Visitors"
- Moved CSAT-related columns ahead of GitHub related columns
- Added the Help (?) icon to the Documentation and Reference pages to link to Metrics definitions.

Metrics definition change:

- Changed the **DwellTimeDelta** to **DwellRate** to measure the median dwell time as a percentage of estimated time to read. Formula is MedianDwellTime/Estimated reading time using a positive % to make it more explainable. The score for a topic changed to a positive value, but doesn't impact the score ranking. Original formula (MedianDwellTime – Estimated reading time)/Estimated reading time)

May 2020

- Expanded the content scope to cover all Topic types, including **How-to**, **Troubleshooting**, and **End-user-help**. See the latest score formula and grouping at Primary KPI Score.
- Expanded the content scope to cover all topic keys except /archive and /previous-versions, and moved learning path and learn module to MS Learn Report.
- The **Dynamics** and **Office** content were separated from **Content & Learning** scope, and new reports created for BAG Content (Dynamics 365, Power Platform, and Data Integration) and Microsoft 365 using the same metrics as C&L.
  - C&L: https://aka.ms/contentperformancedashboard
  - BAG Content: https://aka.ms/contentperformance-bag
  - Microsoft 365: https://aka.ms/contentperformance-m365
- Flipped the charts on the **Overview** page, showing the **Page Views to lower-performing articles** on the top of **Page Views to higher-performing articles** to make it more readable.
- Snapshot the monthly **Page View** to higher-performing, lower-performing articles on en-us by the dimensions defined in Overview filters to avoid the **% of PV** change from time to time when metadata gets changed.
- **Thousand Eyes Testing** traffic was excluded beginning on April 18, 2020, which caused 1.5M decrease in monthly page views for hub/landing pages.
- Added **Total Page Views** on the **By Service/By Product* page so we can leverage the volume of Page Views when evaluating the **% of PV to higher-performing articles** by Service or Product.
- Added **GitHubNewIssues** on the left of **GitHubOpenIssuesLink** on **Documentation** and **Reference** page to know how many new issues were created in the last month before clicking the GitHubOpenIssuesLink to view the detailed issues in GitHub.
- Added **RatingVerbatims** on the left of **RatingVerbatims** link on **Documentation** and **Reference** page to know how many **Verbatim** were created in the last month before clicking the **RatingVerbatims** and to view the detailed verbatim in Azure Data Explorer.
- Updated the **Canonical Functions** for Content Performance dashboard V2 data. Some query examples were updated at How to use Content Performance Data.

January 2020

- Some new canonical functions are available that you can use to drill down for a specific article

- ReferralsDrilldownByTopicKey: Drill down to the most important performance KPIs by referrer name, referrer domain, and referrer URL
  - HubLandingPageDrilldown: Drill down the content performance for the page opened for a specific hub-landing page
  - TopBeforeAndAfterPagesDrilldown: Provide a customer journey for the most popular pages viewed before and after a certain page, immediately, in the same session, and within last N days respectively.
- Article Assessment (Beta) is available as a new page on the Content Performance Dashboard so you can view the impact of your published changes for a single article. You specify the page, the publish date, and the weeks before and after publish for which you want to see the data. Then, when you run the report, you get the results to see whether your changes affected change in traffic, bounce, click-through, etc.

December 2019

- On the **Documentation** page and the **Reference** page:
  - Added GitHub issues links. When you click the link in the **GitHubOpenIssuesLink** column, another browser tab opens GitHub.com and displays the list of open GitHub issues for the article.
  - Added verbatim links. When you click the link in the **RatingVerbatim** column, another browser tab opens Azure Data Explorer and automatically runs a query to retrieve all new verbatim that were logged during the reporting period.
- The report UI has been cleaned up to remove redundant or confusing items and to highlight key resources for providing feedback and learning to use the report.
- The article type selector buttons have been moved to a standard filter on the right panel, and the tiny hard-to-decipher charts on the main page have been removed.
- The PrimaryKPIScore metric article in the contributor guide has been updated to list the article types covered by the content performance model. More information has been added to document how the model groups certain content types together based on factors beyond the ms.topic value.

November 2019

- Added the **Performance by Service/Product** page to provide a ranking of services and products by the percentage of page views that go to the higher performing content. You can expand the plus signs to view subservice and technology performance as well.
- The weekly and monthly trend reporting has been changed to use Azure Data Explorer (the web version of Kusto). Mac users and anyone without Kusto installed can now use the Kusto queries for content performance.
- Instead of having to follow 12 manual steps to obtain referrer details, you can now click an icon in the ReferralsDrilldown column of the Documentation, Learn, and Reference tabs to obtain the referrer information. The function behind this feature loads Azure Data Explorer, populates the query, connects to the right database, all in one click. More info

October 2019

The **reference-architecture** topic type is now available as the **Ref. Arch** filter button on the Documentation page.

- You can now filter by employee departments on all report pages using the **Department** filter.

- (BETA) Trend reporting for a given topic has been partly automated. You can now click the **Weekly/Monthly** button to start Kusto and get the trend lines for key metrics.

FAQs

Public/private repository pairs

In the RepoName filter, both public and private repositories are listed for repository pairs such as azure-docs and azure-docs-pr. If both appear for your repository, select them both to get a full view of the content performance data for that repository pair.

Test traffic

The `Thousand Eyes` testing tool introduced around 1.5M page views monthly on 30+ docs hub-landing pages and this traffic was excluded from 4/18/2020 onwards.

Data differences between Content Performance and Documentation dashboards

You might notice differences in data points between the Content Performance Dashboard and the Documentation Dashboard. The main difference is that the documentation dashboard includes data for all locales, while the content performance dashboard contains only EN-US data. Additionally, there are differences in how some of the data is calculated. For example, bounce is measured differently. For the Documentation Dashboard, we use the industry standard definition for bounce rate: the % of sessions on docs with only one page view, and time on site is less than 5 seconds. There is no session concept for page-level performance measurement. So, in the Content Performance report, we measure bounce rate based on page view behavior instead.

Connect to CGA Kusto data clusters

6/25/2021 • 2 minutes to read

Use Kusto to deep-dive into content data metrics. In this article, you learn how to request access to the Kusto data clusters and connect to them using client applications.

NOTE

Data clusters are supported by CGA (Cloud+AI Customer Growth Analytics). For more information, see CGA's website.

Request access to Kusto clusters

To use these resources, you must join the **CL Dashboard User Group** security group through MyAccess. The security group has been granted access to the **followercgadataout** cluster by the CGA team.

- For most users, no action is required. The entire C&L team, as well as any members of the previous Content Learning Dashboard User Group, have been added to the new security group.
- If you don't have access, you should request access to the new CL Dashboard User Group (links below).

Select the links in the table to submit a MyAccess request. A pre-populated form should appear. MyAccess links can be buggy though. If the pre-populated form does not appear, enter the hyperlink text into the MyAccess search (for example, "CL Dashboard User Group").

KUSTO CLUSTER	ROLE	NOTES
CGA Kusto Follower (followercgadataout.westus.kusto.windows.net) through CL Dashboard User Group	Read-only	**Required** for all users. The default cluster.
CloudMine-Data (https://1es.kusto.windows.net)	CloudMine-Data reader	**Required** for all users to view GitHub issue information in the Drill down dashboard
CGA Kusto Public (cgadatamall.westus.kusto.windows.net)	Read-only	**Optional** for all users. A general purpose public cluster for ad hoc queries

Connect to a Kusto cluster

You can connect to Kusto using many different applications. In this section, you learn how to connect using Kusto.Explorer (Windows only) and Data Explorer (web browser).

Before you attempt to connect to the Kusto cluster, make sure you have read-access to the clusters. For more information, see Request access to the Kusto data clusters

You should check to see if your reporting needs can be met by existing dashboards before querying Kusto directly. If your needs aren't met, help us improve by submitting a dashboard request.

Connect with Azure.Explorer

Azure.Explorer offers a rich desktop experience for Windows users to explore data in the Kusto Query Language. For more information on installation and setup, see Kusto.Explorer installation and user interface.

After you install Azure.Explorer connect to the Kusto cluster:

1. Select the **Connection** tab.

2. Select **Add connection**.

3. For **Cluster connection**, enter your cluster address.

   For example, `cgadatamall.westus.kusto.windows.net` or `followercgadataout.kusto.windows.net`

4. For **Security**, select `AAD -Federated`.

Connect with Data Explorer

Data explorer lets you query Kusto clusters directly from your browser.

1. In your browser, enter the address of the cluster you want to connect to.

   For example, `cgadatamall.westus.kusto.windows.net` or `followercgadataout.kusto.windows.net`

2. Your browser will redirect you to dataexplorer.azure.com.

3. After you authenticate with your Microsoft credentials, you will be taken to the interactive querying tool.

Next steps

In this article, you learned how to connect to the CGA Kusto clusters. To learn how to query docs content data, see How to use content performance canonical functions.

How to use content performance canonical functions?

6/28/2021 • 5 minutes to read

In this article, you learn about canonical functions that provide answers to commonly asked data questions. Canonical functions output data that has been processed using CGA's business logic and can reduce your reliance on manual data processing.

In this article you learn how to use the following canonical functions:

FUNCTION NAME	DESCRIPTION
ContentKPI functions	Aggregated content performance data for each topic
ReferralsDrilldown	Understand how users came to a given topic and the quality of those referrals
HubLandingPageDrilldown	Understand where users go immediately after visiting a hub or landing page
TopBeforeAndAfterPagesDrilldown	Understand other docs that a user went to before or after visiting a given page (user journey)

WARNING
CGA doesn't guarantee performance in their Kusto clusters. Permission requests to the cgadataout.kusto.windows.net cluster may be denied due to cluster management policies. CGA recommends using the content performance dashboard before querying Kusto directly. Create a dashboard request for needs that aren't served by existing dashboards.

Prerequisites

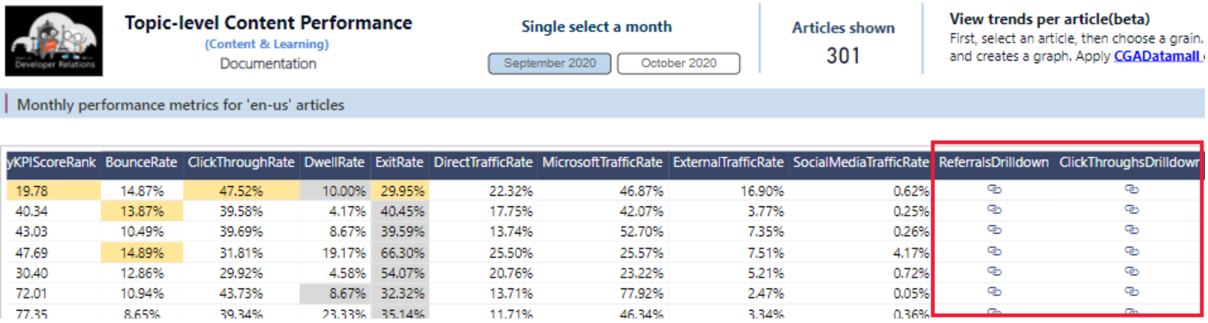
You must have access to the **cgadataout** or **cgadatamall** cluster. For more information, see Connect to CGA Kusto data clusters.

How to use the canonical functions

Canonical functions are stored in the **Publish** database of both the **cgadataout** and **cgadatamall** clusters. If you have read-access to the clusters, you have access to the functions.

Use the canonical functions in the following ways:

- Select the **Try-it** link above the code snippets in this article to run the query immediately in your browser.
- Copy the code snippets in a query client like Kusto.Explorer and run the query.
- Select the links in the Content Performance Dashboard for drill down queries.



You should update parameters and include filters to narrow the result-set. This improves performance and helps you avoid memory limits and timeouts.

ContentKPI functions

ContentKPI functions join the TopicMetadata table and fact tables. They aggregate the results by `TopicKey` or `ContentID`. You can set the temporal granularity to `Weekly` or `Monthly`. The functions return topic metadata and all the metrics reported in the Content Performance Dashboard.

The functions are available at cluster('cgadataout.kusto.windows.net').database('Publish'). And have the following names:

- ContentKPIByTopicKeyMonthlyV2
- ContentKPIByTopicKeyWeeklyV2
- ContentKPIByContentIdMonthlyV2
- ContentKPIByContentIdWeeklyV2

Example: Top 100 lower-performing articles by Page Views from a docset

Try-it

```
cluster('cgadataout.kusto.windows.net').database('Publish').ContentKPIByTopicKeyMonthlyV2
| where RepoName == 'azure-docs-pr'
| where Locale =='en-us'
| where Date == datetime(2020-04-01)
| invoke GetPrimaryKPIScoreV2()
| as T
| join kind=inner
(
    T
    | summarize Score25Pctile = percentile(PrimaryKPIScore, 25) by PageType, Date
) on PageType, Date
| extend IsLowerPerforming = PrimaryKPIScore < Score25Pctile
| where IsLowerPerforming
| order by PageViews desc
| extend Rank = row_number(1, prev(Date)!=Date)
| where Rank <= 100
| project    Date,
            TimeGrain,
            TopicKey,
            TopicType,
            Title,
            LiveUrl,
            PageViews,
            BounceRate,
            ClickThroughRate,
            DwellRate,
            ExitRate,
            OrganicSearchRate,
            DirectTrafficRate,
            PaidSearchRate,
            ExternalTrafficRate,
            MicrosoftTrafficRate,
```

```
        MicrosoftTrafficRate,
        SocialMediaTrafficRate,
        CSATHelpfulRate,
        CSATResponseRate,
        UserEngagementRate,
        StartTutorialRate,
        AvgCopyTryEngagement,
        DeployClickRate,
        Freshness,
        LinesChanged,
        LastCommitDateTime,
        GitHubOpenIssueCount,
        GitHubTotalIssueCount,
        DaysSinceLastReview,
        LastReviewed,
        Locale,
        RepoName,
        TranslationType,
        DevLang,
        MSProd,
        MSTechnology,
        MSService,
        MSSubService,
        Custom,
        Author,
        MSAuthor,
        Manager,
        Bounces,
        CSATHelpfulRatings,
        CSATTotalResponses,
        GitHubNewIssueCount,
        ClickThroughs,
        CopyTryClicks,
        EstimatedSecondsToRead,
        Exits,
        FirstSteps,
        MedianDwellTimeSeconds,
        OrganicSearchReferrals,
        TutorialLinkClicks,
        Visitors,
        RatingVerbatims,
        PrimaryKPIScore,
        PageType
```

ReferralsDrilldownByTopicKey

The ReferralsDrilldownByTopicKey function drills down on content metrics based on where a user came from. It shows how different users interacted with page depending on how they got to the page.

Users are grouped by `ReferrerType`, `ReferrerDomain`, and `ReferrerUrl` for a given `TopicKey` by weekly or monthly. The function returns all referrals with more than 10 page views.

Parameters

Update the following parameters to narrow your query:

- TargetTopicKey: *TopicKey* for a target page, you can copy this from the Content Performance Dashboard
- StartDate, EndDate: *datetime*, the start and end date you want to track. For example `datetime(2019-10-01)`, `datetime(2019-12-1)`
- TimeGrain: *string*, `month` or `week`

Example: Evaluate referral performance

Here is an example to drill down on `https://docs.microsoft.com/en-us/azure/java/` and exclude `Direct Traffic` referrals. Sort by `PageViews` and evaluate the performance for different referrer links.

Here is a screenshot for the data return by the function, you can try-it and check more KPIs.

	ReferrerType	ReferrerDomain	ReferrerName	ReferrerUrl	PageViews	Visitors	BounceRate	ExitRate	ClickThroughR...	StartTutor
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/	1224	851	0.121732026143791	0.14705882...	0.6781045751...	0.016339
azure/java/	Organic Search	www.google.com	google	www.google.com/	925	600	0.105945945945946	0.25621621...	0.6713513513...	0.008648
azure/java/	External	github.com		github.com/Azure/azure-sdk-for-java	363	204	0.154269972451791	0.25895316...	0.6363636363...	0.002754
azure/java/	Microsoft	azure.microsoft.com		azure.microsoft.com/en-us/develop/java/	209	136	0.076555023923445	0.18660287...	0.7033492822...	0.014354
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/app-service/app-service-web-get-started-java	161	115	0.0807453416149068	0.08695652...	0.7701863354...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-in/azure/	92	62	0.16304347826087	0.08695652...	0.5869565217...	0.032608
azure/java/	External	github.com		github.com/azure/azure-sdk-for-java	49	26	0.163265306122449	0.12244897...	0.6938775510...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/app-service/containers/quickstart-java	49	38	0.0816326530612245	0.20408163...	0.6938775510...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/java/spring-framework/	37	23	0.135135135135135	0.13513513...	0.6486486486...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/java/azure/eclipse/azure-toolkit-for-eclipse	32	23	0.09375	0.125	0.71875	
azure/java/	External	www.azul.com		www.azul.com/downloads/azure-only/zulu/	27	16	0.111111111111111	0.11111111...	0.7407407407...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/java/api/overview/azure/	26	17	0.0384615384615385	0.15384615...	0.9230769230...	
azure/java/	External	dzone.com		dzone.com/	26	22	0	0.38461538...	0.6923076923...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/java/java-sdk-add-certificate-ca-store	25	11	0.04	0.08	0.72	
azure/java/	Microsoft	azure.microsoft.com		azure.microsoft.com/en-us/tools/	24	13	0.125	0.29166666...	0.4166666666...	0.041666
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/java/azure/eclipse/azure-toolkit-for-eclipse-create-hello-wo...	24	15	0.166666666666667	0.20833333...	0.5416666666...	0.041666
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-java	24	12	0.208333333333333	0.04166666...	0.4583333333...	

[Try-it](#)

```
cluster('cgadataout.kusto.windows.net').database('Publish').ReferralsDrilldownByTopicKey(
    TargetTopicKey = '0981839c-30f2-6a8a-2b38-2078eb69b302',
    StartDate = datetime(2019-10-01),
    EndDate = datetime(2019-10-31),
    TimeGrain='month')
 | where ReferrerType <> 'Direct Traffic'
 | order by PageViews desc
```

HubLandingPageDrilldown

The HubLandingPageDrill down shows the next page that a user visited immediately after visiting a hub landing page.

Use this query to assess link quality on hub-landing pages.

Parameters

- HubLandingPageUrl: *Url*, hub-landing page url, for example `https://docs.microsoft.com/en-us/azure/`
- StartDate: *datetime*, such as datetime(2019-10-01)
- EndDate: *datetime*, such as datetime(2019-10-31)

Example

The following example drills down on the Azure hub-landing page 'https://docs.microsoft.com/en-us/azure/'. We can find:

- The most popular links on the Azure Home page
- The bounce rate, view length, and CTR on each page opened

HubLandingPageVisitors	ToNextPageVisitors...	NextPageTopicType	NextPageUrl	NextPagePV	NextPageVisitors	NextPageA...	NextPageBounceRate	NextPageCTR
81618	0.0615550491313191	landing-page	https://docs.microsoft.com/en-us/azure/virtual-machines/windows/	6830	5024	10	0.103660322108346	0.7008784773
81618	0.0546080521453601	guide	https://docs.microsoft.com/en-us/azure/architecture/guide/	5849	4457	13	0.123097965464182	0.5467601299
81618	0.0519738293023598	article	https://docs.microsoft.com/en-us/azure/guides/developer/azure-developer-guide	5827	4242	15	0.124935644413935	0.5050626394
81618	0.0454679114901125	landing-page	https://docs.microsoft.com/en-us/azure/virtual-machines/linux/	5032	3711	9	0.131955484896661	0.6649443561
81618	0.0419515302996888		https://docs.microsoft.com/en-us/learn/azure/	4399	3424	12	0.0572857467606274	0.7458513298
81618	0.0297728442255385	hub-page	https://docs.microsoft.com/en-us/azure/devops/	3406	2430	10	0.0819142689371697	0.6931884908
81618	0.0285108677007523	hub-page	https://docs.microsoft.com/en-us/azure/index	3792	2327	22	0.0342827004219409	0.7850738396
81618	0.0284251023058639	landing-page	https://docs.microsoft.com/en-us/azure/app-service/	3252	2320	15	0.0916359163591636	0.6273062730
81618	0.0264524982234311	landing-page	https://docs.microsoft.com/en-us/azure/architecture/	3036	2159	14	0.077733860342556	0.7025691699
81618	0.0255335832781984	landing-page	https://docs.microsoft.com/en-us/dotnet/azure/	2732	2084	12	0.0962664714494876	0.6401903367
81618	0.0245043985395378	overview	https://docs.microsoft.com/en-us/azure/sql-database/	2700	2000	16	0.0907407407407407	0.6325925925
81618	0.0244186331446495	landing-page	https://docs.microsoft.com/en-us/azure/python/	2734	1993	10	0.131309436722751	0.5790051207
81618	0.0222009850768213	landing-page	https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/	2694	1812	18	0.0805493689680772	0.7041573867
81618	0.0218579234972678	landing-page	https://docs.microsoft.com/en-us/azure/backup/	2337	1784	12	0.106118955926401	0.6110397946

[Try-it](#)

```
cluster('cgadataout.kusto.windows.net').database('Publish').HubLandingPageDrilldown(
    HubLandingPageUrl = 'https://docs.microsoft.com/en-us/azure/',
    StartDate = datetime(2019-10-01),
    EndDate = datetime(2019-10-31))
```

TopBeforeAndAfterPagesDrilldown

TopBeforeAndAfterPagesDrilldown shows a customer journey from the pages viewed before and after viewing a target page. This function returns results for 3 different time periods:

- ImmediatelyBefore/After: How many visitors viewed a given page immediately before and after visiting the target page. Pages with high visits indicate a strong relationship with the target page.
- SameSessionsBefore/After: How many visitors viewed a given page during the same session. Pages with high visits are correlated with the target page, since many users viewed them in the same session.
- NDaysBefore/After: How many visitors viewed a given page within `N days` before, or after, viewing the target page. This data suggests that these pages may be related.

Parameters

- TargetUrl: *Url*, the page you want to evaluate, such as `https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-your-first-function-visual-studio`
- StartDate, EndDate: *datetime*, such as datetime(2019-10-01), datetime(2019-10-31)
- DaysBeforeAndAfter: *int*, N days before or after, for example 30 here means last 30 days before, and next 30 days after viewing current page
- TopNUrls: *int*, the Top page with most visitors, 100 means top 100 pages by its visitors

Example

This example drills down on the following page `https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-your-first-function-visual-studio`.

TopBeforeOrAfterUrls	VisitorsImmediatelyBefore ▼	VisitorsImmediatelyAfter	VisitorsSameSessionBefore	VisitorsSameSessionAfter	VisitorsNDaysBefore	VisitorsNDaysAfter
https://docs.microsoft.com/en-us/azure/azure-functions/	2337	115	2698	746	3484	1819
https://www.google.com/	1655					
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-azur...	546	10	833	356	1536	956
https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview	421	123	933	464	1669	1077
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-func...	347	209	436	519	769	813
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-func...	327	216	431	385	663	645
https://docs.microsoft.com/en-us/azure/azure-functions/functions-develop-vs	279	321	770	722	1555	1679
https://docs.microsoft.com/en-us/azure/azure-functions/index	189		208		278	
https://docs.microsoft.com/en-us/dotnet/azure/	177		196	89	381	267
https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale	165	79	297	174	789	837
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-function-...	140		228	79	463	289
https://docs.microsoft.com/en-us/azure/azure-functions/functions-add-output-bin...	137	690	273	652	473	977
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]

Try-it

```
cluster('cgadataout.kusto.windows.net').database('Publish').TopBeforeAndAfterPagesDrilldown(
    TargetUrl = 'https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-your-first-
function-visual-studio',
    StartDate = datetime(2019-10-01),
    EndDate = datetime(2019-10-31),
    DaysBeforeAndAfter = 30,
    TopNUrls = 100)
```

Contacts

- If you have any question regarding there functions, contact Edison Dai. You can create a ticket using the dashboard data requests process. @mention msmbaldwin in the work item for awareness.

Next steps

In this article, you learned how to run canonical functions in CGA's Kusto clusters.

Learn how to write your own queries, by looking at example queries submitted by our content-community. For more information, see Community Kusto queries.

Community Kusto queries

5/10/2021 • 4 minutes to read

In this article, you learn about community-submitted Kusto queries that other contributors have found useful.

These queries are not officially supported and are intended to help you understand different Kusto queries. For a list of supported queries, see How to use canonical functions.

Prerequisites

You must have access to the **cgadataout** or **cgadatamall** cluster. For more information, see Connect to CGA Kusto data clusters.

Queries

Trends over time for a particular URL

Run the query in Lens Explorer

```
let startDate = datetime(2020-1-1);
let endDate = datetime(2020-5-1);
cluster("Cgadataout").database("CustomerTouchPoint").PageView
| where StartDateTime between(startDate .. endDate)
| where Url == "https://docs.microsoft.com/en-us/azure/architecture/"
| summarize Visits=dcount(enrich_session_id), PageViews = count(), FromSearch = countif(ReferrerType has
'Search'), NotSearch = countif(ReferrerType !has 'Search'), Bounces = countif(IsBounce == 1) by
bin(StartDateTime, 1d)
| render timechart
```

Low-to-zero-visits query

Run the query in Lens Explorer

You can change the matchURL variable to narrow down to a particular docset. Start and end dates determine the time window. The results are ordered by PageViews descending, so you can see what got no views in a given time range.

```
let urlPattern="https://docs.microsoft.com/en-us/sql/";
let allLivePages =
cluster('cgadataout.kusto.windows.net').database("CustomerTouchPoint").TopicMetadata
| where LiveUrl contains urlPattern
| where IsLive == 1
| where isempty(RedirectUrl)
| project TopicKey, TopicType, Title, LiveUrl;
let viewedPages =
    cluster('cgadataout.kusto.windows.net').database('Publish').ContentKPIByTopicKeyMonthlyV2
    | where LiveUrl contains urlPattern
    | where Date > ago(60d)
    | project TopicKey, LiveUrl, PageViews;
allLivePages
| join kind = leftanti (viewedPages) on TopicKey
```

Page views and visits for specific services in last N days

```
cluster("Cgadataout").database("CustomerTouchPoint").PageView
| where Site == "docs.microsoft.com" and Locale == "en-us"
| where StartDateTime >= ago(90d)
| where MSService contains "sentinel"
| summarize PageViews = count(), Visits = dcount(enrich_session_id) by Url
```

All content for Content & Learning published between two dates

Use these queries to find all content published for tech docs, Learn, and the architecture center.

The output will contain three columns - all articles, learn, and architecture. To determine the tech doc totals, add a column to calculate the total using this formula: `=b2-(c2+d2)`.

For reporting, we usually separate reference from conceptual. For standard reporting, the following types are considered reference:

- managed-reference
- reference
- language-reference
- error-reference

Updated articles: all articles that existed prior to the first date specified in the query and that were updated between the specified dates.

```
cluster("Cgadataout").database("CustomerTouchPoint").TopicMetadata
| where Site == 'docs.microsoft.com' and IsLive == '1' and Locale=='en-us'
| where TopicType != 'language-reference' and TopicType != 'managed-reference' and TopicType != 'reference'
and TopicType != 'error-reference' and TopicType !='page-not-found' and TopicType !='portal' and TopicType
!='struct' and TopicType !='interface' and TopicType !='enum' and TopicType !='callback' and TopicType
!='function' and TopicType !='class' and TopicType !='method'and TopicType !='ioctl' and TopicType
!='structure' and TopicType !='enumeration'
| where startofday(LastReviewed) between(datetime(2019-02-01) .. datetime(2019-02-28))
| where startofday(FirstPublishDateTime) !between(datetime(2019-02-01) .. datetime(2019-02-28))
| summarize updatedTotal=dcount(TopicKey),
    updatedLearn=dcountif(TopicKey, LiveUrl matches regex "^http?s://docs.microsoft.com/([^/]*)/learn/"),
    updatedAAC=dcountif(TopicKey, LiveUrl matches regex "^http?
s://docs.microsoft.com/([^/]*)/azure/architecture/")
    by Product,Service
```

New articles: all articles published for the first time between the specified dates.

```
cluster("Cgadataout").database("CustomerTouchPoint").TopicMetadata
| where Site == 'docs.microsoft.com' and IsLive == '1' and Locale=='en-us'
| where TopicType != 'language-reference' and TopicType != 'managed-reference' and TopicType != 'reference'
and TopicType !='page-not-found' and TopicType !='portal' and TopicType !='struct' and TopicType
!='interface' and TopicType !='enum' and TopicType !='callback' and TopicType !='function' and TopicType
!='class' and TopicType !='method'and TopicType !='ioctl' and TopicType !='structure' and TopicType
!='enumeration'
| where startofday(FirstPublishDateTime) between(datetime(2019-02-01) .. datetime(2019-02-28))
| summarize updatedTotal=dcount(TopicKey),
    updatedLearn=dcountif(TopicKey, LiveUrl matches regex "^http?s://docs.microsoft.com/([^/]*)/learn/"),
    updatedAAC=dcountif(TopicKey, LiveUrl matches regex "^http?
s://docs.microsoft.com/([^/]*)/azure/architecture/")
    by Product,Service
```

Courtesy of: Presley/Jian

All abandoned articles in Content & Learning content sets

```
let startDate = ago(30d); //Start date of page view window
let endDate = now(); // End date of page view window
let TopicsTable = cluster("Cgadataout").database("CustomerTouchPoint").TopicMetadata //scopes to only
content owned by C+L
| where Site == 'docs.microsoft.com' and IsLive == '1' and Locale=='en-us' //scopes to docs site and en-us
content
| where LastReviewed < ago(90d) //set datetime to 90 or 120 days ago to list all stale and/or abandoned
content
| where TopicType !in ("managed-reference", "reference") //excludes most reference content from this query
| where Pillar == "Azure"; //choose the pillar to scope results to avoid exceeding the query record limit
let cachedPageViewTable = materialize(cluster('Cgadataout').database("CustomerTouchPoint").PageView
| where Site == "docs.microsoft.com" and Locale == "en-us"
| where IsMSInternalTraffic != true and enrich_rip_isp != "microsoft corporation"
| where StartDateTime between(startDate .. endDate)
| project StartDateTime, TopicKey, enrich_session_id, IsBounce, PageViewId, Title, Site, Locale, TopicType,
Url, IsMSInternalTraffic, enrich_rip_isp, LinkClickEvents, LinkClickEventsJson, VisitorId);
let metricsByTopicKey = cachedPageViewTable
| summarize PageViews = count(), Visits = dcount(enrich_session_id),  Bounces = countif(IsBounce == true)
//, Title = any(Title), TopicType = any(TopicType), Url = any(Url)
by TopicKey
| extend ['Bounce %'] = round(Bounces * 100.0 / PageViews, 1);
let output = TopicsTable
| join kind = leftouter (
metricsByTopicKey
) on TopicKey
| project
  Locale
, Service
, Author
, TopicType
, Title
, LastReviewed = format_datetime(LastReviewed, 'yyyy-MM-dd')
, LiveUrl
, Visits = iff(isnotempty(Visits), Visits, 0)
, PageViews
, Bounces
, ['Bounce %'] = strcat(tostring(iff(isnan(['Bounce %']), todouble(0), ['Bounce %'])), '%')
| sort by Visits desc;
output
```

Next steps

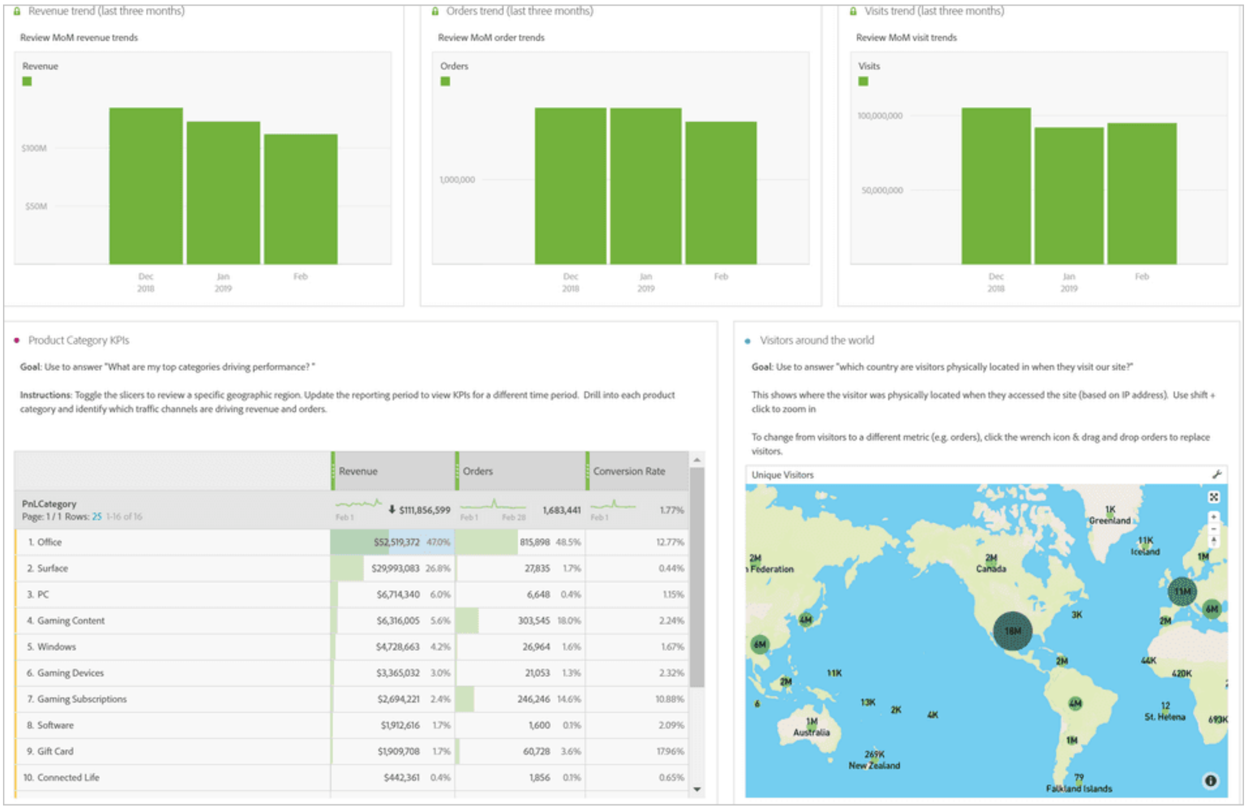
For more sample queries using the Kusto Data Out, see CGA Kusto Cluster "CGADataOut" Data Models.

Adobe Analytics

6/24/2021 • 2 minutes to read

Contributors can use the Adobe Analytics platform to explore web analytics for Microsoft domains. This is available to all Microsoft employees, free of charge.

For more information, see Get started - Microsoft Adobe Analytics.



Adobe Analytics vs Kusto and PowerBI

Adobe Analytics is an alternative analytics platform to Kusto and PowerBI that contributors can use to generate reports and explore web metrics.

Adobe Analytics uses the same JSLL (Java Script Logging Library) tagging system that feeds into the CGA reporting Kusto clusters that provide the data for most official C&L PowerBI reports. As a result, the data across platforms should be comparable, though data discrepancies may appear due to differences in data processing.

Metadata and global markets

Metadata information such as ms.author, ms.service, ms.technology, etc. are not available in Adobe Analytics. These metadata values **are** available in Kusto, and their downstream PowerBI reports, due to additional processing that occurs in the Kusto cluster data pipeline. As a result, Adobe Analytics queries cannot use metadata values to filter datasets, you must rely on URL patterns in Adobe Analytics.

Many official C&L PowerBI reports only display metrics from the `en-us` lang-locale. Although global data is available in the Kusto clusters, they require custom Kusto queries to. You can use Adobe Analytics to query data on global markets without learning the Kusto Querying Language.

Low barrier to entry

Unlike Kusto, you don't need to learn a new querying language to use Adobe Analytics. Instead, you create queries with a visual explorer to filter and transform data.

For more learning resources, see Adobe Analytics onboarding videos and guided learning.

Support

For help and support, reach out to the Adobe Analytics support alias, AskMarTech@microsoft.com.

Training and office hours are also available. For more information on dates and events, see Training & Events.

How to view and act on customer verbatim comments

3/5/2021 • 4 minutes to read

To find the verbatim comments for the articles you own, use one of these methods:

- For a single article, use the Microsoft Docs Metrics report (Preview) **Details** link to view an Azure Data Explorer dashboard filtered to the article you were looking at, and the time period you had selected: one week or one month. You can update the dates in the query to see more verbatims.
- The new preview version of the Verbatims report https://aka.ms/DocsVerbatimsReport shows verbatim comments for all content in docs. You can filter it by author, product, service, and other values, to show the respective verbatim comments. For more information, see Metrics: Verbatims report.

The Verbatim reports use an ML model that categorizes the verbatim comments. See Verbatim categories in this article for descriptions of those categories and subcategories. In those reports, the ML-determined subcategory is shown next to the verbatim. Both reports also show international verbatim comments machine translated to EN-US. The accuracy of the ML model for categorizing verbatim comments isn't 100% accurate. We have a backlog item to improve the machine-learning categorization.

What customers expect when they leave feedback

Understanding customer needs is key to building great content and helping drive adoption and customer satisfaction. Because of scale, this understanding is always limited – but we can do a lot to improve the "signal" with data in ways that are clear and actionable. One of the many channels for better understanding our customers is through customer feedback verbatims.

Customers' biggest expectation in the feedback loop is that we address the issues expressed in the verbatim. Feedback and comments can also be directly actionable for content owners, localization teams, and site engineering teams. We need to act on feedback, if possible. If not, customers will cease to give feedback. Unfortunately the verbatim comments are anonymous, so you can't respond directly to a customer comment.

Few examples of the broad categories of customer feedback verbatim comments include:

REQUEST	EXAMPLE
Request for more information	- Please explain like you would to a kid what Microsoft Authenticator does!(**Multi-Factor Authentication**) - Any tips on how to do repository authentication? Can't seem to find any docs on it. (**Service Fabric**) - How many IPs one can add to one NIC? How many NICs one can add to a machine? (**Virtual Network**)
Requests for specific fixes	- you should update the correct installation link to the sap NetWeaver library.(**Power BI**)
Requests for specific examples	- Would be nice to have an example demonstrating interop for custom struct types and methods for a library, for example, gmp, gtk etc. (**.NET Core**)

REQUEST	EXAMPLE
International experience	- WHY I SEE SQL COMMANDS TRANSLATED IN ITALIAN??? (**SQL Server**) - Please display Microsoft sites in English when I navigate to English sites, don't magically transform them to Dutch without asking for it.(**Azure Active Directory**)

Impact of verbatim fixes and responsiveness on CSAT

Analysis show that addressing an actionable verbatim on a timely manner has a direct correlation with article CSAT value. The more customer feedback verbatim we address on a timely manner, the highest the article satisfaction rate will be. The following two figures show the impact of addressing actionable verbatim on customer satisfaction. The green shade is the actual CSAT value of the article while the yellow shade shows the predicted CSAT. If all the actionable verbatim comments for an article are addressed, we'd expect to see a **2%-47.8%** increase in CSAT. The improvement depends on the count of actionable verbatim comments for the article during the week you're measuring.

Verbatim categories

Our content gets more than 1M verbatim comments a year from customers. We would need extra resources to read and address all of this feedback. To help, we use a machine learning text classification tool that autogroups verbatim comments into 22 predefined subcategories of issue types, which we have grouped into categories. The following table describes each verbatim category.

CATEGORY	SUBCATEGORY NAME	DESCRIPTION
Form or Presentation	Presentation-StyleVisuals	The user is commenting about the look of the website.
	Presentation-BrokenLink	The article contains broken links.
	Site-CannotFind	User cannot find what they are looking for.
	Site-Feature	Comment about the website.
	Site-FeedbackControl	Comment about the feedback control on the website.
Content	Content-Accuracy	The article content is inaccurate.
	Content-BadExample	The example is not working or inappropriate.
	Content-Incomplete	The content is incomplete.
	Content-NeedExample	The article needs an example.
Localization	Loc-BadMT	The machine translation is inaccurate.
	Loc-HelpExperience	The localized content is hard to use.

CATEGORY	SUBCATEGORY NAME	DESCRIPTION
	Loc-NeedsLoc	The topic needs translation.
	Loc-Other	Miscellaneous localization issues that do not fit any other subcategory.
	Loc-PrefersENU	The user prefers English language content.
	Loc-Translation	The translation is inaccurate.
Product	Product-Experience	The comment is about the product user experience.
	Product-International	The comment is about the product from a foreign perspective.
	Product-Link	A download link is broken.
	Product-Sentiment	General product feedback.
	Product-SubscriptionSupport	Feedback about the subscription model or usability.
Kudos	Kudos	The article was helpful.
Other	Other	The comment does not fit any of the other subcategories.
Rant	Rant	Offensive comment, spam, or other remarks that are not actionable.

Next steps

Learn more about the preview version of the Verbatims report.

Have feedback about the preview version of the Verbatims report? Please enter it in this Verbatims Report Feedback Excel file.

Request enhancements to data, dashboards, or queries

6/24/2021 • 3 minutes to read

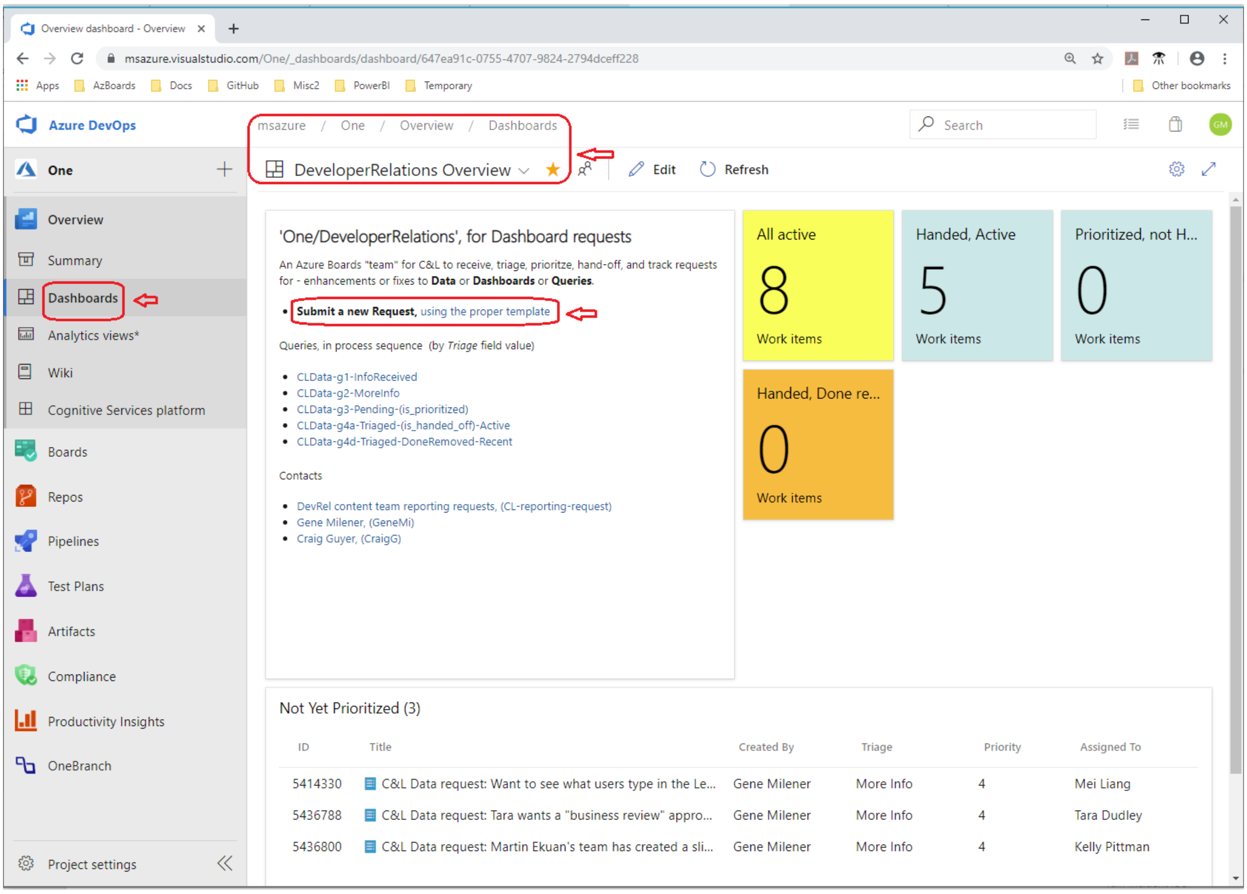
Anyone in DevRel can request enhancements or updates to the reports and dashboards that are maintained by the DevRel Analytics team. To submit a request, go to https://aka.ms/dashboard-requests. Requests can target the following items:

- **Dashboards**: only those dashboards maintained by the Analytics team or CGA.
- **Kusto queries**: to populate any dashboard.
- **Data**: the data available to Kusto queries.

The Content Data Advocacy Group (CDAG) triages the requests. Approved requests are implemented by the DevRel Analytics team or CGA (Customer Growth Analytics).

Use our dashboard

To enter your idea, or to view queries for submitted requests, go to https://aka.ms/dashboard-requests. This Azure DevOps dashboard is at the hierarchy **msazure / One / Overview / Dashboards**:



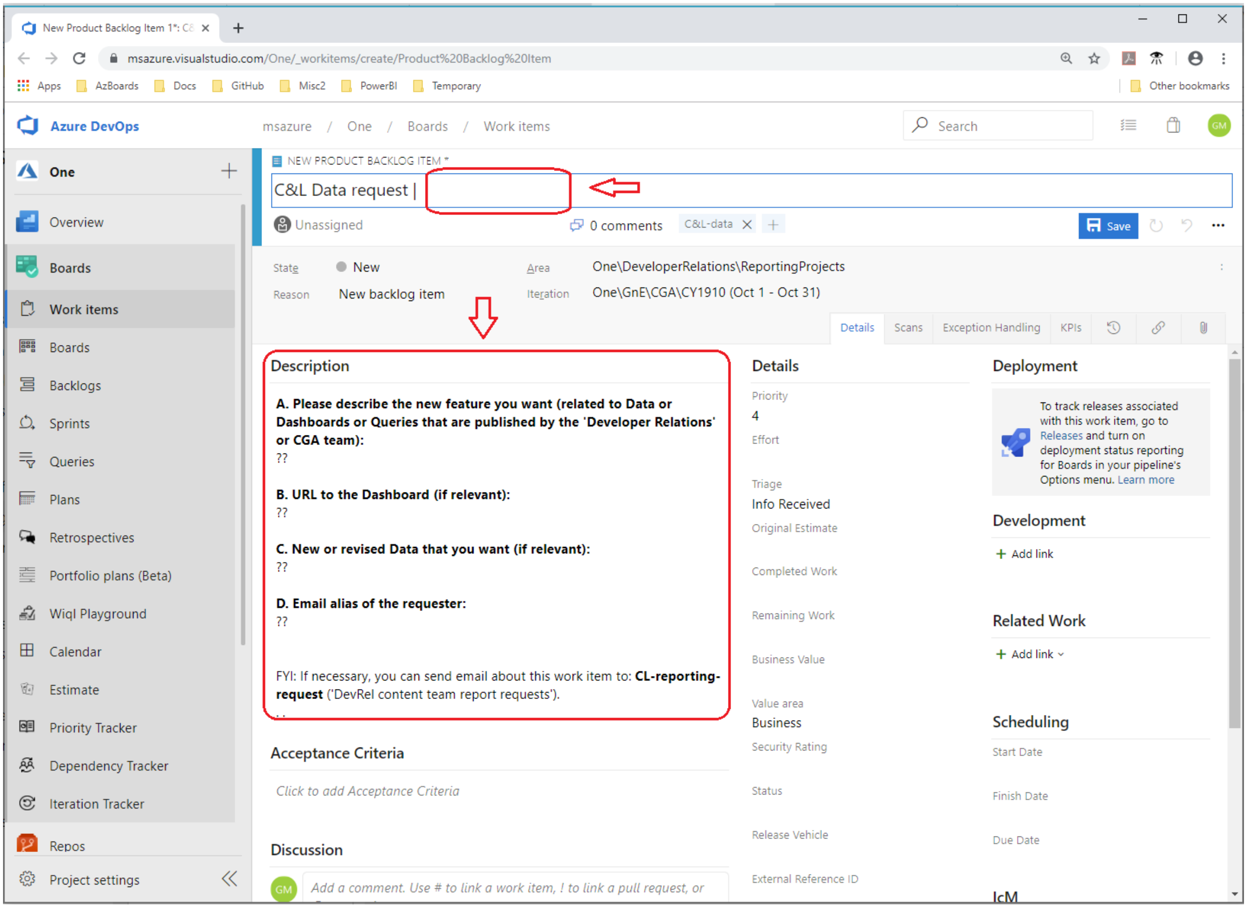
Use the `Submit a request` link

Clicking the **Submit a request** link invokes the proper template, of type **Product Backlog Item**. To complete the template, edit only the following fields:

- **Title**: Append your words to the given prefix.
- **Description**: For each prompt, replace the `??` with your information.
- Optionally, you may add an attachment, such as a screenshot.

Do not edit any other fields.

Finally, save your request. Optionally, you can click **Follow** to be notified when the item changes.



Triage

When a request is submitted, its **Triage** field is set to *Info Received* and the **Priority** is set to *4*. A request moves through the following triage states:

1. Info received (not triaged)
2. More info (not triaged)
3. Pending (not triaged)
4. Triaged

Two times per month at regular Content Data Advocacy Group (CDAG) meetings, new items are reviewed, prioritized, and the **Triage** state is updated. This meeting is open to all CDAG, including members from Content Contributor Success (CCS), Docs, Learn, Patterns and Practices (PnP), and Operational Excellence and Governance (OE&G). Occasionally, CDAG will request additional info from managers of the content teams or from Mei Liang and the Analytics team in order to triage the request.

After an item is prioritized and triaged, the Analytics team handles implementation of the request. CDAG interacts directly with the DevRel Analytics team. If an item requires work by CGA, the Analytics team handles the request and communication with CGA, and reports back to CDAG and the original requestor in the work item. CDAG and the original requestor do not need to contact CGA directly.

If a member of the Analytics team is present at the triage meeting with CDAG, then items can be handed off and assigned immediately; otherwise, CDAG will @mention Mei or other members of the Analytics team or will send email to inform them of new items that have been traiged and are awaiting assignment.

After the Analytics team completes the request, they update the **State** field to *Done* or *Removed*.

Who

As of October 2020, the following people or groups fill relevant positions to the Dashboard Request process:

POSITION	WHO	TASKS
Process coordinator	CDAG (cdag@microsoft.com)	• Maintains this article. • Reviews new requests for clarity. • Holds bi-weekly meetings where triage occurs. • Updates requests during the meetings. • Tracks handed-off requests that missed their implementation target dates. • Maintains our dashboard, and its queries. • Advertises the process to writers.
Content manager sponsor	Jill Reinauer (jillfra)	• Helps decide priorities among the new requests.
Analytics Representative	Mei Liang (mliang)	• Accepts or rejects requests during hand-off meeting. • Sees that accepted requests are implemented.

Other types of requests

- *SiteHelp*: For bugs, such as on Docs
- *Microsoft Forms*: Feature requests to Docs and Build
- *ListeningPost*: For process, policy, and other ideas
- *ListeningPost--Anonymous*: For process, policy, and other ideas
- *Doc Operations*: Cory Fowler (as of October 2019)

Getting Data on API Documentation Content

5/12/2021 • 2 minutes to read

We want the API documentation experience to be as data-driven as possible, and created the API documentation data dashboard. That page is your one-stop shop for everything that has to do with usage of API resources on docs.microsoft.com.

https://msit.microsoftstream.com/embed/video/d69cebfb-f32f-47de-91da-aa504552fb66?autoplay=false&showinfo=true

Getting Data for Your API Documents

We understand the data presented in the dashboard covers a more "global" view of the API documentation space, and you are interested in knowing how *your* content is performing. To do that, you can use the flexibility of Kusto and Lens Explorer. In this document, we are outlining some sample queries that you can run to get information about API docs performance.

To get started, you can use information from the dashboards we already have. See this video for a short intro guide on how to get the data for a specific documentation page:

https://msit.microsoftstream.com/embed/video/c6c853ee-dc9c-45aa-b605-8b935fda37c4?autoplay=false&showinfo=true

You can download Kusto Explorer yourself and try many of the possible queries to dissect the data. The query given in the video is listed here - try it!

```
PageView
| where Site == "docs.microsoft.com"
| where StartDateTime >= startofday(ago(9999d))
| where Url contains "https://docs.microsoft.com/en-us/dotnet/api/system.string.format"
| where enrich_url_pg_query_string !contains "traffic-source=cats"
| extend event_time=tostring(startofmonth(StartDateTime))
| extend event_time_reduced = todatetime(extract("\\d{4}-\\d{2}-\\d{2}",0,event_time,typeof(string)))
| summarize pageviews=dcount(PageViewId,2), visitors=dcount(VisitorId) by event_time_reduced
| sort by event_time_reduced asc
| render timechart
```

If you want an example description of how to read one of the many existing dashboards that we have in the API documentation data dashboard, take a look at our guide: How To: Interpret API Browser Data.

Issues or Missing Data?

Contact the Docs Support team - we'll be happy to address any issues, and route them appropriately.

How to perform customer research

6/24/2021 • 4 minutes to read

This article outlines how we prioritize and complete customer research on the Developer Relations Content and Learning team. It explains:

- Why customer research is important
- How to know if you need to do research
- How to complete customer research

Why is customer research important?

"You can't build a product without understanding the problem you're solving and the people for whom you're solving it. Various forms of research are the best way of understanding people who aren't you. It's really as simple as that."- Laura Klein, Author of *UX for Lean Startups*. Customer research keeps us customer focused and helps us uncover our customers' needs. More importantly, research helps us focus the *right* problems.

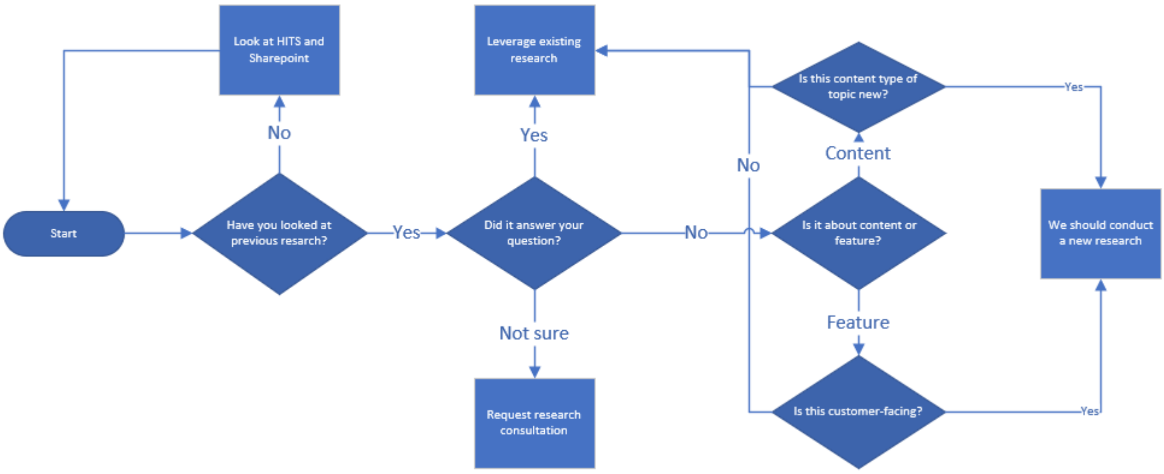
Types of Research

There are three main types of research we might conduct to get feedback on content and features on docs.microsoft.com:

- **Generative research:** Research that helps inform site-wide changes or broad strategic initiatives. Generally, this research tells us who are users are and what goals and behaviors they have.
  - Examples: content model research, ecosystem research, behavioral archetypes project
- **Feature research:** Research on adding new features to the site.
  - Examples: Learn personalized homepage, GitHub Issues research; notifications on Docs, tooltips
- **Content research:** Research that a content developer or docset owner does to help them better organize their content. This research helps us understand what customers are looking for and determines if our content is easy to find.
  - Examples: Topic landing page study, Windows Dev Environment Overview, Architecture Content Customer Research

How do I know if I need to conduct research?

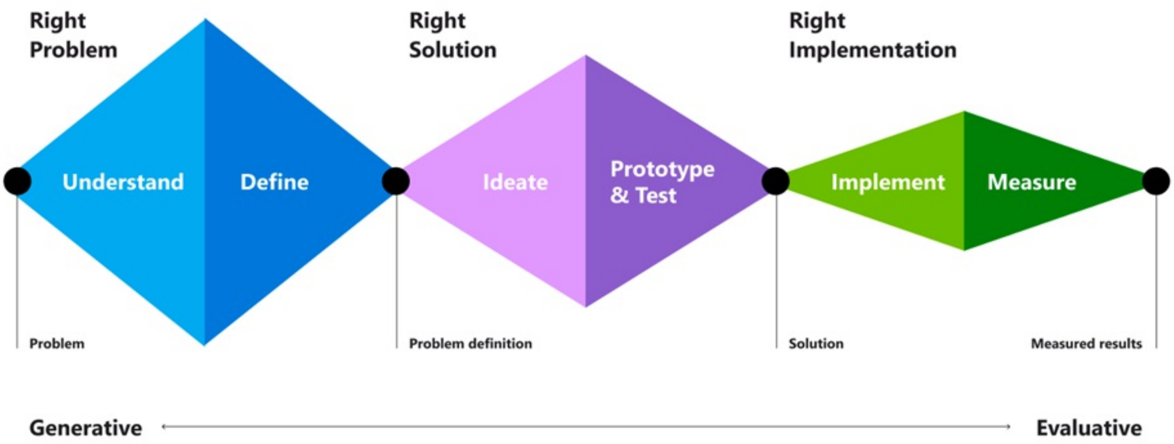
If you're not sure whether or not your feature or content requires research, see the decision tree below.



When should you conduct research?

Typically, there are three times when you might conduct research: in the problem discovery phase, in the solution phase or in the measurement phase.

- In the problem discovery phase, you'll focus on identifying the problems your users are facing.
- Then, in the solution phase, you'll validate whether your solution is usable and addresses the problem. Usually, you'll iterate a few times on it.
- Lastly, once your solution is live, you'll evaluate it and assess how people are using it, whether it meets people's expectations, and how it's performing from a KPI perspective.



Conducting your research

Below is your step-by-step guide to customer research and experimentation.

Before you begin:

1. **Figure out what you're trying to learn.** State the objective of your study. The objective is what you want to get out of the research. Here are a few tactics to further define your study.

- *Hypothesis*: (example) We believe 80% of customers are not happy with X feature because of Y.
- *Research questions* are specific areas that you want to study/gather information. (example) What are customers' impressions of the site?

2. Review research that's already been done.
   - Check out HITS (internal database full of research from other teams) for examples of studies that have

been performed.
- Look through what is posted on the Customer Research SharePoint, which has a list of research performed by the Content and Learning team.

3. **Gather baseline metrics about how things are currently performing.** It's important to know the current state so that you can track any changes that happen. If you don't have baseline metrics, you can't show progress.
   - Content Performance Dashboards:
     - Content and Learning
     - Dynamics
     - Microsoft365
   - Kusto
   - Microsoft Docs Metrics (MDM)

File an Azure Boards (previously called VSTS) work item to start the process

- What problem you are trying to solve? What are you hoping to learn?
- Who is your target audience?
- List a date you'd like to see this completed by.

Please note, the research process requires at least two weeks lead time. We will not be able to help you with a page that is launching tomorrow

Once your Azure Boards item has been triaged, someone from the research team will contact you to set up a meeting.

Below is a brief description of what to expect in each meeting.

1. **Introduction and define problem.** Review objectives, hypothesis, and/or research questions. Go over analysis of existing data to get a baseline if available.
2. **Discuss research methods.** Determine which method to use, how to recruit customers, screening guidelines, and create a research plan.
   - *Next steps: create research artifacts*
3. **Review prepared research materials** (survey outline, wireframe, prototype, etc.) We will review your research materials together.
   - *Next Steps: Conduct research*
4. **Analyze results and share knowledge** Write-up results in HITS format and post to SharePoint. Decide whether or not more research needs to be done. (Implementation experimentation, update contributor guide, share results).
   - *Next Steps: Write up research and post on SharePoint. Host a Share & Learn session.*

Expectations

- **Partnership** Please keep in mind that this is a partnership. We will work with you to provide guidance and assistance to help you complete your own research.
- **Come prepared.** At the end of each meeting, you'll be assigned action items to complete before the next meeting. Come to each meeting prepared so that we can make the best use of our time.
- **Track your work in Azure Boards.** This helps hold people accountable, allows the team to load balance, and provides a visible way to document what research is going on. Additionally, it allows others to learn from the work being done and shares knowledge across the team.
- **Have a growth mindset.** We know this might be out of your comfort zone or you might not have any experience in this area, but that's what we're here for. Reach out if you have any questions. This is a learning

process for all.

- **Share your results.** Good research makes everyone better. Share your findings with the team to show how customer research translates into actionable items.

Next Steps

If you're ready to get started, file an Azure Boards work item! We look forward to hearing from you!

Choose a customer research methodology

6/24/2021 • 4 minutes to read

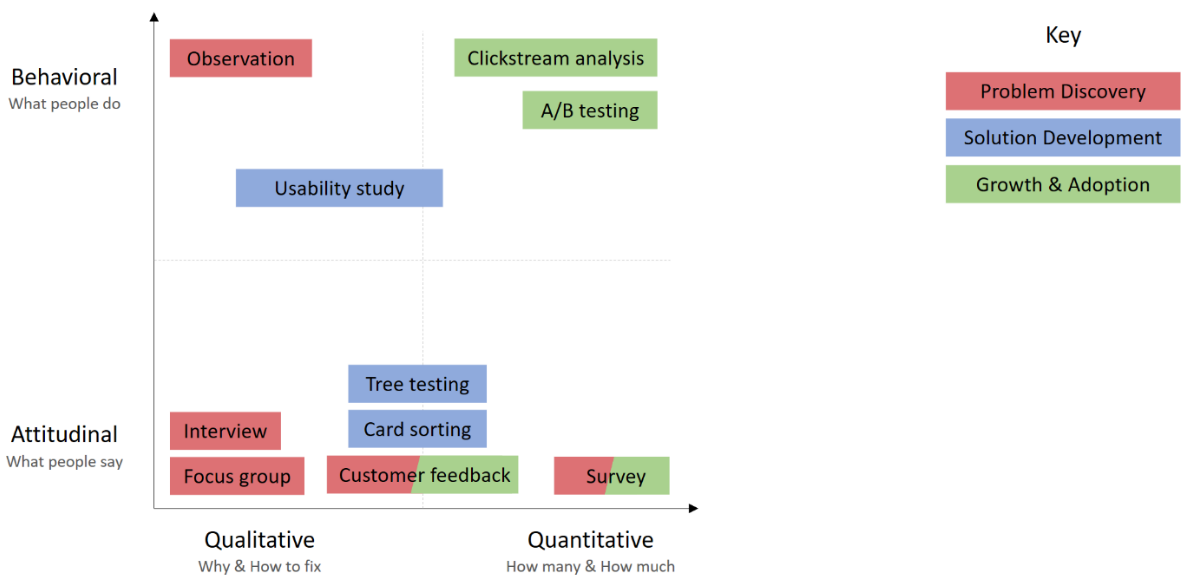
There are many different types of customer research. To decide which method to use, it's important to determine what you're trying to learn and which phase you're in as outlined in the following table:

Research methods and tools by product phase

	Problem discovery	Solution development	Growth & adoption
Questions	What problems do customers have? What's already been done and being used?	Are we building the right thing? Can users find what they're looking for?	Did we build the right thing? How are we doing?
Approach	Qualitative and quantitative	Mainly qualitative	Mainly quantitative
Research Methods (Need customers)	Observation, interview, focus group, survey, customer feedback	Card sorting, tree testing, usability study	A/B test, Click stream, survey, customer feedback
Other tools (We can do on our own)	Competitive analysis, task analysis, stakeholder interview, persona	Literature review, storyboard, journey map, guideline review, heuristic evaluation, cognitive walkthrough	

Another way to think about different types of research is whether they're attitudinal or behavioral as well as qualitative or quantitative. This chart depicts where research methods fall on these two axes.

Research method by data type



Research method by inquiry

If you're not sure where to start, try focusing on what information you're looking to learn.

Who are our customers? What problems do they face?

Interviews

Interviews can be used for lots different scenarios. For example, you might use an interview to ask participants about how they complete certain tasks. Your customers may remember some specific cases where the product worked well or poorly and can often provide more vivid details about these incidents. You can use this tactic to get an idea of the strengths and weaknesses of your content in helping users accomplish their tasks. You can also use interviews to help you identify questions to ask in a broader survey. It's important to note that in customer conversations we're focusing on participants' past experiences - not asking them to speculate about the future. Customer interviews may seem daunting at first, but it's helpful to keep in mind that it's simply a conversation between two human beings.

Surveys and questionnaires

Surveys provide you with answers similar to what you'd get from user interviews. The downside is that you can't dive deeper into those answers as there's no direct interaction with the customer. On the plus side, they allow you to get a larger volume of responses, which can open up the opportunity for more quantitative analysis. It's important to ask carefully worded questions that are free from bias.

Can people easily find information on your website?

Card sorting

This technique helps generate or validate a structure, like navigation, a TOC, or content organized into categories. To do a card sort, make a list of the concepts you need to organize, write them on cards, and ask users to organize them into categories. To generate new structures, do an open sort, where participants put the cards into groups and then name the groups themselves. To validate a structure you've already developed, do a closed sort, where you provide categories to sort the cards into. The results of a card sorting study can help you to decide the structure of your article, how to label your table of contents and/or how to group your content.

Tree testing

Tree testing helps assess the findability of content; it can be used for any kind of navigational structure but is particularly well-suited to tables of contents. In a tree test, you create a tree structure (without any UI) and define items for participants to find. Participants then try to find the items you've identified by navigating through the tree and noting when they think they've found it. Because they separate the actual navigation structure from the UI, tree tests are useful to see how understandable the labels you've chosen are and whether the structure you've defined lets people find things they're looking for. At the end of the study, you'll have metrics including task success rate, time to complete the task, and an analysis of routes users took through the site tree before selecting an answer (correctly or incorrectly).

What do people think?

Focus groups

A focus group is when you bring together a group of participants to ask them about a certain topic. For example, you might host a focus group to ask participants how they like to learn or what their initial reaction to a product is. Having a group of users together can also help them to jog memories and ideas in each other that they may not have otherwise remembered. Beware of group think as it can influence the way participants respond in these kinds of settings. Generally, our team hires vendors to complete these types of studies.

What do customers do?

A/B testing

For information about A/B testing please see the experimentation wiki

Observations

Observations involve studying people as they go about their everyday lives or tasks. This helps understand how people complete a task or use a work-around for a known problem. For example, you might use an observation study to see how developers find and use technical documentation. This is usually part of an interview and is brief. Observations are exploratory whereas usability studies generally have a goal or task to be completed.

Search engine optimization (SEO)

SEO is important because 60-70% of our traffic comes from search. By using tools such as google analytics, we can understand what customers are searching for and tag our pages with the most effective key words. More information on SEO can be found in the SEO and analytics presentation video.

Content reporting

Content reports can be used to gather performance on a page. These reports include engagement metrics such as bounce rate, CSAT, views, and traffic data. For more information, see Report and interpret content engagement metrics.

Usability testing

In usability testing, you come up with a list of tasks and observe how correctly or incorrectly users accomplish those tasks. For example, you might perform a usability test to understand how many users successfully register for a MS Learn account. The data you gather might include how long it takes users to complete the task, how many errors they make, where they deviate from our ideal and how many users complete the task.

UserTesting

6/28/2021 • 2 minutes to read

What tools do we have available to conduct studies?

Our main user research tool is UserTesting.com. UserTesting is a platform for getting rapid customer feedback on almost any customer experience you can imagine, including websites, mobile apps, prototypes, and real world experiences. You'll receive audio and video recordings of real people speaking their thoughts as they complete tasks you specify. With UserTesting, you can run moderated and unmoderated tests to get connected with your customers on a global scale. You can use this resource to deep dive in a problem space and get feedback on concepts and prototypes. To learn more about UserTesting, see their FAQs

How do I get a UserTesting license?

To take advantage of this resource, you'll need to submit a request below. You'll receive instructions to set up your UT seat and then be asked to complete some prerequisite e-learning courses before receiving your license.

What is the process?

1. Complete e-learning courses
2. Submit a request for a license
3. Receive an account from the research team
4. Build study plan with the help of the research team (there are templates available here)
5. Receive approval to launch study

Note, we require at least three business days' notice to launch a UserTesting study

What training is required?

In order to get access to UserTesting.com, you must complete the three online eLearnings listed below:

- Getting Comfortable with Customer Conversations - In this training you'll learn best practices for having customer conversations without bias. This includes preparing for conversations, moderating conversations, and documenting insights.
- Problem-Focused Thinking- In this training you'll learn the difference between problem-focused and solution-focused thinking and how to be more problem-focused in your work.
- Ethics and Privacy in Customer Interactions - In this training you'll learn how to act ethically and compliantly when interacting with customers.

After you've completed these trainings, submit a request by filing this work item

FAQs

What type of UserTesting account will I receive?

You will receive a license that gives you two months of access. During that time, you will be able to run up to three studies (moderated and/or unmoderated) with a maximum sample size of 6 per study. We currently have 10 seats available for the entire DevRel team. If after two months, you would like to run another study, feel free to reach back out. Note, you will not have to take the required training again.

Can I get quantitative as well as qualitative feedback?

Yes, you can collect both quantitative and qualitative feedback with this UserTesting trial license.

How quickly will I receive results?

Moderated studies on UserTesting depend on your calendar availability. Typically sessions are scheduled within 24 hours of launching the test. Unmoderated studies are faster and can be picked up by qualifying participants shortly after launching. Timing varies depending on the complexity of the customer profile you're looking for, however most studies are completed within 2 hours.

When should I conduct research?

Reference this guidance in the Contributors' Guide.

What resources are available?

Once you have a UserTesting account, there are many helpful trainings and resources available. Below are a few to help get you started:

Meet Insight Core

Course 1: The Basics

Course 4: Setting up your Test in Insight Core

Use Cases in the UserTesting Platform

There is also a UserTesting + Microsoft Resource Guide which features:

- Upcoming webinars and events
- Recordings of past webinars
- Helpful resources and links

Customer interviews

7/7/2021 • 3 minutes to read

What is it?

Interviews are a great way to get customer data. At first, a customer interview may seem scary, but, in reality, it's just a conversation between two people.

When do I use it?

We use interviews for lots of different things. You can use interviews to ask participants about how they finish certain tasks. They may remember times when something worked well or poorly and might provide more details about these events. Use interviews to get an idea of the strengths and weaknesses of your content and understand how useful it is at helping customers accomplish tasks. Also, use interviews to help you gather questions to ask in a broader questionnaire or survey. It's important to note that we're focusing on participants' past experiences. We're not asking them to speculate about the future.

How do I do it?

One of the most important parts of the interview is the preparation. To prepare for a customer interview, you need a hypothesis, a screener, and a discussion guide.

- A *hypothesis* is an informed guess that changes over time. It's created based on:
  - Existing customer data and your assumptions about the customer.
  - The problems the customer faces.
  - The tasks the customer needs to do, also known as jobs-to-be-done.

  The hypothesis is the basis of your study. The information you learn through your research will help you to refine your hypothesis.

  To formulate your hypothesis, try one of these examples:

  - We believe that [customers] are most frustrated about [job-to-be-done] because [problem]. We believe addressing this [problem] will increase [metric].
  - We believe this [solution] will increase/decrease [End-User Metric].
- A *screener* is a set of specific questions that come before the interview and check if a person qualifies for your study. Tips for making a screener:
  - Keep the screener short. Target between 5 and 10 questions.
  - Avoid questions that only require "yes" or "no" answers.
  - Don't make the "correct" answer obvious. People will try to answer in a way that will qualify them for your study.
  - Always provide a neutral answer like "none of the above" or "I don't know."
  - Aim to be diverse (varied genders, ages, ethnicity, location) and inclusive (varied computer confidence and people with different abilities)
- A *discussion guide* is a script or outline for the conversation. It's important to make sure that you're asking the right questions to get the data you need. Customer interviews are most valuable when they provide actionable feedback and confirm or disprove our assumptions. Here are some tips to writing a discussion guide:

- Always have more questions than you think you need.
- Ask open-ended questions. Open-ended questions leave space for the customer to share their experience and help you remove bias from the discussion. For example: Tell me about the last time you used technical documentation. What made this experience good or bad?
- Add quantitative questions to your interview. Ask customers to rank things on a scale from one to five. Base the ranking on how useful something is or how frustrated they are.
- Quick pulse calls are a great way to ensure that you're heading down the right path before you invest further into your research.

How many customers do I need?

For small changes, you need between five and seven interviews. For a large change or an entirely new approach, you want to hear from as many customers as possible. The best indicator that you've done enough research is that you stop hearing new information from customers. At that point, you've reached what's called theoretical saturation.

What tools are available?

These tools available to connect directly with customers:

- Moneyball - The Moneyball concierge team finds and schedules participants using their own database.
- UserTesting - UserTesting is an online platform for moderated and unmoderated interviews. They have their own panel of over 2 million global users.
- Azure Advisors - Azure Advisors is a community of customers who have opted in to provide feedback on Microsoft products.

Customer interview basics

- Asking The Right Questions On User Research Interviews - provides advice on how to conduct a customer interview.
- Formulate a Discussion Guide - a step-by-step guide to create a discussion guide.
- Screener Library - examples of pre-built screeners.
- Screener Guide - how to make a screener.
- Problem-focused Interview Workbook - a step-by-step guide to writing a hypothesis, screener, and discussion guide.
- Interviewing Developers - an example discussion guide by Mimi Gentz
- Reducing Bias - e-learning course from the Signal team.
- Moneyball Online Training - e-learning courses.

Become an interview expert

- Interview Tips
- Avoid Leading Questions to Get Better Insights from Participants
- Understanding Cognitive Biases

Surveys/Questionnaires

7/7/2021 • 2 minutes to read

What is it?

A survey is a list of questions intended to gather information about a group of people. Surveys are quick to administer. They allow you to gather large a sample size, but can be tricky to do well. It's important to have well-crafted thoughtful questions to make sure that you're getting the correct information.

When do I use it?

Surveys are a good way to follow up on qualitative data, like interviews, to go deeper, and get more quantitative data to substantiate your hypothesis. They can also be a good way to surface areas of concern from a large group of people. Then you can go deeper in with follow-up customer conversations.

How do I do it?

Making a concise and effective survey can be tough. Start by listing what you want to learn from your survey. Begin by writing down the answers you're hoping to get. Then turn the answers into the questions that you'll ask. It will help you gain actionable feedback and avoid asking unnecessary questions. Here are a few tips for writing effective survey questions:

- Make sure to use simple language and break down large ideas into multiple questions.
- Avoid leading questions and offer balanced answers. Make sure to provide an equal amount of positive and negative choices.
- Always include a "none of these" or "I don't know" option.
- Avoid survey fatigue by asking 5-10 questions. If possible, show a progress bar so participants know how far along they are.
- Make sure to pre-test your survey before putting it out in the wild. Send your survey to a few colleagues for feedback and to see if any questions were unclear

How many customers do I need?

Heres one article from SurveyMonkey and one article from Qualtrics explaining population, margin of error, and confidence level.

What tools are available?

Microsoft Forms

Writing good questions

Try to write clearly. Unbiased questions are best. If you don't know how to write good survey questions, check out these articles:

- Surveys 101: A Simple Guide to Asking Effective Questions
- Survey Questions 101: Do You Make These 7 Mistakes?

- Guidance for Crafting Survey Questions
- Example Survey that Mimi Gentz Created for Cosmos DB Research
- Survey Question Best Practices and Recommended Questions

Become a survey expert

- LinkedIn Learning – How to Create a Survey
- LinkedIn Learning – What Makes a Good Survey
- LinkedIn Learning – How to Analyze Survey Results

Customer research and experimentation FAQs

6/28/2021 • 2 minutes to read

Which process answers which types of questions?

Customer research provides qualitative data and answers questions like:

- "why?"
- "what do users say?"
- "what do users think?"

Experimentation provides quantitative data and answers questions like:

- "how many?"
- "what do users do?"

What do you do first? Does one come before the other?

In general, we recommend doing research first. Once you have an idea of the direction you'd like to go, you can use experimentation to validate. In other words, if you don't know where to start, do some research. If you have a well-formed hypothesis, do an experiment.

A few things to note: with A/B testing, anyone can land on your page. With research, you can test with a specific audience. Lastly, if there is significant risk, we should always test with experimentation.

Where can I find information about everything that is offered?

If you want to learn more about the research program, see how to perform customer research. To read more about the experimentation program,see the experimentation wiki

Azure Pulse customer feedback program overview

5/10/2021 • 2 minutes to read

The Azure CXP runs a program called Azure Pulse that collects customer feedback through Technical Account Managers and Cloud Solution Architects. The customers in this program are selected for strategic importance, risk of churn, and edge case scenarios. The feedback items are periodically routed to a Content & Learning PM for triage to the appropriate content owner in the Content & Learning organization for a response.

Handling Azure Pulse feedback items quickly is high priority and required activity for content teams. Progress and completion are tracked by CXP and shared monthly in Azure Fundamentals.

Process overview

1. A Microsoft field employee such as a Technical Account Manager or a Cloud Solution Architect creates an Azure Pulse work-item based on customer feedback.
2. Azure CXP classifies the feedback items by product and documentation feedback is labeled as "Content Services".
3. A week before Fundamentals, the Pulse team sends a status/reminder mail to encourage teams to make progress before to Fundamentals.
4. The Azure Pulse Lead for Content & Learning, Mangesh Sangapu (msangapu), is responsible for triaging and assigning new items to appropriate content developers and/or team managers for the affected Azure service. The Pulse Lead sends an e-mail notification to the affected developers and/or team managers indicating a new Pulse work-item has been assigned to them.
5. The manager or delegate must review the feedback, investigate as needed, and prepare a "customer ready message" (CRM) as a response the field employee can provide to the customer. The response should address feedback provided and give next steps or the plan of action for addressing the feedback.

Query

To query for new or existing items, visit: https://vstfrd/Azure/RD/_workitems?id=e404e8af-db6c-4ea3-bcc0-91fa062fc813&_a=query

Program goals

1. White Glove / Customer Rescue
2. Drive improvements into Azure services and experiences
3. Build engineering empathy
4. Committed customer pool for design reviews and private previews

Service owner guide

See the Pulse Service Owner Guide for complete details. SLAs are documented on the last page.

Azure CPE PSAT Program overview

6/24/2021 • 2 minutes to read

The Azure PSAT survey program tracks & provides Azure all-up customer experience measurement & insights with quantitative data and compete benchmarking. The Azure PSAT team partners with the C+AI engineering teams and the CMR (Customer & Market Research) team on survey design, survey result analysis, action plans, and improvement tracking.

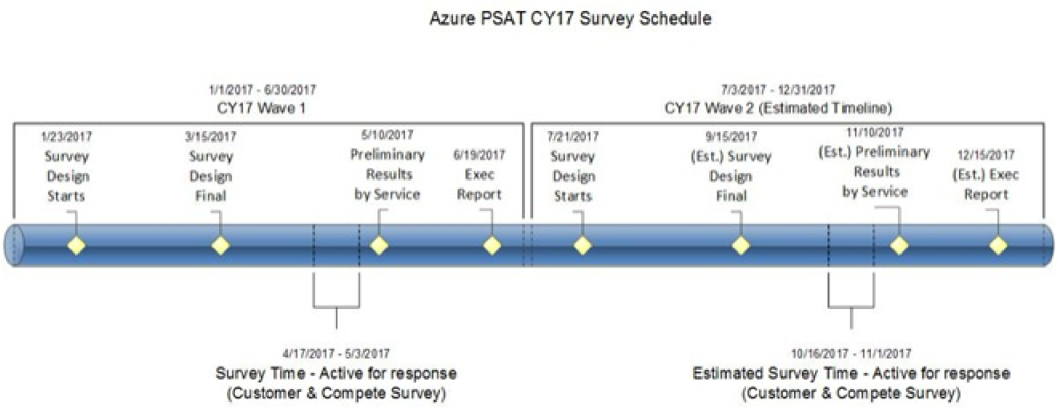
Survey Coverage

The Azure PSAT survey was reestablished by Azure CXP team in Nov 2015 and has been running twice a year since then. The survey program now runs the Azure customer survey and Azure compete survey in parallel and integrates with Azure customer telemetry data.

- Azure Customer Survey: Microsoft-identified email survey to Azure customers. World-Wide English Survey covering all existing Microsoft Azure customers with billing revenue >$0, and opt-in for marketing promotion communications (that is, customer feedback surveys).
- Compete Survey: Microsoft not revealed as survey sponsor. The US English Email Survey to panel samples is provided by a third-party survey vendor focusing on US AWS and US GCP.

Program Goals

1. Understand customer satisfaction and related drivers
2. Drive improvements into Azure services and experiences
3. Compare Azure customer feedback with the feedback of top competitors



Timeline

Resources

- For more info, go to the Azure PSAT Portal
- Executive Reports: Azure PSAT Report Archive
- For Newsletter Join Azure PSAT Distribution Group
- Azure CXP Program Lead: Amira Youssef

GitHub issue and public pull request reports

6/24/2021 • 8 minutes to read

There are several types of reports you can run for GitHub issues:

- You can sign up for scheduled reports that can be emailed to your inbox, as described later in this article Azure GitHub reports.
- Download and run GitReporter tool.
- Run reports at the GitHub Issues dashboard https://aka.ms/githubissues.

Getting started with using Azure GitHub reports

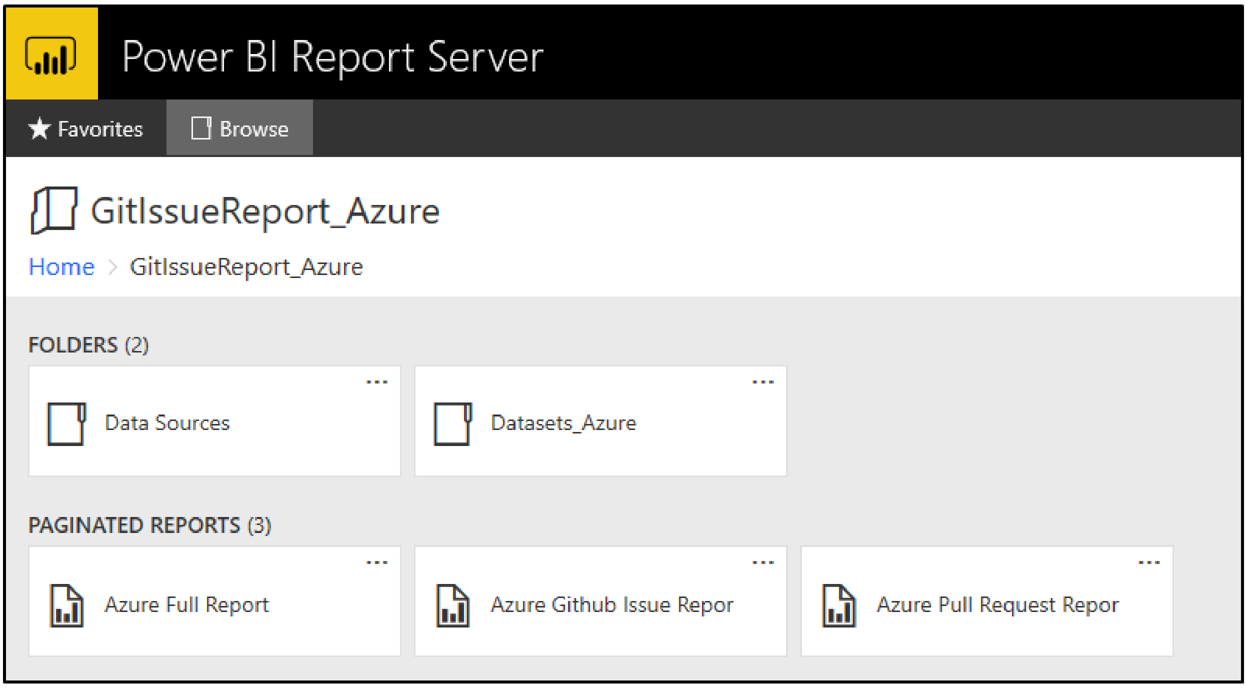
GitHub reports combine PRs and GitHub issues into a report for the Azure repo. You can receive these reports automatically and filter on them for details such as service, author, and manager. Here's how to configure and use these reports.

> **NOTE**
> Contact Van (vanto@microsoft.com) with any feedback or issues.

There are three reports created:

- **Azure Full Report** (PRs + issues) - `http://roguereporting.westus2.cloudapp.azure.com/reports/report/GitIssueReport_Azure/Azure%20Full%20Report`
- **Azure GitHub Issue Report** - `http://roguereporting.westus2.cloudapp.azure.com/reports/report/GitIssueReport_Azure/Azure%20Github%20Issue%20Report`
- **Azure Pull Request Report** - `http://roguereporting.westus2.cloudapp.azure.com/reports/report/GitIssueReport_Azure/Azure%20Pull%20Request%20Report`

> **NOTE**
> Please copy and paste the above links into your browser. The SSRS Service is not using HTTPS.



Location of these reports: `http://roguereporting.westus2.cloudapp.azure.com/reports/browse/GitIssueReport_Azure`

IMPORTANT

If you are unable to access the Power BI Report Server, and get an unauthorized message, please contact Van (vanto@microsoft.com) for access.

The color codes in the age column correspond with the SLA for pull requests:

- 0 - 7 days: **Green**
- 8 - 14 days: **Yellow**
- 15 days or more: **Red**

NOTE

The data source behind these reports is refreshed on an hourly basis between **6:30AM** and **6:30PM** (Pacific Time) daily. This means that if you update a PR or issue outside of these hours, you will not see the data refreshed in the reports until the next refresh window.

There are two ways to consume these reports:

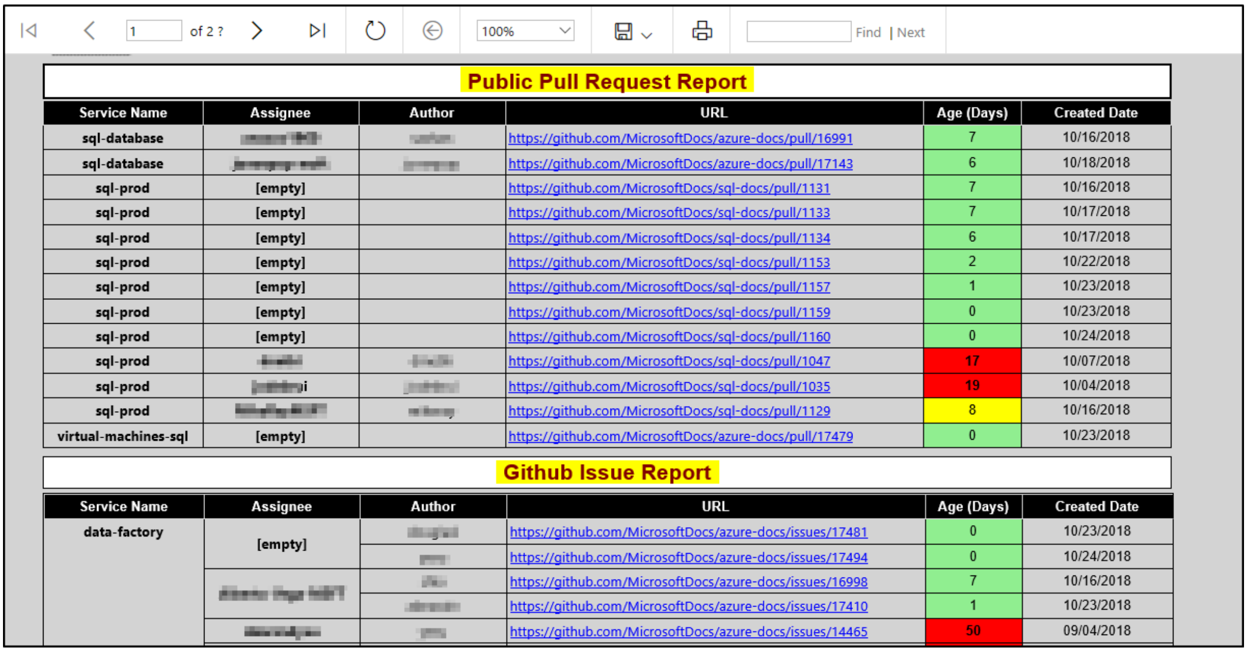
- Interactive reporting - view and configure the report in the web GUI of the Report Server.
- Subscribe to reports - configure the report to be sent via email.

Interactive reports

Interactive reporting allows the user to filter the results based on preference of services and assignees. The user can also export the results to their desired format, such as Excel or PDF.

The **Azure Full Report** incorporates both PRs and GitHub Issues into one report. **Azure GitHub Issue Report** only has Issues, and **Azure Pull Request Report** only has PRs.

Azure Full Report



Public Pull Request Report

Service Name	Assignee	Author	URL	Age (Days)	Created Date
sql-database	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/16991	7	10/16/2018
sql-database	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/17143	6	10/18/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1131	7	10/16/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1133	7	10/17/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1134	6	10/17/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1153	2	10/22/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1157	1	10/23/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1159	0	10/23/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1160	0	10/24/2018
sql-prod	[illegible]	[illegible]	https://github.com/MicrosoftDocs/sql-docs/pull/1047	17	10/07/2018
sql-prod	[illegible]	[illegible]	https://github.com/MicrosoftDocs/sql-docs/pull/1035	19	10/04/2018
sql-prod	[illegible]	[illegible]	https://github.com/MicrosoftDocs/sql-docs/pull/1129	8	10/16/2018
virtual-machines-sql	[empty]		https://github.com/MicrosoftDocs/azure-docs/pull/17479	0	10/23/2018

Github Issue Report

Service Name	Assignee	Author	URL	Age (Days)	Created Date
data-factory	[empty]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/17481	0	10/23/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/17494	0	10/24/2018
	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/16998	7	10/16/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/17410	1	10/23/2018
	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/14465	50	09/04/2018

There are five parameters available for filtering in the **Azure Full Report**.

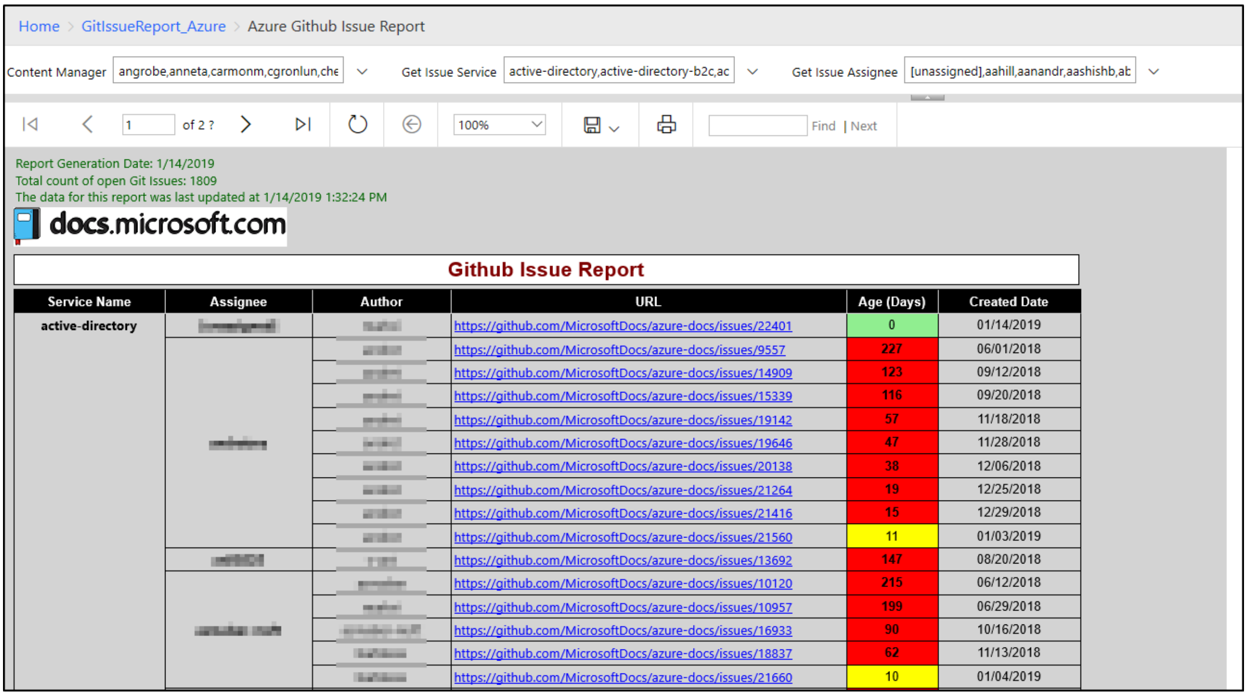
- **Content Manager**: Filter based on the Content Manager alias, which will limit the result set to services owned by the Content Manager(s)
- **Git Issue Service**: GitHub Issue Report filter based only on the selected service(s)
- **Git Issue Assignee**: GitHub Issue Report filter based only on the selected assignee(s)
- **Git PR Service**: Pull Request filter based only on the selected service(s)
- **Git PR Assignee**: Pull Request filter based only on the selected assignee(s)

The top-level parameter is Content Manager. Based on this selection, the services that are available will be narrowed down to the services owned by the selected Content Manager(s). The next level parameter is Service. Based on this selection, the Assignee will be narrowed down to the Assignee(s) that has Issues or PRs opened in that service.

> **NOTE**
>
> If you select a Content Manager and/or a Service that doesn't have GitHub Issues currently open, there will be a null value in the parameter selection for Assignee. The full report must have at least 1 selection in each parameter, or the report will fail to render.

Once the user is done with their parameter selection, they can select the **View Report** button located on the upper right-hand corner to render the report.

Azure GitHub Issue Report



Home > GitIssueReport_Azure > Azure Github Issue Report

Content Manager: angrobe,anneta,carmonm,cgronlun,che | Get Issue Service: active-directory,active-directory-b2c,ac | Get Issue Assignee: [unassigned],aahill,aanandr,aashishb,ab

1 of 2 ? | 100% | Find | Next

Report Generation Date: 1/14/2019
Total count of open Git Issues: 1809
The data for this report was last updated at 1/14/2019 1:32:24 PM

docs.microsoft.com

Github Issue Report

Service Name	Assignee	Author	URL	Age (Days)	Created Date
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/22401	0	01/14/2019
	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/9557	227	06/01/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/14909	123	09/12/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/15339	116	09/20/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/19142	57	11/18/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/19646	47	11/28/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/20138	38	12/06/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/21264	19	12/25/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/21416	15	12/29/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/21560	11	01/03/2019
	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/13692	147	08/20/2018
	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/10120	215	06/12/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/10957	199	06/29/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/16933	90	10/16/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/18837	62	11/13/2018
		[illegible]	https://github.com/MicrosoftDocs/azure-docs/issues/21660	10	01/04/2019

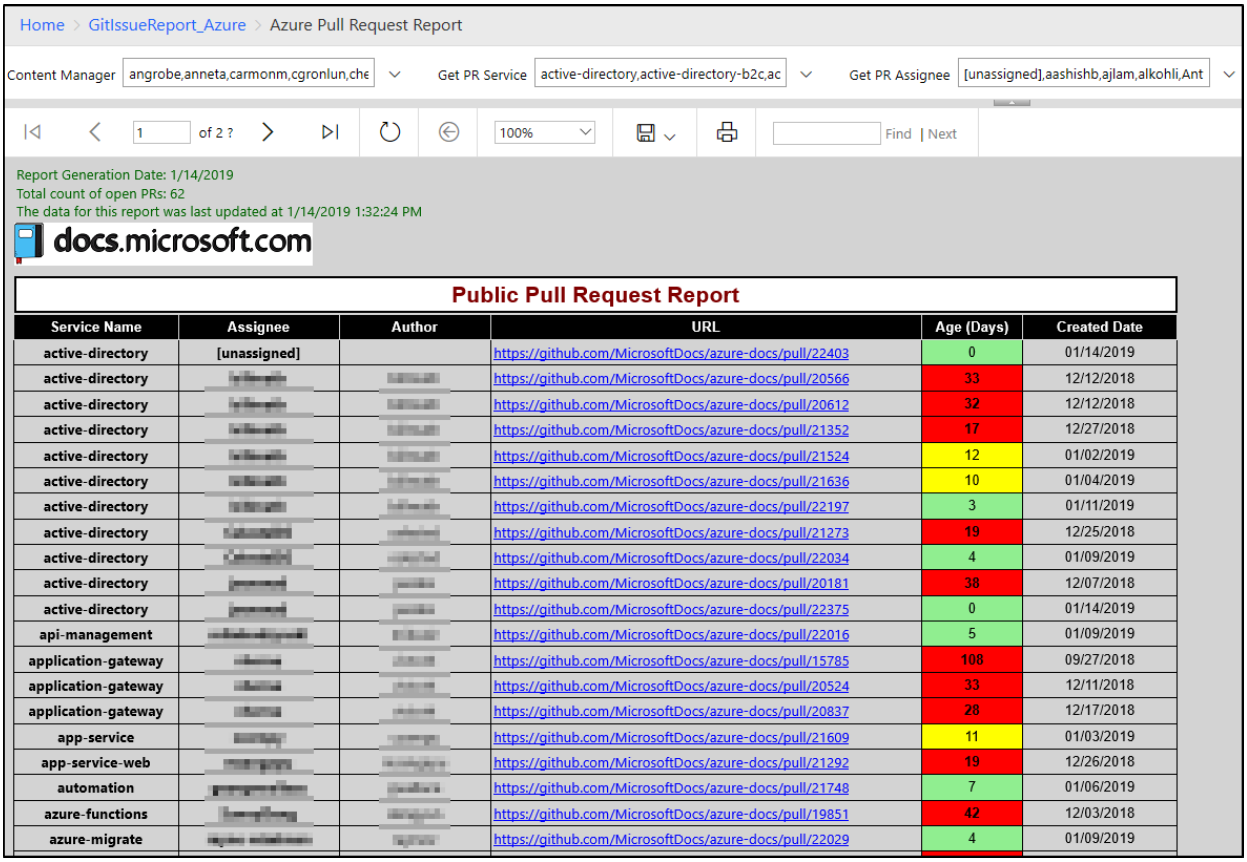
There are three parameters available for filtering in the **Azure GitHub Issue Report**.

- Content Manager, Git Issue Service, and Git Issue Assignee

The following table describes the columns and definitions of the GitHub Issue Report.

COLUMN	DEFINITION
Service Name	Refers to the Service listed in the GitHub Issue. The value is obtained from the labels assigned from GitHub.
Assignee	Refers to the current designated user that is under Assignees in the GitHub Issue. This username is the GitHub username.
Author	Refers to the author of the article from the GitHub Issue when it was opened. The value is obtained from the **Document Details** of the GitHub Issue, and is the Microsoft alias of the author. This value is also populated from the ms.author metadata for the article when the Issue was first opened.
URL	Refers to the URL of the GitHub Issue.
Age (Days)	Refers to the number of days that has passed since the GitHub Issue was opened and when the report execution.
Created Date	Refers to the date that the GitHub Issue was created.

Azure GitHub Pull Request Report


Home > GitIssueReport_Azure > Azure Pull Request Report

Content Manager: angrobe,anneta,carmonm,cgronlun,che | Get PR Service: active-directory,active-directory-b2c,ac | Get PR Assignee: [unassigned],aashishb,ajlam,alkohli,Ant

1 of 2 ? | 100% | Find | Next

Report Generation Date: 1/14/2019
Total count of open PRs: 62
The data for this report was last updated at 1/14/2019 1:32:24 PM

docs.microsoft.com

Public Pull Request Report

Service Name	Assignee	Author	URL	Age (Days)	Created Date
active-directory	[unassigned]		https://github.com/MicrosoftDocs/azure-docs/pull/22403	0	01/14/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20566	33	12/12/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20612	32	12/12/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21352	17	12/27/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21524	12	01/02/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21636	10	01/04/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22197	3	01/11/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21273	19	12/25/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22034	4	01/09/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20181	38	12/07/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22375	0	01/14/2019
api-management	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22016	5	01/09/2019
application-gateway	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/15785	108	09/27/2018
application-gateway	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20524	33	12/11/2018
application-gateway	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20837	28	12/17/2018
app-service	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21609	11	01/03/2019
app-service-web	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21292	19	12/26/2018
automation	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21748	7	01/06/2019
azure-functions	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/19851	42	12/03/2018
azure-migrate	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22029	4	01/09/2019

There are three parameters available for filtering in the **Azure Pull Request Report**.

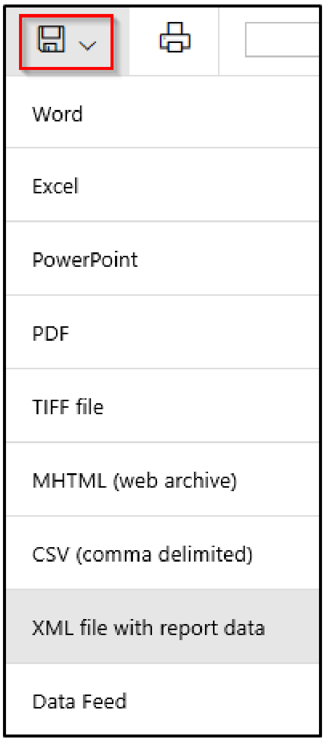
- Content Manager, Git PR Service, and Git PR Assignee.

The following table describes the columns and definitions of the GitHub Pull Request Report.

COLUMN	DEFINITION
Service Name	Refers to the Service listed in the GitHub Public Pull Request. The value is obtained from the labels assigned from GitHub.
Assignee	Refers to the current designated user that is under Assignees in the GitHub Issue. This username is the GitHub username.
Author	Refers to the Microsoft alias that corresponds to the user in the Assignee column.
URL	Refers to the URL of the GitHub PR.
Age (Days)	Refers to the number of days that has passed since the GitHub PR was opened and when the report was executed.
Created Date	Refers to the date that the GitHub PR was created.

Export the results from the report

You can export the results from the report after choosing your desired parameters and rendering the report by selecting the **save** icon in the report controls. Choose your desired export format:



To print the report results, select the **print** icon next to the **save** icon.

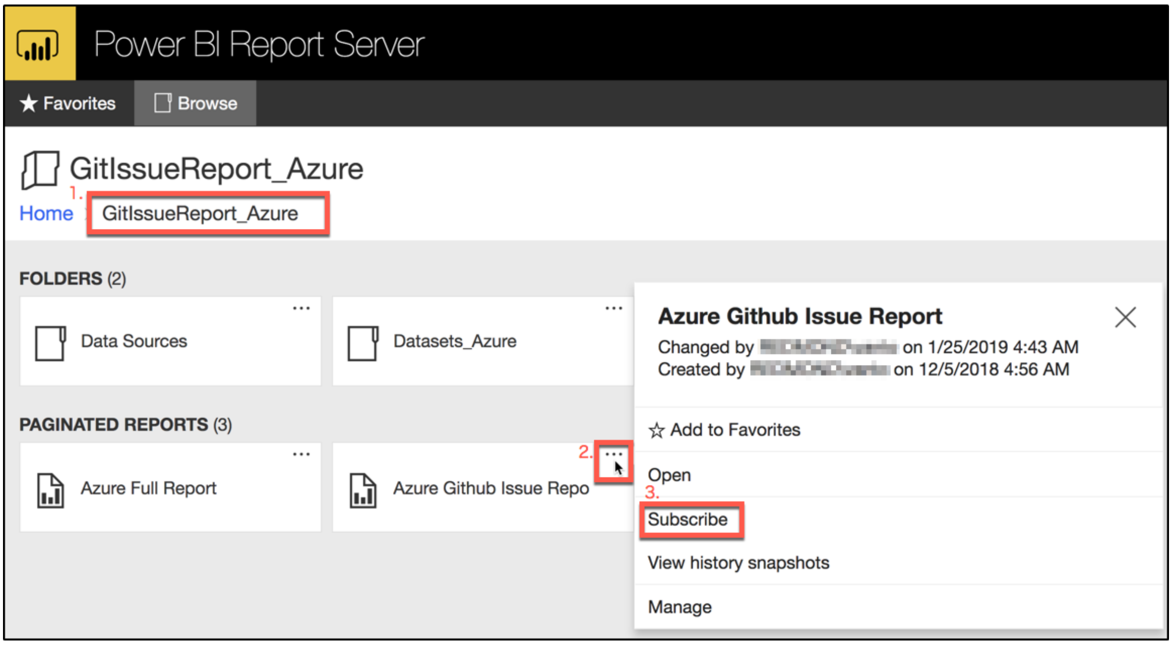
Subscribe to reports

Subscriptions are the ideal way to consume these reports, as they allow for reports to be automatically generated and sent via email, in any format of your choosing. The simplest format is MHTML, which is readable in email. You can set up subscriptions for yourself, your team(s), or your engineering partners.

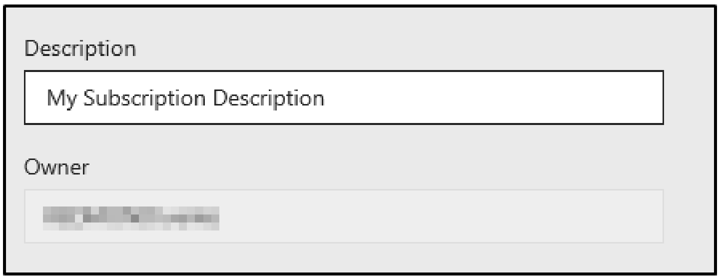
Setting up subscriptions

1. Go to the **GitIssueReport_Azure** folder in the Power BI Report Server, and select the **"..."** button next to the report that you want to subscribe to. Select **Subscribe**.

   ```
   http://roguereporting.westus2.cloudapp.azure.com/reports/browse/GitIssueReport_Azure
   ```

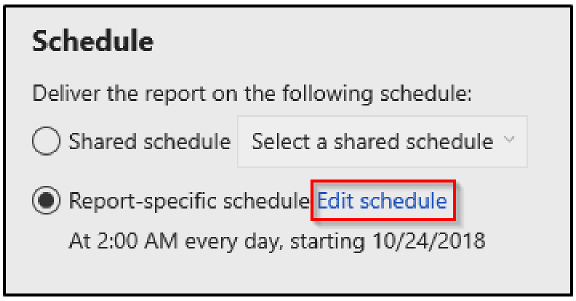


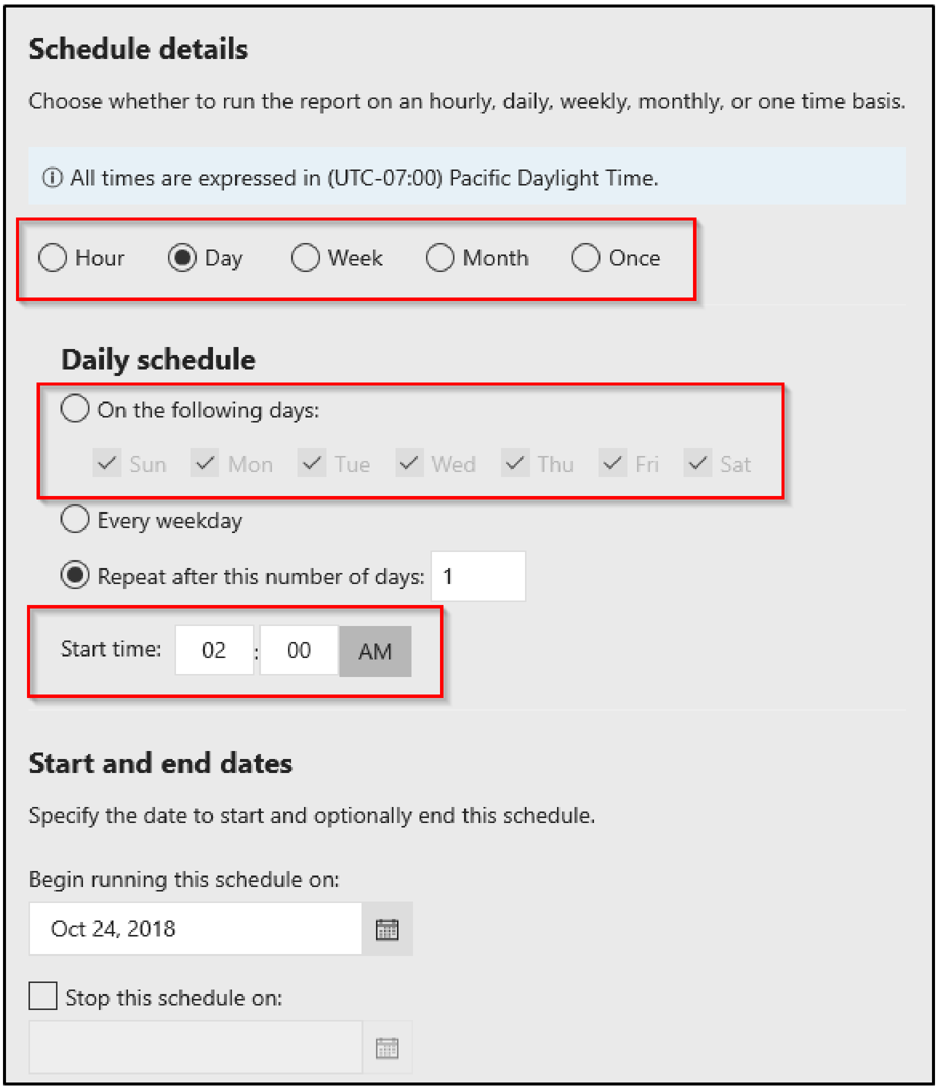
2. Enter a description for this subscription.



3. Type of subscription: **Standard subscription** (default)

4. In the **Schedule** portion, select **Edit schedule** and configure the **Schedule details**.



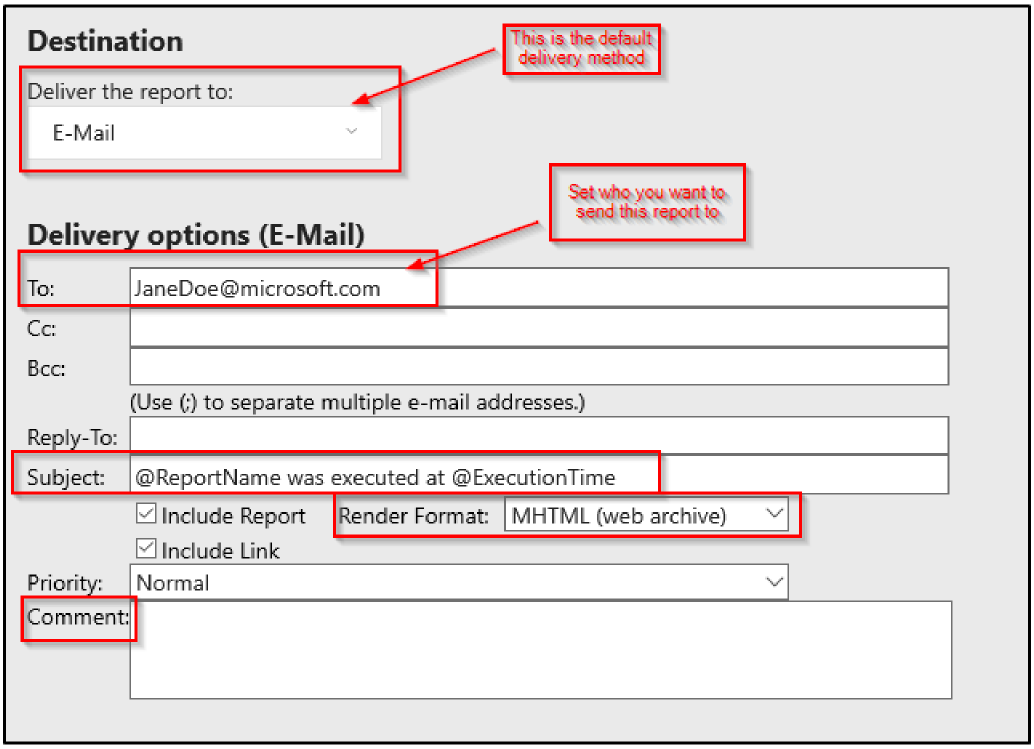


5. Once you're done editing the schedule, select **Apply** at the bottom of the page.

6. The next section of configuring your subscription is to set the delivery method to **E-mail**.

7. Add an email address you want to send this report to, and then *optionally* configure the **Delivery options** for email: **Subject** and **Render Format**. If you want to send the report to multiple email

addresses, separate the email addresses using a semicolon (;).

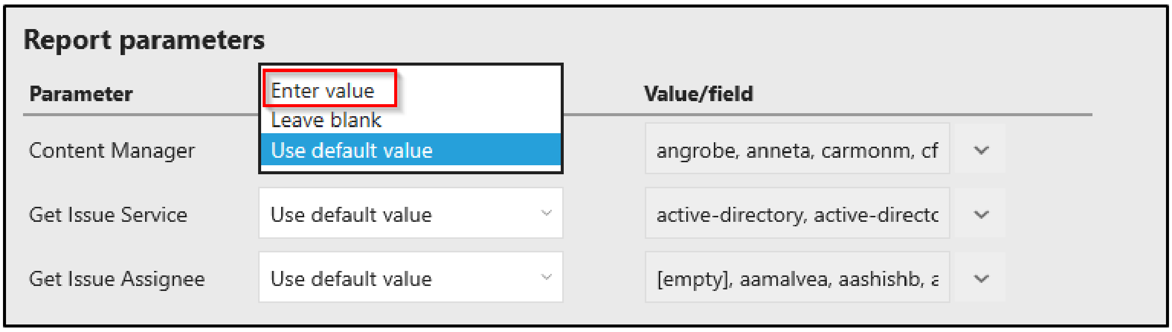


> **IMPORTANT**
>
> Use fully qualified domain addresses for the email address. For example, JaneDoe**@microsoft.com**. You will get an error if you only add an alias.

> **NOTE**
>
> Normal users will not see a section for wider distribution such as adding emails to the **Cc** or **Bcc** line. Add additional emails on the **To** line. If there is a need for greater permissions, contact Van (vanto@microsoft.com).

8. Set the **Report parameters**. Each parameter can be set by selecting the drop-down box next to the Parameter name and choosing **Enter value**. You will then be able to provide a selection of available values to set.

   - Choosing a Content Manager will filter the other Parameters, Service, and Assignee. Choosing a Service will also narrow the Assignee Parameter to assignees that have an open Issue or PR under that Service.

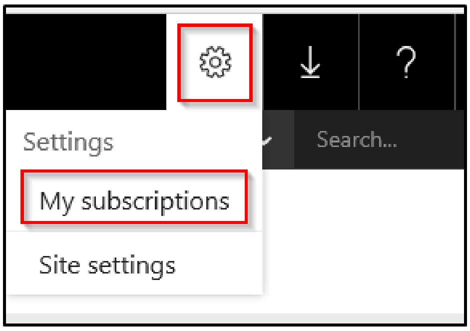


> **WARNING**
>
> If any of the report parameters have null values, the whole subscription will fail to run. An example of this is choosing to subscribe to the full report, and specifying a Content Manager or Service that does not have any open GitHub Issues for the execution of the report. Consider creating a subscription for the GitHub issue report separately. If the subscription does fail, it will mean that there were no GitHub Issues for that particular service when the subscription ran. This only applies to GitHub Issues, as PR reports will still run with null values.

9. Select the **Apply** button at the bottom of the new subscription to create your subscription.

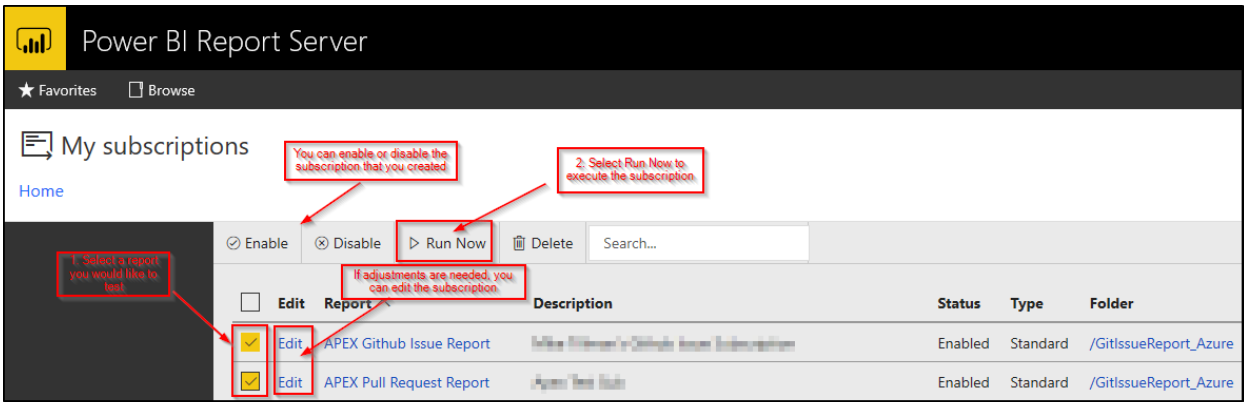
Managed and test your subscription

1. To manage your subscription, open the **Settings** (gear wheel) icon on the upper right-hand corner of the Report Portal, and select **My subscriptions**.



2. On the **My subscriptions** page, you can:

   - Select your report, and run it immediately.
   - Edit your subscriptions.
   - Enable, disable, or delete your subscription.
   - Check the status of the subscription, the last run time, and the results.



Once you are done with the settings and testing, that should be it! You should now get automated emails to the report with little maintenance.

Known issues

- If you try to render any of the reports that do not have GitHub Issues associated with the report, there will be no values picked up in the Parameter's data source. The report will fail to render, as the Multi-value Parameter must not be null. For example, if you try to render the Azure Full Report for a particular Content Manager or Service, and there are no Assignees Parameter available (because there are no open Issues under that Service), the report will not render correctly.
  - You can get around this issue by running only the report for GitHub Issue, or GitHub Pull Request.
- If you are running into an issue with your browser caching your session, and returning blank parameters,

you can get around this issue by checking all parameters and making sure that they reset by selecting all values.

How to host a hack-a-doc

7/20/2021 • 17 minutes to read

This article describes two different approaches to organizing a hack-a-doc. We coin the term hack-a-doc as a collaborative, focused event to create, review, and edit Docs to achieve immediate impact on a stated business goal.

The first case study is a hack-a-doc designed to reduce support pain by improving documentation for the Power BI service. The second case study is a hack-a-doc designed to improve outdated documentation (freshness) for the Azure HDInsight service.

There is an attached PowerPoint that describes best practices for planning and hosting an event.

> **NOTE**
>
> Include the Global Experiences team in your Hack-a-doc planning so they can properly forecast their localization efforts. Create a Localization Request ticket for the localization team's awareness.

Case Study 1 - Power BI support hack-a-doc

The Power BI content team hosted a five-hour hack-a-doc and closed 79 (out of 108) doc bugs. This story summarizes the experience and guidance to run your own hack-a-doc.

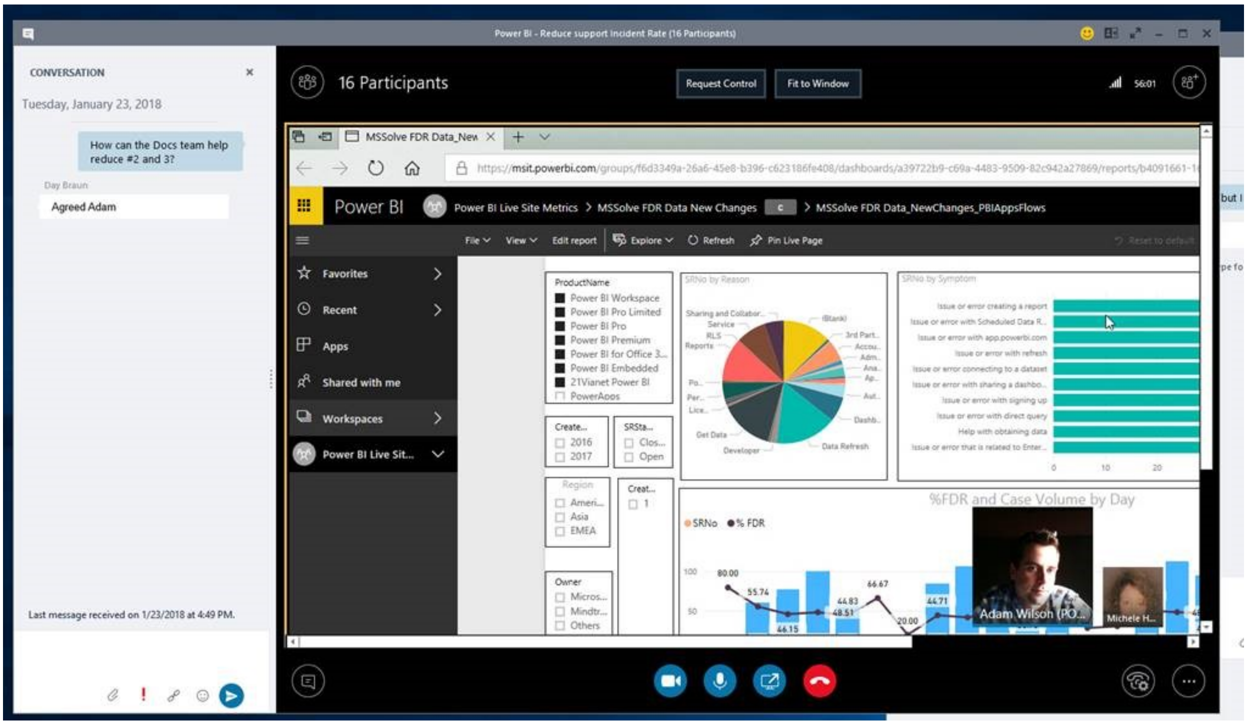
Background

The Power BI hack-a-doc (HaD) focused on deflecting support volume, but hack-a-docs can be used in many other situations: closing bugs, improving freshness, getting consensus on mockups, prioritizing work, and more. The beauty of a hack-a-doc is that it happens in real time and coordinates input from a large group of people across multiple organizations.

The Power BI hack-a-doc Support teams (CSS, Supportability, Serviceability) and Dev team (PMs, developers, Marketing) were faced with an overwhelming amount of support-related work. In many cases, more hours of “support” ticket work than work hours in a week. There were also an incredibly high volume of doc comments, Support site activity, Disqus comments, doc feedback, and Community Forums questions. The Support overhead was crippling the organization.

The Dev team undertook a study to look at the metrics for the high volume. The research number that got our attention was 23% of the volume could be reduced with documentation and user education. (later research showed volume as high as 40%).

Row Labels	Count of ServiceRequestNumber	Count of ServiceRequestNumber2
Gateway built-in Diags	135	16.11%
DOCUMENTATION	125	14.92%
USER EDUCATION	91	10.86%
BETTER ERROR	57	6.80%
External	49	5.85%
Customer Abandoned	45	5.37%
Still active	44	5.25%
External -Authentication	27	3.22%
advisory	23	2.74%
(blank)	22	2.63%
MFA Enablement	19	2.27%
Customer Resolved	17	2.03%
TESTING	17	2.03%
DCR	17	2.03%
COMMS	11	1.31%
DISCOVERABILITY OF DOC	11	1.31%
BUG	11	1.31%
Self service - Whitelisting	9	1.07%
PBIT	9	1.07%
Unknown	8	0.95%
User Error	8	0.95%
Clear token/cache options	7	0.84%
Content Pack Reliability	6	0.72%
Intermittent issue - Self Resolved	6	0.72%
User resolved	6	0.72%
External - Connectivity	5	0.60%
Content Pack Authentication	4	0.48%



Prep work

The writing team met with members of the Support and Dev orgs and discussed plans and strategies. One of these plans was to take the 23% tickets out of the proprietary tool, anonymize them, and create a spreadsheet. Few Microsoft employees have access to ticket information because of compliance and security restrictions, the help of the Support org was critical.

73	1248953	117100416444949	websites.net/?typ	Audit Log Access Delay	Y	Explicitly note that it can take up to 48 hours for audit log			DISCOVERABILITY OF DOC
74	1219274	117100816460245	websites.net/?typ	Orphaned Content Packs cannot	N	Update article to note that if the app		ki/service-organizat	DOCUMENTATION
75	1235153	117101016471680	websites.net/?typ	Resource Not Allowed - SPO cus	N	Document error message" Correlation id: undefined, http			DOCUMENTATION
76	1249444	117101016473107	websites.net/?typ	Advisory - Filtering import query		Unable to quickly find documentation on use of paramete			USER EDUCATION
77	1249556	117101091305201	websites.net/?typ	Data Import - From Folder					USER EDUCATION
78	1251155	117101116474400	websites.net/?typ	Gateway upgrade					USER EDUCATION
79	1219127	117101216485120	websites.net/?typ	Timeout on refresh	Y	N/A	User Effort	esh-troubleshootir	DOCUMENTATION
80	1229056	117101216486385	websites.net/?typ	Advisory - Design	N	How To: Options for storing and displaying images in Pow			DOCUMENTATION

Create Azure Boards bugs

The support items in the spreadsheet were converted into Azure Boards doc bugs by one person from Supportability and one person from Content and assigned to writers. To streamline, we used an existing chart that identified writers by their content areas.

	PM	Area	Notes	Writer	CSS	Marketi	Dev	UX	EE	K
15	Chris F	Data refresh		David	Didier/Bal	Jacob Grir	Doina B	mattlo	Brian Schumer's team	
16	Chris F	Email subscriptions		Michele	Didier/Ra	Jacob Grimm			Brian Schumer's team	
17	Chris F	Email subscriptions		Michele	Bala	Jacob Grir	parianoo	mattlo	Brian Schumer's team	
18	Chris F	HOME onboarding		Michele	Didier/Ra	Jacob Grimm			Brian Schumer's team	
19	Chris F	Pro trials		Mark?	Didier	Jacob Grir	parianoo	mattlo	Brian Schumer's team	
20	Chris F	query strings		Michele	Didier	Jacob Grir	parianoo	mattlo	Brian Schumer's team	
21	Chris F	real-time streaming datasets		David?	Tushar	Jacob Grir	parianoo	mattlo	Brian Schumer's team	
22	Chris F	Usage metrics		Michele	Tushar	Jacob Grir	parianoo	mattlo	Brian Schumer's team	

and a page we created in the CSS Power BI wiki that had links to Azure Boards bug templates by author and/or content area.

So a single-click to open a **pre-populated** bug template.

Create a VSTS documentation ticket [edit]

Use the following templates to create a **new documentation article** request or a **documentation bug**. Change the **Iteration** field to reflect the due date a
team -- for example, preface the title with "*CSS*"-<Bug Title>.

Area	Create	Default owner
Analysis Services - Azure, SSAS, PBIPrem	New article or Doc bug	Owen Duncan
DAX Reference	New article or Doc Bug	Owen Duncan
Power Query/M Reference	New article or Doc Bug	Owen Duncan
Power BI Desktop	New article Doc Bug	David Iseminger
Power BI service	New article Doc Bug	Michele Hart
Power BI Premium, Admin, gateways	New article or Doc Bug	Michael Blythe
Power BI Dev (including PBIE and CV)	New article Doc Bug	Mark Ghanayem
Power BI Report Server/Reporting Services	New article or Doc bug	Maggie Sparkman, Mark Ghanayem
Power BI mobile	New article Doc Bug	Maggie Sparkman
Power BI Miscellaneous	Doc bug or request for new article	Power BI content team

These items will be assigned to a doc owner to work on. When an item is created, or updated, the assigned owner will be notified by an email notification.

There were some bugs that were hard to assign, so we used our weekly PM:UE Sync meeting to fill in the blanks. Additionally, we had one writer act as triage. If a PM or writer had an issue with an assigned bug (such as wrong owner or not enough info), the triage person was responsible for finding the right owner or working with Support to dig into the ticket for more info.

Bugs:

- Created in Technical Content database and we ensured all parties had access
- Used already-created doc bug templates that had as many fields pre-populated as possible, including writer, iteration, Problem description, and problem solution.
- Created tags that we could use in queries and metrics: “Support Volume” assigned to all bugs, “Support Solution” assigned to lower-pri bugs that already has a solution as part of the bug.
- Worked with PM team managers to come up with a list of content area tags (remember, the PMs and Devs are the specialists so these tags will help them the most) and added these additional tags (for example gateway, refresh, connectors, DAX) to help participants pick bugs to work on
- Updated spreadsheet with Azure Boards IDs
- Created Azure Boards queries for “Support Volume” and “Support Solution.”

In total, the spreadsheet recorded 840 Support tickets, and 140 were identified as being solvable with docs and UE, and 108 were able to be converted to Azure Boards doc bugs. Not all, because many support tickets just didn’t have enough information to create an actionable doc bug.

Hack-a-doc idea envisioned vs reality

There was team-wide urgency and pressure to get these bugs closed quickly. Erin suggested a Hack-a-doc.

We started with assumptions and plans that we quickly realized would not work. In the final week, we completely redesigned the HaD.

Ideas we ended up dropping

1. Logistics – set up triads of people (CSS + PM + writer) in advance, and pre-assign bugs. Dropped because attendees would be coming and going, many PM "owners" were away at conferences, too much up-front work. We wanted this experience to be *frictionless*.

2. For each HaD bug, assign the "owner" PM and writer. Dropped because not all PMs could attend and/or assign replacements. Didn't fit our goal of *frictionless*.

3. Do the edits in real time using **our** tools and process (such as Git), which meant sending installation/setup instructions ahead of time. Rejected because using our tools proved to be a barrier to participation for everyone but writers.

In my time at Microsoft I've been involved in at least two other hack-a-docs that I can recall, even spent time creating and prioritizing lists of articles for these hack-a-docs, and neither was successful....no one showed up. So we were nervous.

Sample email invitation

> Hello, thank you for participating in our hack-a-doc. We will be there at 7 a.m. to work with people in advanced time zones, but those members attending in person are welcome to start at 8am (late arrivals are still welcome).
>
> **The high number of support calls and posts and tickets is putting a large strain on the Power BI organization. This hack-a-doc is intended to help reduce that strain by adding oft-requested information to the Power BI docset.**
>
> This Azure Boards query returns the first set of bugs that we'll tackle at the hack-a-doc. These bugs were collated by the Serviceability team and categorized as issues that could have been solved with documentation. For the hack-a-doc, we'll create ad-hoc teams as people arrive in the room or online. Each two-person team will consist of one writer and one non-writer who will work together to resolve bugs, update existing Power BI docs, and write new docs if necessary. RSVP as soon as you can.
>
> If you are in the Redmond area, we'd prefer that you attend in person....THERE WILL BE FOOD!
>
> To prepare for the hack-a-doc, you may want to do the following tasks. This is optional.
>
> 1. Sign up for GitHub, it's quick and free.
>
> 2. Go to https://repos.opensource.microsoft.com and join MicrosoftDocs (if you haven't already joined).
>
> 3. Open the Azure Boards query.
>
> 4. Review the list of bugs and pick out some that fall into your area of expertise. For example, look at the bug title and tags and if those descriptors don't help, open the bug itself. Come to the hack-a-doc with a list of bugs you'd feel qualified to work on.
>
> For remote employees, we'll still put you on teams and create Skype meetings, but won't be able to provide food.
>
> I'll send out more information over the next eight days. Please forward the meeting invite to anyone I may have missed.
>
> Thank you for helping to reduce support volume, Michele (remote, no food for me), and Michael (on-site Hack-a-doc facilitator)

Identify participants

Start with your PM, Dev, Support, Marketing, and other partners and network out from there. Contact managers to help identify invitees. Continually ask invitees to suggest others who would be an asset to the effort. If someone can't make it, ask them to recommend a replacement. In spite of this effort, you'll inevitably leave people out and miss people...we did. Apologize and ask if you can follow up with them after.

Hack-a-doc details

As early as possible (and repeat), communicate the goals of the Hack-a-doc and how you expect to achieve them. Entice people with food and prizes. Make the experience as easy as possible - limit pre-work, requirements, processes. Communicate often and using different tools and strategies, such as messaging or calling individuals directly, meeting with managers so they're onboard and a partner in achieving the goal, and mentioning in other meetings.

Schedule to accommodate different time zones and allow attendees to come and go (or participate even if they only have one hour available).

For our team and audience, we quickly realized that since we weren't going to have participants in the same room for the entire five hours, creating working teams ahead of time would not work. We switched to a more free-flowing ad-hoc model and communicated far and wide.

Our reasoning was that only the PMs are Power BI "specialists" but the Support folks have a breadth of knowledge and could work on almost any bug. Same for writers, writers can write about anything as long as they have a knowledgeable partner. So we moved from pre-assigned teams of three to ad-hoc duos. And we added a facilitator in the room who assigned duos as people arrived in person or on Skype. When anyone was free, they posted on Skype and let the facilitator know. We left the Skype meeting open for conversations and messaging.

We did little doc editing in real time because it slowed the process of collecting solutions. Instead, we focused on getting the experts to the table and writing down the ticket solutions any way possible. In some cases, we edited the docs directly in VS Code (Fantastic). In some cases, we edited the docs in the browser (Great). And, in most cases, we took notes in the Azure Boards ticket, in Word, in email, and in OneNote (perfectly wonderful). For the latter method, the writer in the duo was responsible for claiming ownership of the bug in Azure Boards, adding the notes, and then reassigning the bug back to the doc owner. Any notes outside of Azure Boards were added to the Azure Boards bug before assigning back to the doc owner.

After the Hack-a-doc event

Once the actual hack-a-doc was done, writers concentrated on applying the solutions to the docs, closing the bugs, and pushing out PRs. The hack-a-doc was on a Wednesday, and we had a goal to close the bugs we worked on within two business days (if they weren't already closed). Because the doc bugs for the hack-a-doc were created by only two team members, it was their responsibility to set the bugs to closed once the doc owner marked the bug as complete. Work continues on the remaining few.

The prize was awarded using the participant list and a randomly generated number. A follow-up email was sent to all participants thanking them for their help before, during, and after the hack-a-doc. A post-mortem was held by the writing team and, less-formally, with Support and Product teams.

What next?

Ensure that you have processes in place to measure the success of your hack-a-doc.

1. track tickets - volume, category
2. track "tree" solutions - how many involved docs? How many involved a doc with a "Support" tag?
3. Track volume of Comments for each updated doc
4. Redo initial investigation of support tickets categorized by solution. Are docs and user education still 23% or more?
5. From Support site - track keywords, paths, doc hits (especially docs updated during hack-a-thon)
6. In many cases, tickets were identified as having a doc solution but user did not even search. Or, user searched but results didn't surface correct docs. This information should guide our next effort - SEO and redesign of Support site (already partially completed) so customers are unable to submit a ticket without at least searching, seeing a list of docs that might answer the question.

Key takeaways

1. Freeform ad-hoc worked better than structured
2. Having a facilitator

3. Letting participants know that their attendance for the full 5 hours is not required – people will come and go
4. Limit the requirements - we started with many (for example sign-up for our repo, install Git and markdown tools, review all bugs and add your name in Azure Boards) and ended up removing them ALL. They were barriers to participation.
5. Only the PMs and Devs are “specialists” so structure the hack-a-doc around them
6. Don’t have someone else set up the meeting invite. We were not able to make changes to the logistics and to attendees getting invited at different times and receiving different invitation details.
7. Communication: email, call, IM over and over again. As much 1:1 as possible, no matter the time zone. Explain the win: win benefit, entice with food and prizes, emphasize that managers and up are supporting this event and even attending, ask for substitutes if someone is unable to attend, follow-up, remind. Repeat and repeat.
8. Expect pushback and be ready with the value prop. Win: win. "Yes, you’re overloaded, that’s one of the reasons we’re doing this event. Just commit to an hour or two of your time and together we can make this better." *Hi so-and-so, Just confirming that you will be attending the hack-a-doc on May 30. Sounds like you’ll be a key member of the project. Are there other PMs and Devs you recommend who could join us and help with these types of doc updates? Is there a particular CSS Engineer you work with?*

Wins (in addition to closing 79 bugs)

The Hack-a-doc was a wonderfully collaborative experience with members of CSS, PMs, Devs, Supportability/Serviceability, Marketing, and writers working in ad-hoc teams to tackle a problem. In addition to closing bugs, we forged new relationships across teams and, I believe, developed more appreciation for the role we each play in making Power BI successful.

- Improved Support tickets with “trees” being used again
- Improved Support tickets with new categories created to align better with PBI components (another level of dropdowns)
- Increase in PMs and Support contributing/writing docs and submitting doc bugs
- Invited to Support monthly presentations to teach engineers how to update docs.
- Complete engagement between Content and CSS with CSS committing to not only improved bug details BUT to submitting doc bugs, cutting out the middleman (Devs, PMs). CSS is giving a $50 monthly prize to the engineer who opens the most doc bugs in Azure Boards or who makes the most doc updates using the browser version of GitHub. Basically, one writer tracks CSS contributions to docs and picks a winner, communicates that to Support Manager (Day Braun) who awards the gift card.

Case Study 2 - HDInsight docs freshness hack-a-doc

Several content developers planned a one-day hack-a-doc event for our partner product group PM team. After several months with no assigned resources on a technology area, Azure HDInsight, we found 89 of the top 100 HDInsight docs were outdated, that is abandoned or stale. We needed to identify a way to get help from experts that know the product fairly well and can help update the docs with the latest revisions to freshen them up.

In the one day event, we made a huge dent on our HDInsight doc freshness by reviewing 69 docs, editing 65 of those docs during the session via pull requests. We reached 75% of the high priority stale doc pages in our to-do list.

Best practices we followed:

1. **Set a theme that is focused and general enough to be inclusive** – our stated goal was to "Improve doc freshness" with a secondary goal of "shift PM culture to contribute to docs more often"
2. **Divide and conquer the planning effort**, so that one person doesn't have to handle all the stressful planning. Someone handle logistics and food purchasing, someone else handle planning and communication with stakeholders, someone else handle the instructional printouts and work assignment,

someone steer the group and offer feedback, etc.

3. **Get the buy-in from the management** on your team and the partner teams. Need to be sure that there is enough common interest to make the event worthwhile. Ask for time budgeting to make sure enough people can attend to make critical-mass, and financial budgeting to get money for food and swag.

4. **Ask the partner team's leadership to evangelize** the event ahead of time, and **request that they attend the event**. Having the team's GPM and their Partner PM director attend our event really kept people motivated to attend and contribute.

5. **Keep to-do lists and notes** in a simple, bite-sized way. A shared OneNote hosted in a Teams share worked well since everyone could type notes simultaneously and see the state of our to-do lists in near-real-time. Its light enough that we didn't fight with permissions or too many fields in DevOps items.

6. **Plan lists of priority docs to work on**. Use metrics to plan priorities (page views, customer satisfaction, bounce rate, etc.) Having some predefined categories of work sorted top-down in priority order in different OneNote tabs helped us match the subject-matter expertise of the attendees. Everyone had something to do without overlapping with each other.

7. **Book the right size conference room and the right location** that is near the team you are targeting. Easy location allows the team flexibility to come and go for high priority meetings. We had Teams phone call open for any remote participants. One of our content developers flew in for the week to help facilitate the event in person.

8. **During the event, do as much live editing of docs as possible.** Get the changes in Pull Requests to avoid creating an even longer backlog of work. Edit in the browser for light edits, using the private repo to get Acrolinx and preview builds. When the work was too much or too complex to finish, we used OneNote to making clear notes on what follow-up is needed, and track the follow-up work in Azure DevOps after the event was over.

9. **Celebrate often** – Every time someone did a pull request, we wrote their name on a raffle ticket into a hat, for a prize drawing at the end of the day. This activity became a competitive energy building tool. High energy and focus are important for events like this. We took a few photos to capture the memory as well.

10. **Measure results and communicate** - We counted our accomplishments (pull request numbers are easy to quantify in our OneNote lists), and will trend our CSAT and freshness metrics to show a clear before and after. We sent out summary emails to the team and their leadership to explain the effort and our results.

Case Study 3 – Cosmos DB docs content refactor hack-a-doc.

Azure Cosmos DB team had a hack-a-doc to refactor the API/sub-service specific content. It was a three-day event with audience from Content, PM, Dev & CSS teams. We were successful in splitting the content by API and fill in some known gaps. We are 60% complete on the items we tracked for this event. This hack-a-doc was remote only. In addition to the tips from other case studies, the following are some considerations if you are running a hack-a-doc over teams or in a distributed environment:

- Make sure to sync with the teams ahead to get the inventory of work items to tackle during the event. Assign owners and SMEs for different items ahead.

- In addition to heavy/medium weighted tasks, track some work items with minor edits. These tasks are helpful for first-time contributors to get started without much effort.

- If the hack-a-thon is for the whole day or a set of days, have a clear agenda for each day. Explicitly callout if you plan to hold a short session on how to contribute to docs and the time it's time slot. New contributors can make time for it.

- Offer the flexibility for team members to leave the call and go back to their desks to work on the content. They can also have offline meetings with a smaller group of SMEs while making the content updates.
- Keep a track of the Teams conversation to address any questions. It's best to use the same Teams link if your hack-a-thon is multiple days as the conversation history is saved.
- Have progress check-ins twice a day to see how things are going, and if there are any blockers.
- Use 30 min at the end of the event for show and tell, where all the contributors can show their content updates and PRs.
- After the event, measure the success and celebrate. Get the metrics on number of files added, updated in the release branch, and items completed in the spreadsheet. Send a summary email with the highlights, and learnings.
- You can offer swag to the top contributors. Work with the DevRel admin team to arrange the swag, get the shipping labels printed, and ship the swag from one of the Microsoft buildings where shipping facility is available.

Using the Docs experimentation platform to improve my articles

7/7/2021 • 2 minutes to read

WARNING

The content on this page has been deprecated. Contact Sara Lerner for information about how to do user research, including A/B testing, using UserTesting.com.

Docs experimentation platform

Docs has a built-in experimentation platform allowing authors to test in-page article changes with a subset of their audience. All you need is to do is:

- Fill out an experiment form on the CGA site.
- Create one more .md file with the changes and save it in the same directory in GitHub as the original one.
- Add some metadata to both articles.

Once you start the experiment, based on the configuration you provided, a specific percent of users will see the alternative article until the system records the configured number of page views. For example, 1000 page views. The experiment automatically stops and all users only see the original article.

During and after the experiment, you can see all the stats:

- Verbatim comments.
- LiveFyre comments.
- Ratings on the Microsoft Docs Metrics tool (MDM).

You can use the metrics reported in MDM to determine if you want to go mainstream with those changes and stick with the original.

Finish instructions on learning when and how to run an experiment on the Docs platform are available in the following resources.

PowerPoint Deck:

- A/B testing for everyone, in 3 steps

If you have questions, email CGA Support.

What is an experiment?

What makes an experiment different than other types of data analysis are three parts: making a *controlled change* to the user experience, making a *prediction* about the effect (the hypothesis), and intentionally structuring *data capture* to measure the actual effect. Experimentation may sound intimidating, but is actually an approachable and intuitive way for non-data scientists to become more data-driven. Why? Because a well-designed experiment yields results that are easy to interpret. Simply comparing the predicted effect to the actual effect is where the learning happens. We invite you to give it a try!

The CSI data science team will partner with writers and PMs to enable the following tests:

- A/B tests
- Before vs. After tests
- Email-driven experiments
- Usability tests
- Surveys
- Click-pathing experiments

Docs Build validation overview

6/24/2021 • 4 minutes to read

Whenever you submit a pull request to contribute to docs.microsoft.com, Docs Build runs a set of validation rules to check the quality of the contribution. Some validations are built into the Docs Build engine. Some are "custom" validations that are configured in an external service and then integrated into Docs Build. The main difference is that custom validations can be configured into rulesets. Different rulesets can be applied to different Docs repos. Once applied to a repo, the user experience for custom validations is identical to the experience for other Docs Build validations: Results appear the same way in PR conversations and build reports.

For example, here's a PR that contains both Docs Build validation results (`Author cannot be recognized in GitHub` and `Invalid file link`) and custom validation results (`Missing attribute` and `Invalid value for ms.date`):

Validation status: warnings

File	Status	Preview URL	Details
articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Warning	View	Details

articles/active-directory-b2c/active-directory-b2c-access-tokens.md

- [Warning] `Missing attribute: either ms.prod or ms.service is required. Use ms.prod for on-premise products, or ms.service for cloud services.`
- [Warning] `Invalid value for ms.date: '04/160/2019'. Must be a date in format MM/DD/YYYY, no more than 30 days from today.`
- [Warning] `Author mmmacy cannot be recognized in GitHub.`
- Line 18: [Warning] `Invalid file link:(~/articles/active-directory-b2c/active-directory-b2z-reference-tokens.md).`

For more details, please refer to the build report.

If you click the **build report** link from the GitHub comment, you'll see your PR results and any outstanding results in the repo. Many rules include error codes linked to detailed documentation. To provide doc feedback, click the speech bubble next to the error code to file a GitHub issue in the Docs Feedback repo.

Severity	Type	Code (click link for details)	File	Message
Warning	User	ms-prod-or-service-missing	articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Missing attribute: either ms.prod or ms.service is required. Use ms.prod for on-premise products, or ms.service for cloud services.
Warning	User	ms-date-invalid	articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Invalid value for ms.date: '04/160/2019'. Must be a date in format MM/DD/YYYY, no more than 30 days from today.
Warning	User	GitInfoNotFound	articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Author mmmacy cannot be recognized in GitHub.
Warning	User	InvalidFileLink	articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Invalid file link:(~/articles/active-directory-b2c/active-directory-b2z-reference-tokens.md).
Warning	U:	ms-author-invalid Click to provide feedback for Code or Message	articles/cosmos-db/how-to-use-...ns-gremlin.md	Invalid value for ms.author: 'olignat' is not a valid Microsoft alias.
Suggestion	User	h1-missing	articles/cloud-shell/persisting-shell-storage.md	H1 is required. Use a single hash (#) followed by a space to create your top-level heading.

Currently, custom validation is supported for English conceptual Markdown articles in Docs Build v2 and Docs Build v3. A subset of custom validation is available for Learn included Markdown files in Docs Build v2, but not yet for Learn YAML files. Custom validation for reference is expected to be available in FY21 after reference migrates to Docs Build v3. Custom validation isn't available for localized repos because content issues should be fixed in the English source.

A new severity, "Suggestion", has been implemented to help roll out new validations. New validations are initially released as Suggestions, which don't impact build or block workflow automation such as PRMerger. After eight weeks they'll be escalated to Warnings, which don't block builds but do block PRMerger. If a new validation has unusually high impact and more time is required for initial cleanup, we might extend this period.

Custom validation rulesets include *metadata rules* and *content rules*.

Metadata rules

Metadata rules check for the presence and validity of various attributes, mostly used for content reporting. Different rulesets have been defined for different content areas. For example, some metadata requirements are global across the Docs platform, while others only apply to repos that use certain content reporting features.

Metadata can be set in multiple ways: specified directly in the Markdown file, specified at a folder level in docfx.json, or specified globally in docfx.json. Validation respects this hierarchy and returns results appropriately.

Metadata rules aren't yet available for Learn included Markdown files.

Content rules

Content rules check for common Markdown mistakes that cause Docs site problems, including broken content, accessibility bugs, and localization problems. Most content rules are global across the Docs platform.

Custom rulesets

There are two main custom rulesets in use for conceptual Markdown content:

- The Docs Platform ruleset, which is required for all repos unless an explicit exception is granted.
- The Developer Relations/CGA reporting ruleset, which is a superset recommended for all conceptual repos and required for repos that use CGA reporting.

The Docs team can create alternative rulesets for teams outside of Developer Relations only if the teams have different reporting requirements from Developer Relations. These exceptions must be approved by the Docs Governance Committee.

Docs Platform ruleset

The Docs Platform ruleset checks that an accountable owner is available for each article on Docs, and that basic content standards are met. It's applied to all English conceptual repos by default.

The Docs Platform ruleset contains the following rules:

- author-missing
  - Additionally, native Docs Build checks that the value of `author` is a valid GitHub ID.
- h1-empty
- h1-in-moniker
- h1-missing
- h1-no-space
- h1-not-first
- hard-coded-locale
- insecure-link
- ms-date-invalid

- multiple-h1s
- title-missing

Developer Relations/CGA Reporting ruleset

The Developer Relations/CGA Reporting ruleset contains the following rules:

- author-missing
  - Additionally, native Docs Build checks that the value of `author` is a valid GitHub ID.
- deprecated-attribute
- h1-empty
- h1-in-moniker
- h1-missing
- h1-no-space
- h1-not-first
- hard-coded-locale
- insecure-link
- ms-author-invalid
- ms-author-missing
- ms-date-invalid
- ms-date-missing
- multiple-h1s
- 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
- title-missing

To request onboarding to the full Developer Relations/CGA Reporting ruleset, file a work item via this template.

Learn ruleset

The Learn ruleset only runs on included Learn Markdown files. Metadata validation isn't yet supported for Learn, so only content rules that apply to included files are included.

- hard-coded-locale
- insecure-link

Other custom rulesets

Alternative rulesets have been enabled for the Office Developer, Marvel, and Protocols teams. There's also an "approved-exceptions" rulesets for non-conceptual repos that had unexpected results with the Docs Platform ruleset.

To request onboarding to one of these rulesets, file a work item via this template. Be sure to specify which ruleset is appropriate for your team.

For more information about alternative rulesets, contact mbradley.

allowlists

Some validations, including metadata validations for `ms.prod`, `ms.service`, and `ms.topic`, check values against an allowlist. You can view allowlists on on the Taxonomy lists page.

To request new allowlist values, follow the process described in Metadata changes.

PR only validation rules

11/2/2020 • 2 minutes to read

A challenge for rolling out automated validation rules has been that Docs repos contain many thousands of articles, many of which aren't actively maintained. To turn on a new validation rule, we used to have to fix the issue in all existing files to avoid too much noise in the repos. This meant we often missed catching issues in new and updated content because the impact of reporting on old content was too high.

We now have the ability to turn on rules as "PR only". This means results only appear in validation reports if the issue was found in files in a PR. Older files aren't checked until they're touched. This allows content contributors to focus on the content they're working on now, without having to worry about dormant content.

For rules enabled as PR only, such as description-missing, you'll see the following validation behavior in PR comments and build reports:

- In the PR comment, you'll see the PR only result like any other validation:

File	Status	Preview URL	Details
help-content/contribute/validation-ref/alt-text-bad-length.md	Suggestion	View	Details

  help-content/contribute/validation-ref/alt-text-bad-length.md

  - Line 2, Column 1: [Suggestion-description-missing] `Missing required attribute: 'description'. NOTE: This Suggestion will become a Warning in late October, 2020.`

  For more details, please refer to the build report.

- If you click the **build report** link from the PR comment, you'll see the full build report for the repo. In the **Validated Files** table, you'll see the files in the PR and any validation results, including PR only results:

  Validated Files

File	Status	Preview URL	Details
help-content/contribute/validation-ref/alt-text-bad-length.md	Suggestion	View	• **Line 2: [Suggestion]** Missing required attribute: 'description'. NOTE: This Suggestion will become a Warning in late October, 2020.

  The **Details** table in the build report shows all outstanding Errors, Warnings, and Suggestions in the repo, *not* including PR only rules.

- If you run a full branch build, for example by forcing a build via Docs Portal, the build report won't contain any PR only results.

If you'd like to fix all issues in your content including PR only rules, you can request a test report by contacting mbradley.

alt-text-bad-length

3/5/2021 • 2 minutes to read

Applies to: All content types

This validation catches alt text that's outside the recommended range of 40 - 150 characters (these values are configurable). Because this range is currently a guideline and not a hard requirement, this validation hasn't been enabled on a production ruleset for Docs or Learn at this time. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Alt text '{text}' is out of range at {length} characters. Alt text should be between {min} and {max} characters.
```

The min and max values are configurable, but 40 - 150 (inclusive) is recommended. Alt text of fewer than 40 characters is generally not descriptive enough to meet Accessibility guidelines, and alt text longer than 150 characters is clumsy for screen readers. Although these aren't hard requirements, running this rule on your repo can give you an overall picture of how good your alt text is, and help identify areas for improvement.

For more information about accessibility requirements for images, see Accessibility guidelines for multimedia.

Resolution

The following examples show how to add alt text for images in standard Markdown and in the Docs `:::image:::` Markdown syntax with `type='content'` (for standard conceptual images):

```
Standard Markdown image:

![Screenshot showing how to force a build from the Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.](media/force-build.png)

Docs Markdown content image:

:::image type="content" source="media/force-build.png" alt-text="Screenshot showing how to force a build from the Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.":::
```

If your image is too complex to meaningfully describe within 150 characters, you should use the Docs `:::image:::` Markdown syntax with `type='complex'`. For complex images, the alt text should be a high-level description of the image. The image content should describe in detail what the image conveys:

```
:::image type="complex" source="../media/page-views.png" alt-text="Graph showing page view trends.":::
   <Detailed description of the page view data shown in the graph goes here...>
:::image-end:::
```

If your image is purely decorative, such as a product icon, alt text isn't required, but you must use the Docs `:::image:::` Markdown syntax with `type='icon'` to indicate that the image is decorative:

```
:::image type="icon" source="../media/docs-icon.svg":::
```

> **TIP**
>
> To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

alt-text-bad-value

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

`Alt text for an image can't be the same as the image file name.`

Image alt text must meaningfully describe the content and intent of the image. Using the name of the image as alt text provides no additional information and isn't allowed.

Resolution

Alt text must accurately describe the purpose of the image in at least 10 and no more than 250 characters. Between 50 and 150 characters is recommended. The following examples show how to add alt text for images in standard Markdown and in the Docs `:::image:::` Markdown syntax with `type='content'` (for standard conceptual images):

```
Standard Markdown image:

![Illustrates how to force a build from the Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.](media/force-build.png)

Docs Markdown content image:

:::image type="content" source="media/force-build.png" alt-text="Illustrates how to force a build from the Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.":::
```

If your image is too complex to meaningfully describe within 250 characters, you should use the Docs `:::image:::` Markdown syntax with `type='complex'`. For complex images, the alt text should be a high-level description of the image. The image content should describe in detail what the image conveys:

```
:::image type="complex" source="../media/page-views.png" alt-text="Graph showing page view trends.":::
   <Detailed description of the page view data shown in the graph goes here...>
:::image-end:::
```

If your image is purely decorative, such as a product icon, alt text isn't required, but you must use the Docs `:::image:::` Markdown syntax with `type='icon'` to indicate that the image is decorative:

```
:::image type="icon" source="../media/docs-icon.svg":::
```

For more information about accessibility requirements for images, see Accessibility guidelines for multimedia.

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

alt-text-missing

3/5/2021 • 2 minutes to read

Applies to: All content types

Suggestion

`Missing alternate text for image '{image path}'. Alt text is required for accessibility.`

Screen readers rely on alternate text so that users with visual impairments can hear a meaningful description of an image.

Resolution

If your image is intended to convey meaning, you must provide alt text. The alt text must accurately describe the purpose of the image in at least 10 and no more than 250 characters. Between 50 and 150 characters is recommended. The following examples show how to add alt text for images in standard Markdown and in the Docs `:::image:::` Markdown syntax with `type='content'` (for standard conceptual images):

```
Standard Markdown image:

![Illustrates how to force a build from the Docs Portal by clicking Manual Publish, specifying a branch,
then clicking Publish.](media/force-build.png)

Docs Markdown content image:

:::image type="content" source="media/force-build.png" alt-text="Illustrates how to force a build from the
Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.":::
```

If your image is too complex to meaningfully describe within 250 characters, you should use the Docs `:::image:::` Markdown syntax with `type='complex'`. For complex images, the alt text should be a high-level description of the image. The image content should describe in detail what the image conveys:

```
:::image type="complex" source="../media/page-views.png" alt-text="Graph showing page view trends.":::
   <Detailed description of the page view data shown in the graph goes here...>
:::image-end:::
```

If your image is purely decorative, such as a product icon, alt text isn't required, but you must use the Docs `:::image:::` Markdown syntax with `type='icon'` to indicate that the image is decorative:

```
:::image type="icon" source="../media/docs-icon.svg":::
```

For more information about accessibility requirements for images, see Accessibility guidelines for multimedia.

> **TIP**
>
> To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

author-missing

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

```
Missing attribute: author. Add the current author's GitHub ID.
```

The `author` attribute identifies the author of the article by GitHub ID.

Resolution

Add the current author's GitHub ID to the YML header:

```
---
author: meganbradley
ms.author: mbradley
---
```

The value should be the *current* owner of the article, not the original author if ownership has changed.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

author-not-found

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

```
Invalid value for author: '{author}' is not a valid GitHub ID.
```

The value specified for author must be a valid GitHub ID.

Resolution

Update the `author` value with the current author's valid GitHub Id. We recommend that the author is a full-time employee, rather than a short-term vendor.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

bookmark-not-found

7/7/2021 • 2 minutes to read

Applies to: All content types

Warning

`Cannot find bookmark '{bookmark-id}' in '{parent-file}'.`

You're trying to link to a heading in the current file or another file that doesn't exist.

Resolution

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Verify the heading you want to link to and update the link.

To link to a section in the current article, use a hash symbol, followed by the words of the heading. Remove punctuation from the heading and replace spaces with dashes.

```
[Managed Disks](#managed-disks)
```

To link to a heading in another file, use a relative link to that file, followed by a hash symbol and the words of the heading. Remove punctuation from the heading and replace spaces with dashes.

```
See [the Resolution section in h1-empty](h1-empty.md#resolution).
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

code-block-indented

4/27/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Indented code blocks aren't allowed. Use a Markdown code block surrounded by triple backticks.`

Code blocks created by indentation aren't allowed on Docs, because you can't specify a code language for correct colorization in indented code. Indentation also isn't allowed as a way to create other block text, because indented text renders as code and isn't localizable.

Resolution

If the indented text is intended to be a code block, use a Markdown codeblock with a valid devlang instead:

````
Before:

        public static void Log(string message)
        {
            _logger.LogInformation(message);
        }

After:

    ```csharp
    public static void Log(string message)
    {
        _logger.LogInformation(message);
    }
    ```
````

Supported devlang values are listed in the devlang allowlist.

For short lines of code, you can use inline code with single backticks:

```
Before:

        I want this to look like code.

After:

`I want this to look like code.`
```

If you indented a paragraph for emphasis, use an appropriate localizable style instead, such as a note or a quoted plain text paragraph:

```
> [!NOTE]
> Use an alert such as a note when appropriate.

"Use quotes if appropriate... Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Etiam efficitur id orci ac euismod.
Cras risus justo, molestie ac nulla eget, porttitor aliquam
lectus. Aliquam eu ipsum vestibulum, tempus ante at, porta dui."
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

code-block-unclosed

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

````
Unclosed code block. Code blocks must begin and end with triple backticks (```).
````

Resolution

Close your code block with triple backticks:

````
```md
Here's a properly closed Markdown block.
```
````

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

column-header-missing

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Column headers are required for tables. Add appropriate header text, or remove any extra table cells that
cause an extra column.
```

Screen readers require column headers. Without headers, users who can't visually scan the whole table before reading won't know what to expect from each column.

Resolution

Add a header to each column that meaningfully describes the content of the column. Here is the Markdown for a basic table with headers:

```
|header 1 |header 2 |header 3 |
|---------|---------|---------|
|content 1|content 2|content 3|
|and......|so.......|on.......|
```

In some cases, extra table cells cause this message. The extra table cells create an extra column without a header. This case can be difficult to spot in Markdown. This table, for example, looks pretty good at a glance:

```
|header 1 |header 2 |header 3  |
|---------|---------|----------|
|content 1|content 2||content 3|
|and......|so.......|on........|
```

But there's an empty cell in the first row, which creates a fourth and header-less column on the published page:

header 1	header 2	header 3	
content 1	content 2		content 3
and......	so.......	on........	

If you can't immediately see what's wrong with your Markdown table, check the file preview on review.docs.microsoft.com.

Data matrix tables

A header might not be appropriate for the left column of your table if the table is a data matrix - that is, a table with both a top header and a header column.

To create a data matrix table, leave the first column header blank and bold the text in the rest of the cells in the first column:

```
|                    |top header 1 |top header 2 |top header 3|
|--------------------|-------------|-------------|------------|
|**column header 1** |content 1a   |content 2a   |content 3a  |
|**column header 2** |content 1b   |content 2b   |content 3b  |
|**column header 3** |content 1c   |content 2c   |content 3c  |
```

TIP

The Docs Authoring Pack for VS Code includes a function to convert a regular Markdown table into a data matrix table. Just select the table, right-click, and select **Convert to data matrix table**.

What about custom layouts?

The `:::columns:::` extension is no longer recommended for creating custom layouts. We've found that many custom column layouts have accessibility issues or otherwise violate Docs style guidelines. Don't create custom layouts. Use standard Docs features.

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

dependent-repo-branch-mismatch

6/28/2021 • 2 minutes to read

Applies to: All content

Suggestion

```
The referenced branch '{branch}' in dependent repo '{repo}' can't be found. Update the branch value of the
dependent repo in the .openpublishing.publish.config.json file.
```

Cross repository references can be used to pull content and other assets into the current repo from another repo. These references are configured in the .openpublishing.publish.config.json configuration file. This Suggestion means that the entry for the specified dependent repo specifies a branch that doesn't exist.

Resolution

Go to the dependent repo to confirm the branch name (usually the default branch, such as `main`). Then update the `branch` property for the repo under `dependent_repositories`:

```
"dependent_repositories": [
   {
     "path_to_root": "repo-name",
     "url": "https://github.com/MicrosoftDocs/repo-name",
     "branch": "main"
     }
   }
 ]
```

For more information, see Cross Repository Reference.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

deprecated-attribute

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`Deprecated attribute: ms.component. Use ms.subservice instead.`

Use `ms.service` to specify cloud services. To specify more detailed information about `ms.service`, you can optionally specify `ms.subservice`. Don't use `ms.component`; it's deprecated for this content.

Resolution

Confirm that your `ms.service` value is correct for your article. Then choose a valid `ms.subservice` value.

Valid values can be found in the ms.service allowlist.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

description-bad-length

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

This validation catches description values that are outside the recommended range of 100 - 160 characters. However, because this range is a guideline and not a hard requirement, this validation hasn't been enabled on a production ruleset for Docs or Learn at this time. If you would like this rule run on your repo, contact mbradley.

> **TIP**
>
> The recommended way to verify that your description will provide meaningful information to users when returned in search results is to use the Search results Preview available in the Docs Preview VS Code extension, which is part of theDocs Authoring Pack for VS Code.

Suggestion

```
Description '{text}' is out of range at {length} characters. Descriptions should be between 100 and 160 characters.
```

Resolution

Write a meaningful description between 100 and 160 characters.

For more information about how to write descriptions, SEO: How to write good meta descriptions.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

description-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Missing attribute: description. Add a meaningful description to be returned in search results. Descriptions
should be between 100 and 160 characters.
```

The `description` attribute is required both for SEO and for site search.

This rule has been enabled for PR only.

Resolution

Write a meaningful description between 100 and 160 characters.

For more information about how to write descriptions, SEO: How to write good meta descriptions.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

dev-language-invalid

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn Markdown

This validation is not yet enabled on production repos because the devlang taxonomy is still in progress. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Dev language '{language}' is not allowed. Specify a valid language slug from the devlang taxonomy. Language values are case-sensitive.
```

Resolution

Supported devlang values are listed in the devlang allowlist. Add a valid language slug. To request new devlang values, email DevRelIA@service.microsoft.com.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

dev-language-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn Markdown

This validation is not yet enabled on production repos because the devlang taxonomy is still in progress. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Dev language missing. You must specify a valid language slug from the devlang taxonomy. Values are case-
sensitive
```

Resolution

Supported devlang values are listed in the devlang allowlist. Add a valid language slug. To request new devlang values, email DevRelIA@service.microsoft.com.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

disallowed-heading

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Heading '{heading-text}' matches the text of a disallowed Docs template heading.
```

Docs generates some article headings, such as "Feedback" and "Is this page helpful?", automatically so that authors don't have to add them manually and users get a consistent experience across the site. If you manually add a heading that matches one of these auto-generated headings, you end up with duplicate headings - an accessibility violation. Screen readers rely on headings to navigate the page for users with visual impairments, so duplication causes confusion.

Resolution

Remove the duplicate heading. If the content of the heading is redundant with autogenerated content, remove the whole section. If the content is unique and required, rename the heading as appropriate.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

docs-link-absolute

4/28/2021 • 2 minutes to read

Applies to: All topic types

Suggestion

```
Absolute link '{URL}' will be broken in isolated environments. Replace with a relative link.
```

Absolute links to other Docs pages will be broken in environments that aren't connected to the internet. They also aren't checked in Docs Build.

Resolution

Use a relative link (for Docs pages in the same docset) or site-relative (for pages in different documents) instead.

For more information, see Add links to articles.

> **TIP**
> The Docs Authoring Pack for VS Code includes functionality to insert relative links, including links to articles in the same repo and site-relative links based on a Docs URL.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-alt-text

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Alt text '{text}' is duplicated. Within an article, alt text must be unique.
```

Users of screen readers rely on meaningful alternate text to understand the purpose of an image. If images with different file names have the same alt text, the alt text doesn't convey enough meaning to distinguish between the images. Duplicate alt text is also a problem for SEO: If multiple images have the same alt text, search engines don't know which image to rank higher in image search.

This rule has been enabled for PR only.

Resolution

Add unique, meaningful alt text to each image. For more information about how to write good alt text, see Accessibility guidelines. For more information about SEO for images, see Improve search rank with diagrams: Image alt text and filename.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-descriptions

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Attribute 'description' with value '{description-text}' is duplicated in '{file-path-1}', ...'{file-path-n}'.
Descriptions must be unique within a docset.
```

For SEO, descriptions must be unique within the current docset, and ideally should be unique across the site.

This rule has been enabled for PR only.

Resolution

Edit one or more `description` so all descriptions in your docset are unique.

For more information about how to write descriptions, SEO: How to write good meta descriptions.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-h1s

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
H1 '{h1-text}' is duplicated with other articles: '{file-path-1}', ...'{file-path-n}'. First level headings
must be unique within a docset.
```

For SEO, H1 headings (`#`) must be unique within a docset, and ideally should be unique across the site.

Resolution

Edit one or more first level heading so all H1s in your docset are unique.

For more information about how to write H1s, see SEO: Tips for writing a heading 1 (H1).

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-h2s

4/28/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Duplicate H2: '{heading text}'. Each H2 in an article must be unique.
```

Duplicated headings on a web page are an accessibility violation. Screen readers rely on headings to navigate the page for users with visual impairments, so duplication causes confusion. This validation checks for duplicate authored H2 headings within an article.

Resolution

Rename H2 headings within your article as necessary so they're all unique.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-pivot-groups

3/5/2021 • 2 minutes to read

Warning

```
Pivot group id '{id}' is duplicated in '{definition-file}'. Only the first group with this ID will be used.
Delete or rename the subsequent group(s).
```

If you define more than one pivot group with the same ID in a pivot definition file, only the first group will be used. Users attempting to reference subsequent groups with the duplicate ID will end up with broken or incorrect pivots.

Resolution

First, try to find all files in your docset that are using the duplicated pivot group ID, and determine which version they are meant to use. Then delete or rename the duplicate(s) and update files as appropriate.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-pivot-ids

3/5/2021 • 2 minutes to read

Warning

```
Pivot id '{pivot-id}' is duplicated in pivot group '{group-id}' in '{definition-file}'. Only the first pivot
with this ID will be used. Delete or rename the subsequent pivot.
```

The specified pivot group contains two or more pivots with the same ID. Only the first one will be used.

Resolution

Remove duplicate pivots or update their IDs.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-titles

4/28/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Attribute 'title' with value '{title-text}' is duplicated in '{file-path-1}', ...'{file-path-n}'. Titles must
be unique within a docset.
```

For SEO, the value of the `title` attribute plus the `titleSuffix` attribute must be unique within the current docset, and ideally should be unique across the site.

Resolution

Edit one or more `title` so all titles in your docset are unique, and/or add `titleSuffix` metadata to add different branding to different folders within the docset.

For more information about how to write titles, SEO: Tips for writing titles.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-toc-entries

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
TOC entry '{file}' is duplicated within a TOC node.
```

Resolution

Only link to each file once in a TOC node.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-uids

3/5/2021 • 2 minutes to read

Applies to: Learn

Error

`UID '{uid}' is duplicated in '{file-list}'.`

Learn UIDs must be globally unique.

Resolution

Verify that each learning path, module, and unit has a unique UID.

Learning path UIDs should follow this format:

```
<prefix>.<learning-path-folder-name>

Such as:
learn.architect-great-solutions-in-azure
```

For more information, see Create a learning path in the Learn Contributor Guide.

Module UIDs should follow this format:

```
<prefix>.<module-folder-name>

Such as:
learn.align-requirements-in-azure
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Unit UIDs should follow this format:

```
<prefix>.<module-folder-name>.<unit-name>

Such as:
learn.add-and-size-disks-in-azure-virtual-machines.introduction
```

Don't include file order prefixes in the unit name, such as "1-introduction". The unit *file name* will include a prefix to indicate its position in the module. The UID shouldn't in case the units get reordered in the future.

For more information, see Create a unit in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

file-not-found

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

`Invalid file link: '{file-path}'.`

The referenced file can't be found.

Resolution

Check the file name and path and update the link. For information about how to use links, see Links.

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

IMPORTANT

Build validation checks relative links within a repo, but for reasons of scale doesn't currently check links to external sites. Also, links can break after content is published. To find all links currently broken in your content, see the broken link report for your repo on Docs Portal. For information about how to use the broken link report, see Broken link report in Docs Portal.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

forbidden-content

3/5/2021 • 2 minutes to read

Applies to: All content

Error

```
Link contains forbidden content. You can't link to air-gapped content. Don't fix this PR - close it
permanently and contact dacoulte@microsoft.com for further instructions.
```

Linking to air-gapped content is a high severity security issue. For more information, see Publishing for the air-gapped cloud (AGC).

Resolution

Close the PR permanently. Don't fix the issue in the same PR. Contact dacoulte@microsoft.com for further instructions to make sure the issue isn't reflected in GitHub commit history.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

forbidden-link

3/5/2021 • 2 minutes to read

Applies to: All content

Error

```
Link '{0}' is forbidden. You can't link to air-gapped content.
```

Linking to air-gapped content is a high severity security issue. For more information, see Publishing for the air-gapped cloud (AGC).

Resolution

Remove the link.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-empty

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`H1 is required. Add content to your top-level heading.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font.

To create an H1, begin a line with a single hash (#) followed by a space, then the heading text.

Resolution

To fix this issue, make sure your H1 includes content, not just a hash.

Bad:

```
---
author: meganbradley
ms.author: mbradley
---
#
This is not an H1
```

Good:

```
---
author: meganbradley
ms.author: mbradley
---

# This is an H1
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-in-include

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn

Suggestion

```
H1 headings aren't allowed in included files. Every parent file should have an H1.
```

Every article must have one and only one H1. The H1 must be in the parent file, not the included file.

Resolution

Remove the H1 from your included file and make sure its parent contains an H1.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-in-moniker

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`H1s are not allowed in moniker sections. Each article should have one and only one H1.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font. An H1 is created by beginning a line with a single hash (#) followed by a space, then the heading text. You can only have one H1 in each file. H1s aren't allowed in moniker sections. The reason is that they can easily cause duplicate or missing H1s depending on how versioning is configured.

You might also get this message if a moniker section has a line of equals signs making a double underline, like this: `=======`. A double underline is an alternative Markdown syntax for an H1. It's also commonly seen in merge conflicts:

```
<<<<<<< HEAD
...
=======
...
>>>>>>> 1d82c7efe18f86136247fb366df5030843199c19
```

Resolution

To fix this issue, remove H1s from all moniker sections. Make sure the H1 at the top of the article is appropriate for all moniker sections.

If an H1 in a moniker section is actually a double underline (`=======`), remove it or replace it with a hashtag heading, like `##`, as appropriate. If the double underline is part of a merge conflict, make sure to also remove the merge conflict beginning and ending markers and the obsolete text.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-missing

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`H1 is required. Use a single hash (#) followed by a space to create your top-level heading.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font.

To create an H1, begin a line with a single hash (#) followed by a space, then the heading text.

Resolution

To fix this issue, add an H1 as the first content after the YAML metadata block in your file:

```
---
author: meganbradley
ms.author: mbradley
---

# This is an H1
```

> **NOTE**
>
> This rule does not apply to included files. If you get H1 results on included files, you most likely need to move your included files into an `includes` folder. The `includes` folder can be at any level in the file path. Based on the path, Docs build will recognize the file as an included file and H1 validations won't run.
>
> A common cause of missing H1s in parent files is misuse of included files: the H1 is in the included file, not in the parent file. This isn't allowed, because using an H1 in an included file either means there are duplicate H1s in parent files or the included file is used only once. H1s should be unique within a content set and included files should only be used to share content among multiple files. If you get `h1-missing` results because the H1 is in an included file, the solution is to move the H1 - and all the included content if the included file is used only once - into the parent file. For more information about included files in Docs, see the Microsoft-internal article Include reusable content in articles.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-no-space

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`A space is required after the hash (#) in H1.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font.

To create an H1, begin a line with a single hash (#) followed by a space, then the heading text.

Without the space after the hash, Docs won't recognize an H1.

Resolution

To fix this issue, add a space after the hash in your H1.

```
---
author: meganbradley
ms.author: mbradley
---
# This is an H1
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-not-first

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`Markdown content is not allowed before H1.`

Only the YAML metadata header can come before the H1 in a Markdown file. For example, you can't put a note before the H1:

```
---
# This is the YAML metadata header
author: meganbradley
---
> [!NOTE]
> You can't do this.

# This is the H1
```

Instead, put the note after the H1:

```
---
# This is the YAML metadata header
author: meganbradley
---
# This is the H1

> [!NOTE]
> This is OK.
```

Another common cause of this issue is byte order marks (BOMs) before the YAML header. BOMs are sometimes introduced by encoding issues when you commit content to a repo. They result in bad rendering in GitHub and should be removed.

Resolution

Remove any content other than the YAML metadata header from before the H1.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

hard-coded-locale

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

```
Link '{URL}' contains locale code '{code}'. For localizability, remove '{code}' from links to most Microsoft
sites.
```

Locale codes, like `en-us`, shouldn't be included in links to certain Microsoft sites. If you include a locale code in a link in English content, it will also be included in localized links, which leads to a bad localized experience. For example, if a link in German localized content includes `en-us`, German customers will find themselves linking to the English article, even if a German version is available.

The following sites are in scope for this validation:

- azure.microsoft.com
- docs.microsoft.com
- msdn.microsoft.com (excluding social.msdn.com, which needs locale to ensure the correct forum is linked to)
- partner.microsoft.com
- technet.microsoft.com

Resolution

Here's how to remove locale codes from links to Microsoft sites:

Before:

```
https://docs.microsoft.com/en-us/vsts/load-test/app-service-web-app-performance-test
```

After:

```
https://docs.microsoft.com/vsts/load-test/app-service-web-app-performance-test
```

TIP

The Docs Markdown extension for VS Code includes a cleanup script for Microsoft links. The script checks all links to Microsoft sites in a repo to ensure that they begin with `https` instead of `http` and doesn't have locale codes, like `en-us`. To run the script:

1. Install the Docs Markdown extension for VS Code.
2. Select alt + M to open the Markdown menu.
3. Select **Cleanup**, then **Microsoft links**.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

heading-empty

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn units

Suggestion

`Empty {heading-level}. Headings must have text.`

Headings within an article are designated by hashes. Make sure each H1 (`##`) - H6 (`#######`) contains heading text.

Resolution

To fix this issue, make sure your heading includes text, not just hashes.

Bad:

```
---
author: meganbradley
ms.author: mbradley
---
##

This is an empty H2.
```

Good:

```
---
author: meganbradley
ms.author: mbradley
---
# This H2 has heading text
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

heading-no-content

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn units

Suggestion

`{heading} contains no content. Add content or remove the unused heading.`

Each heading in an article must contain content, including subheadings. For example, you can have an H2 immediately followed by an H3. You can't have an H2 followed by another H2 with no intervening content, an H3 followed by an H2 with no intervening content, or any heading without content at the end of an article.

Bad:

```
## H2

## Another H2 with no intervening content
```

```
### H3

## H2 with no intervening content
```

```
## H2
<file end>
```

OK:

```
## H2

### H3 with no intervening content
```

Resolution

Ensure that all your headings contain content. Included files, zones, and monikers might affect heading content and order.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

heading-skipped

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn units

Suggestion

```
Skipped heading. {0} is followed by {1}. For accessibility, headings must increment with no skipped levels.
```

Screen readers for users with visual impairments depend on headings being in order to properly convey the structure of a document.

Resolution

Review the current article and any included files and restructure headings so that the published article has no skipped levels. Note that zones and monikers might affect heading level.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

heading-with-underline

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Bad {heading} format. Use hashes to create H1 (#) or H2 (##)`

Markdown supports creating H1s and H2s with underlining, like this:

```
This is an H1
=============

This is an H2
-------------
```

These headings aren't allowed on Docs because they can be created accidentally and can't be validated for all potential issues.

Resolution

Replace headings created with underlines with standard hash headings:

```
# This is an H1

## This is an H2
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

html-link

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

This validation is not enabled on any production ruleset at this time, because HTML links aren't recommended but aren't prohibited. If you'd like to run this rule on your repo to find HTML links, contact mbradley.

Suggestion

`HTML link '{0}' is not allowed. Use a Markdown link.`

To ensure that all links in Docs content are accessible and secure, you must use valid Markdown links instead of HTML links.

Resolution

Use a valid Markdown link. The following are examples:

```
Relative link to another article on Docs:

[hard-coded-locale](hard-coded-locale.md)

Bookmark link to a heading in another article:

[hard-coded-local Resolution](hard-coded-locale.md#resolution)

URL to a website:

[microsoft.com](https://www.microsoft.com)
```

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

image-invalid

3/10/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Error

`Image reference '{image path}' is invalid per the schema. Unexpected attribute: '(attribute}'.`

The schema for the Docs `:::image:::` Markdown extension is as follows:

ELEMENT	DESCRIPTION	EXAMPLES
`:::image:::`	Identifies an image. Required.	`:::image type="content" source="../media/git-and-github/fork.png" alt-text="Illustrates the process flow for forking a repo." loc-scope="other":::`
image content	The content of the `:::image:::`, used for long descriptions of complex images. Required if `type="complex"`, otherwise not allowed. Should contain plain text only - no formatting.	`:::image type="complex" source="../media/git-and-github/fork.png" alt-text="Illustrates the process flow for forking a repo." loc-scope="other":::` `Long description of complex image goes here.` `:::image-end:::`
`:::image-end:::`	Identifies the end of an image. Required only if `type="complex"`.	`:::image ... :::` `...` `:::image-end:::`
`alt-text`	Provides alternative text for screen readers. Required if `type="content"` or `type="complex"`. Not allowed if `type="icon"`. Should contain plain text only - no formatting. For detailed alt text guidelines, see Accessibility guidelines for multimedia.	`alt-text="Illustrates the process flow for forking a repo."`
`border`	Specifies whether to automatically add a gray border to images. Default is `true`. To override the default, add `border="false"`.	`:::image type="icon" source="../media/git-and-github/fork.png" border="false":::`
`light-box`	Specifies an expandable image. Optional. The value is the path to the expanded image.	`:::imagetype="content"source="../media/inline-image.png"alt-text="howtoeditanarticleonDocs"lightbox="../media/e image.png":::`
`loc-scope`	Specifies the localization scope for the image, which may be different than for the parent article or module. The `loc-scope` attribute is optional per the image schema.	`loc-scope="vs-code"`
`source`	The path to the image. Required. The file name extension must be a supported image type: .png, .svg, .jpg, .jpeg, .gif	`../media/git-and-github/fork.png`

ELEMENT	DESCRIPTION	EXAMPLES
`type`	The type of image. Required. Valid values are `content` (default), `complex` (for images requiring a long description), and `icon` (for purely decorative images, such as product logos).	`type="content"`

Resolution

Review the schema requirements above and update your `:::image:::` to comply with them.

> **TIP**
>
> To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

include-not-found

3/5/2021 • 2 minutes to read

Error

```
Included file '{file-path}' can't be found in current repository.
```

Resolution

Verify that the included file exists in the repo and that the INCLUDE path is correct.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

insecure-link

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`Link '{URL}' is insecure. Links to Microsoft sites must use 'https' instead of 'http'.`

Links to Microsoft web sites, like `microsoft.com`, must be secure.

This message means you either used an explicit Markdown link with `http` or you used a raw URL, like `www.microsoft.com`. Raw URLs are converted to insecure links when published to Docs, and shouldn't be used. URLs that use `http` but aren't meant to be clickable should be styled as inline code rather than live links.

Resolution

Change clickable links to Microsoft sites to use `https` instead of `http`.

If your link is a raw URL, change it to an explicit Markdown link beginning with `https`:

```
[www.microsoft.com](https://www.microsoft.com)
```

For web addresses that aren't meant to be clickable, like fully qualified domain names and namespace references, use inline code:

```
`www.microsoft.com:90`
```

TIP

The Docs Markdown extension for VS Code includes a cleanup script for Microsoft links. The script checks all links to Microsoft sites in a repo to ensure that they begin with `https` instead of `http` and don't contain locale codes, like `en-us`. To run the script, first style any links that shouldn't be clickable as inline code, then follow these steps:

1. Install the Docs Markdown extension for VS Code.
2. Select alt + M to open the Markdown menu.
3. Select **Cleanup**, then **Microsoft links**.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

invalid-value

4/28/2021 • 2 minutes to read

Applies to: YAML files

Error

```
Invalid value for '{YAML field}': '{value}'.
```

The YAML field contains a value that's invalid per the relevant taxonomy.

Resolution

Choose a value from the appropriate taxonomy.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

languages-missing

4/28/2021 • 2 minutes to read

Applies to: Learning paths, Learn modules

Suggestion

`Missing required attribute: 'languages'. Add at least one language from the Dev Lang taxonomy.`

Learning paths and Learn modules must specify at least one programming language.

Resolution

Add one or more programming languages as appropriate from the Dev Lang taxonomy to a languages array in index.yml:

```
languages:
- al-language
- brainscript
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

link-out-of-scope

3/10/2021 • 2 minutes to read

Applies to: All content types

Warning

```
File '{file-name}' referenced by link '{link-value}' will not be built because it is not included in build scope.
```

The file has been excluded from the build scope via the `exclude` array in the docfx.json configuration file.

Resolution

Resource files such as included files and readmes can be excluded from the build in docfx.json. If you try to reference an excluded file by a file link or image reference, you'll get this Warning.

To fix the issue, first find the `content` group in docfx.json that includes the out-of-scope file. For example, the `articles` folder of azure-docs-pr:

```
"content": [
  {
    "src": "articles",
    "dest": ".",
    "files": [
      "**/*.md",
      "**/*.yml"
    ],
    "exclude": [
      "iot-edge/**",
      "**/_themes/**",
      "**/includes/**",
      "**/obj/**"
    ]
  }
```

Review the `exclude` array. You should find the path of out-of-scope file there. In this example, if you try to link to anything under `iot-edge/**` you'll get this Warning.

Then either update docfx.json so the file isn't excluded, or move the file to a path that isn't excluded. If you're not sure what to do, ask your repo owner for help.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

link-redirected

3/5/2021 • 2 minutes to read

Applies to: All content types

Suggestion

```
Redirected link '{0}' will be broken in isolated environments. Replace with a relative link or direct URL.
```

Link redirection services such as aka.ms and fwlink aren't recommended. These links will be broken in isolated (offline) environments.

Resolution

For links to other pages on docs.microsoft.com, use a direct relative link. These links will still work in isolated environments.

For links to other web sites, use a direct URL. Although these will still be broken in isolated environments, users will be able to see the URL and can view the content when outside the isolated environment.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

link-text-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Missing link text for link '{link path}'. Add appropriate text.`

Link text is required for links to be visible and clickable, and for screen readers to indicate the link target.

Resolution

Add descriptive link text, such as the title of the target page, to your link.

```
Bad:
[](../contribute-accessibility-multimedia.md)

Good:
[Accessibility guidelines for multimedia](../contribute-accessibility-multimedia.md)
```

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

invalid-tab-group

11/2/2020 • 2 minutes to read

Warning

`Tab group with different tab id set.`

You can only have one tab group per article. Tab groups are defined implicitly based on the tabs used in your article. For example, suppose you define the following three tabs by using them in your article:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

# [PowerShell](#tab/azure-powershell)

PowerShell content...
```

This creates a tab group, which you must use throughout the article. That is, if you use the C# tab again, you must also use Python and PowerShell, and you can't add additional tabs unless you add them to the initial grouping. You also must maintain the tab order in subsequent usage.

Good:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

# [PowerShell](#tab/azure-powershell)

PowerShell content...

# [C#](#tab/csharp)

More C# content...

# [Python](#tab/python)

More Python content...

# [PowerShell](#tab/azure-powershell)

More PowerShell content...
```

Bad:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

# [PowerShell](#tab/azure-powershell)

PowerShell content...

# [C#](#tab/csharp)

More C# content, but no Python tab!

# [PowerShell](#tab/azure-powershell)

More PowerShell content...
```

Bad:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

# [PowerShell](#tab/azure-powershell)

PowerShell content...

# [C#](#tab/csharp)

More C# content...

# [Python](#tab/python)

More Python content...

# [PowerShell](#tab/azure-powershell)

More PowerShell content...

# [JavaScript](#tab/javascript)

JavaScript tab wasn't defined in initial tab group!
```

Bad:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

# [PowerShell](#tab/azure-powershell)

PowerShell content...

# [C#](#tab/csharp)

More C# content...

# [PowerShell](#tab/azure-powershell)

PowerShell content out of order!

# [Python](#tab/python)

More Python content...
```

Resolution

Make sure all required tabs are defined in your initial tab group, and use the same tabs in the same order in all subsequent groups.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

manager-invalid

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

This build issue is a custom validation that isn't implemented on most repos. If you'd like this rule run on your repo, contact mbradley.

Suggestion

`Invalid value for manager: '{value}' is not a valid Microsoft alias.`

Some content groups require the `manager` attribute to identify the author's manager.

Resolution

The value of `manager` must be a valid alias for an individual Microsoft employee. Verify the author's manager's alias and update the value.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

manager-missing

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

This build issue is a custom validation that isn't implemented on most repos. If you'd like this rule run on your repo, contact mbradley.

Suggestion

`Missing attribute: manager. Add a valid Microsoft alias for the author's manager.`

Some content groups require the `manager` attribute to identify the author's manager.

Resolution

Add a valid Microsoft alias for `manager`:

```
---
ms.author: mbradley
manager: jemash
---
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

merge-conflict

7/7/2021 • 2 minutes to read

Applies to: All content types

Warning

`File contains merge conflict markers.`

Git merge conflict markers remain in your article, like this:

```
<<<<<<< HEAD
...
=======
...
>>>>>>> 1d82c7efe18f86136247fb366df5030843199c19
```

You get this message when you mark merge conflicts resolved for a pull request without deleting all the merge conflict markers. Merge conflict markers add meaningless content to the published page.

Resolution

Follow the steps in Resolve the merge conflict article to fix the conflict.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-achievement-not-badge

3/5/2021 • 2 minutes to read

Warning

```
Achievement UID '{uid}' doesn't refer to a badge. Modules must have badge achievements.
```

Resolution

Replace the specified achievement with a valid badge UID.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-achievement-not-found

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

```
Achievement UID '{uid}' is invalid. Specify a valid badge UID.
```

Resolution

Make sure the achievement specified for your module is a badge with a valid UID.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-badge-missing

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

`Missing attribute: badge. A valid badge is required for each module.`

Resolution

Add a valid badge UID, as follows:

```
badge:
  uid: <module-uid>.badge
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-child-not-found

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

`Child UID(s): '{uid-list}' can't be found.`

One or more of the UIDs specified in the `units` property of the module is invalid.

Resolution

Verify that all UIDs specified in the `units` property are valid unit UIDs. Here's an example:

```
units:
- learn.align-requirements-in-azure.introduction
- learn.align-requirements-in-azure.public-private-hybrid
- learn.align-requirements-in-azure.service-models
- learn.align-requirements-in-azure.iaas
- learn.align-requirements-in-azure.paas
- learn.align-requirements-in-azure.saas
- learn.align-requirements-in-azure.quiz
- learn.align-requirements-in-azure.summary
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-child-not-unit

3/5/2021 • 2 minutes to read

Error

`Invalid child UID(s): '{uid-list}'. A module can only have units as children.`

Resolution

Verify that all UIDs specified in the `units` property of the module are valid unit UIDs. Here's an example:

```
units:
- learn.align-requirements-in-azure.introduction
- learn.align-requirements-in-azure.public-private-hybrid
- learn.align-requirements-in-azure.service-models
- learn.align-requirements-in-azure.iaas
- learn.align-requirements-in-azure.paas
- learn.align-requirements-in-azure.saas
- learn.align-requirements-in-azure.quiz
- learn.align-requirements-in-azure.summary
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-uid-missing

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

`Missing attribute: uid. A valid Uid is required for each module.`

Resolution

Add a unique module UID in this format:

```
<prefix>.<module-folder-name>

Such as:
learn.align-requirements-in-azure
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-units-missing

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

`Missing attribute: units. At least one valid unit is required for each module.`

Resolution

Add one or more unit UIDs to the `units` property of the module. Here's an example:

```
units:
- learn.align-requirements-in-azure.introduction
- learn.align-requirements-in-azure.public-private-hybrid
- learn.align-requirements-in-azure.service-models
- learn.align-requirements-in-azure.iaas
- learn.align-requirements-in-azure.paas
- learn.align-requirements-in-azure.saas
- learn.align-requirements-in-azure.quiz
- learn.align-requirements-in-azure.summary
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-author-missing

7/7/2021 • 2 minutes to read

This article applies to Conceptual (Markdown), .NET, REST, and PowerShell content.

Warning

```
Missing attribute: ms.author. Add the current author's Microsoft alias.
```

Resolution

Add the current author's Microsoft alias for the `ms.author` metadata attribute. The value should be the *current* owner of the article, not the original author if ownership has changed. We recommend that the author is a full-time employee or team distribution list (DL), rather than a short-term vendor.

If the alias is a DL, it must also be on the `ms.author` allowlist.

Valid values for `ms.author` DLs can be found on the ms.author allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-author-invalid

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn, .NET, REST, PowerShell

Suggestion

```
Invalid value for ms.author: '{value}' is not a valid Microsoft alias, or is not an allowed distribution
list.
```

Resolution

Update the `ms.author` value with the current author's valid Microsoft alias. We recommend that the author is a full-time employee or team distribution list (DL), rather than a short-term vendor.

If the alias is a DL, it must also be on the `ms.author` allowlist.

See the ms.author allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-date-invalid

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

For Docs content:

`Invalid value for ms.date: '{value}'. Must be a date in format MM/DD/YYYY, no more than 30 days from today.`

For Learn content:

`Invalid value for ms.date: '{value}'. Must be a date in format MM/DD/YYYY, no later than today.`

Invalid `ms.date` values provide incorrect information about how up to date an article is, and can break some site and reporting functionality. For Docs content, dates can be up to 30 days in the future to allow predating for large releases such as Build and Ignite. This isn't currently allowed for Learn because a technical limitation related to the Learn Catalog API can cause partner tools to break if the `ms.date` is in the future.

Resolution

Confirm that the article is up to date with no broken content, then add a valid date in the format MM/DD/YYYY:

```
---
ms.date: 02/19/2019
---
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-date-missing

7/9/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Missing attribute: ms.date. A freshness date is required for this content. Add a date in format MM/DD/YYYY.`

Some content groups require `ms.date` to indicate "freshness" - that is, when the article was last reviewed for relevance, accuracy, correct screenshots, and working links. Freshness isn't the same as the last date the article was *published*, which will show on the page if `ms.date` isn't explicitly specified.

Resolution

Confirm that the article is up to date with no broken content, then add a valid date in the format MM/DD/YYYY:

```
---
ms.date: 02/19/2019
---
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-prod-and-service

7/9/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

```
Both ms.prod and ms.service can't be specified. Use ms.prod for on-premise products, or ms.service for cloud services.
```

Resolution

Either `ms.prod` or `ms.service` is required, and they can't both be present: use `ms.prod` for on-premises products; use `ms.service` for cloud services. Determine which is appropriate for your article, check that the value is correct, and remove the other field.

For valid `ms.prod` and `ms.service` values, see Taxonomy lists. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-prod-missing

7/9/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Missing attribute: ms.prod. If you specify ms.technology, you must also specify ms.prod.`

Use `ms.prod` to specify on-premises products. To specify more detailed information about `ms.prod`, you can optionally specify `ms.technology`, but if you specify `ms.technology`, you must also specify `ms.prod`. The values for `ms.prod` and `ms.technology` must be a valid pair.

Resolution

Confirm that the `ms.technology` value you've specified is correct for your article. Then add the appropriate `ms.prod` value that is a valid parent for the `ms.technology`.

For valid `ms.prod` and `ms.technology` pairs, see the ms.prod allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-prod-or-service-missing

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

```
Missing attribute: either ms.prod or ms.service is required. Use ms.prod for on-premise products, or
ms.service for cloud services.
```

Resolution

Either `ms.prod` or `ms.service` is required, and they can't both be present: `ms.prod` is used for on-premises products; `ms.service` is used for cloud services. Determine which is appropriate for your article, check that the value is correct, and remove the other field.

For valid `ms.prod` and `ms.service` values, see the ms.prod allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-prod-technology-invalid

7/9/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Invalid value for ms.prod: '{value}'.`

`Invalid value for ms.technology: '{value}' is not valid with ms.prod value '{value}'.`

Use `ms.prod` to specify on-premises products. To specify more detailed information about `ms.prod`, you can optionally specify `ms.technology`. The values for `ms.prod` and `ms.technology` must be a valid pair. Either your `ms.prod` value is invalid, or your `ms.technology` value isn't a valid pair with your `ms.prod`.

Resolution

Confirm that your `ms.prod` value is correct for your article. Then choose a valid `ms.technology` value.

For valid `ms.prod` and `ms.technology` pairs, see the ms.prod allowlist. To request new values, see How to request metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-service-missing

7/9/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Missing attribute: ms.service. If you specify ms.subservice, you must also specify ms.service.`

Use `ms.service` to specify cloud services. To specify more detailed information about `ms.service`, you can optionally specify `ms.subservice`, but if you specify `ms.subservice`, you must also specify `ms.service`. The values for `ms.service` and `ms.subservice` must be a valid pair.

Resolution

Confirm that the `ms.subservice` value you've specified is correct for your article. Then add the appropriate `ms.service` value that is a valid parent for the `ms.subservice`.

For valid `ms.service` and `ms.subservice` pairs, see the ms.service allowlist. To request new values, see How to request metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-service-subservice-invalid

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Invalid value for ms.service: '{value}'.`

`Invalid value for ms.subservice: '{value}' is not valid with ms.service value '{value}'.`

Use `ms.service` to specify cloud services. To specify more detailed information about `ms.service`, you can optionally specify `ms.subservice`. The values for `ms.service` and `ms.subservice` must be a valid pair. Either your `ms.service` value is invalid, or your `ms.subservice` value isn't a valid pair with your `ms.service`.

Resolution

Confirm that your `ms.service` value is correct for your article. Then choose a valid `ms.subservice` value.

For valid `ms.service` and `ms.subservice` pairs, see the ms.service allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-topic-invalid

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`Invalid value for ms.topic: '{value}'.`

Resolution

Add a valid value for `ms.topic` that indicates the article type.

For valid `ms.topic` values, see the ms.topic allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-topic-missing

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

```
Missing attribute: ms.topic. Add a valid value to indicate the article type.
```

Resolution

Add a valid value for `ms.topic` that indicates the article type.

For valid `ms.topic` values, see the ms.topic allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

multiple-h1s

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Multiple H1s are not allowed. You can only have one top-level heading.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font. An H1 is created by beginning a line with a single hash (#) followed by a space, then the heading text. You can only have one H1 in each file.

You might also get this message if your article has a line of equals signs making a double underline, like this: `=======`. A double underline is an alternative Markdown syntax for an H1. It's also commonly seen in merge conflicts:

```
<<<<<<< HEAD
...
=======
...
>>>>>>> 1d82c7efe18f86136247fb366df5030843199c19
```

Resolution

To fix this issue, change the heading level of later H1s to H2 (`##`), or otherwise reorganize your file.

```
---
author: meganbradley
ms.author: mbradley
---
# This is an H1

Some content...

## This is an H2
```

If an extra H1 is actually a double underline (`=======`), remove it or replace it with a hashtag heading, like `##`, as appropriate. If the double underline is part of a merge conflict, make sure to also remove the merge conflict beginning and ending markers and the obsolete text.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

multiple-values

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Single-valued attribute '{attribute}' has multiple values. Remove additional values.`

You specified more than one value for an attribute that is only allowed one value.

`Single-valued attribute '{attribute}' is in multi-valued array format. Change to scalar format.`

Attributes that aren't allowed to have more than one value must be specified in the single-valued (scalar) YAML format.

Attributes in scope

The following attributes are required to be single-valued:

- `author`
- `ms.author`
- `ms.date`
- `ms.prod`
- `ms.service`
- `ms.subservice`
- `ms.technology`
- `ms.topic`
- `title`

Resolution

You get this Suggestion anytime you use a multi-valued array for a single-valued attribute, either with multiple values or a single value in the array.

YAML supports the following array formats for multi-valued attributes, such as `ms.custom`:

```
---
# comma-separated bracketed list:
ms.custom: [WIP, generated-via-CI]

# each value on its own line:
ms.custom:
  - WIP
  - generated-via-CI
---
```

These formats aren't valid for single-valued attributes, such as `author`.

If you specified multiple values, remove incorrect values and specify the single value on the same line as the attribute with no brackets:

```
---
author: meganbradley
---
```

If you have a single-valued array, change it to the scalar format above.

> **TIP**
>
> The Docs Markdown extension for VS Code includes a cleanup script for single-valued metadata in array format. The script checks all YAML metadata headers in the repo and converts badly formatted single-valued metadata to scalar format. To run the script, follow these steps:
>
> 1. Install the Docs Markdown extension for VS Code.
> 2. Click alt + M to open the Markdown menu.
> 3. Select **Cleanup**, then **Single-valued metadata**.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

notebook-dependent-schema-failed

6/16/2021 • 2 minutes to read

Applies to: Learn units

Error

```
DependentSchemas validation failure for attribute: 'notebook'. Make sure 'sandbox' is present and set to
'true'. The 'tasks' attribute must not be present when notebook is present.
```

The `notebook` attribute can only be specified for a Learn module unit if the `sandbox` attribute is also present and set to `true`. Only one of `notebook` or `tasks` can be specified for a unit.

Resolution

To specify a Jupyter notebook, make sure `sandbox` is set to `true` and `tasks` is not present in the unit YAML file.

```
### YamlMime:ModuleUnit
uid: learn-test-sandbox.test-orphan2-module.unit-1
title: Unit with Jupyter notebook
metadata:
  title: Unit with Jupyter notebook
  description: Specify a Jupyter notebook in a Learn modulde unit
durationInMinutes: 10
sandbox: true
notebook: http://raw.githubusercontent.com/test.ipybk
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-achievement-not-found

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

```
Achievement UID '{uid}' is invalid. Specify a valid trophy UID.
```

Resolution

Make sure the achievement specified for your learning path is a trophy with a valid UID.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-achievement-not-trophy

3/5/2021 • 2 minutes to read

Error

```
Achievement UID '{uid}' doesn't refer to a trophy. Learning paths must have trophy achievements.
```

Resolution

Replace the specified achievement with a valid trophy UID.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-child-not-found

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

`Child UID(s): '{uid-list}' can't be found.`

One or more of the UIDs specified in the `modules` property of the learning path is invalid.

Resolution

Verify that all UIDs specified in the `modules` property are valid module UIDs. Here's an example:

```
modules:
- learn.pillars-of-a-great-azure-architecture
- learn.design-for-security
- learn.design-for-performance-and-scalability
- learn.design-for-efficiency-and-operations
- learn.design-for-availability-and-recoverability
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-child-not-module

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

`Invalid child UID(s): '{uid-list}'. A learning path can only have modules as children.`

Resolution

Verify that all UIDs specified in the `modules` property for the learning path are valid module UIDs. Here's an example:

```
modules:
- learn.pillars-of-a-great-azure-architecture
- learn.design-for-security
- learn.design-for-performance-and-scalability
- learn.design-for-efficiency-and-operations
- learn.design-for-availability-and-recoverability
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-duplication

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

This validation checks that published URL paths follow best practices for SEO. Because existing paths can't be easily changed and would result in many unactionable results in some build reports, this validation hasn't been enabled on a production ruleset for Docs at this time. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Article URL '{URL}' contains duplicated text '{segment-text}'. For SEO, article URLs should not contain duplication.
```

Duplicate path segments, such as `https://docs.microsoft.com/.../admin/get-help-with-domains/get-help-with-domains`, are bad for SEO and a bad user experience.

Resolution

Update your file paths so that segments aren't duplicated. If your content is already live, you'll have to redirect the old paths.

For more information, see File name and path guidelines.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-modules-missing

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

```
Missing attribute: modules. At least one valid module is required for each learning path.
```

Resolution

Add one or more module UID to the `modules` property of the learning path. Here's an example:

```
modules:
- learn.pillars-of-a-great-azure-architecture
- learn.design-for-security
- learn.design-for-performance-and-scalability
- learn.design-for-efficiency-and-operations
- learn.design-for-availability-and-recoverability
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-too-deep

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Article URL '{URL}' is '{level}' levels deep. For SEO, URLs should be 3 levels or fewer.
```

Deep file paths are bad for SEO.

Resolution

Update your file paths so that they're no more than 3 levels deep.

For more information, see File name and path guidelines.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-trophy-missing

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

```
Missing attribute: trophy. A valid trophy is required for each learning path.
```

Resolution

Add a valid trophy UID, as follows:

```
trophy:
  uid: <path-uid>.trophy
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-uid-missing

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

```
Missing attribute: uid. A valid uid is required for each learning path.
```

Resolution

Add a unique learning path UID in this format:

```
<prefix>.<learning-path-folder-name>

Such as:
learn.architect-great-solutions-in-azure
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-definition-conflict

3/5/2021 • 2 minutes to read

Warning

`Multiple source files are present for '{definition-file}'.`

This means that more than one zone pivot definition file is present in the repo that will be published to the same output path. This usually happens in versioned content when more than one `monikerRange` includes a zone pivot definition file.

Resolution

First, find the conflict. For example, suppose you have two versions of content, each stored in their own directory and associated with a different `monikerRange`. If each directory contains a definition file at the same location with the same name, it will be published to the same output path. Docs uses the output path to resolve zone pivots, so even if the configuration seems correct for each version, the result will be a conflict.

Once you've identified the conflict, you fix it in one of these ways:

- Use a single definition file for all versions. If the versions can all use the same pivot groups, you can use one file to apply to all versions. By default this should be zone-pivot-groups.yml at the docset root.
- Rename and/or move versioned definition files so each outputs to a unique path. Use `zone_pivot_group_filename` metadata to point to the correct output path for each version that doesn't use the default.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-definition-not-found

3/5/2021 • 2 minutes to read

Warning

```
No source file is present for '{definition-file}'. To use zone pivots, you must first define the zone pivot
groups in zone-pivot-groups.yml or specify a different zone pivot definition file in the
'zone_pivot_group_filename' metadata. Cross docset references to zone pivot groups definition files aren't
supported by Docs validation.
```

The system can't find the expected zone pivot definition source file. This is either zone-pivot-groups.yml at the docset root (the default) or a custom definition file specified in the `zone_pivot_group_filename` metadata.

Resolution

First you must determine what zone pivot definition file is expected for your article. This can be tricky because it depends on how your repo is configured. By default, the expected definition file is zone-pivot-groups.yml at the root of your docset. For example, in the azure-docs-pr repo, the `articles` folder is the root of the docset, meaning that everything under `articles` is published to `https://docs.microsoft.com/en-us/azure`. Therefore, by default the Docs system expects a source file called zone-pivot-groups.yml as a direct child of `articles` in the azure-docs-pr repo. In sql-docs-pr, the root is the `docs` folder, which publishes to `https://docs.microsoft.com/en-us/sql`. Therefore the default definition file for sql-docs-pr is zone-pivot-groups.yml as a direct child of `docs`. Each repo is different, so if you don't know which folder is your docset root you can ask for help as described in "Need help?" below.

Alternatively, you can specify a different definition file in the `zone_pivot_group_filename` metadata. You can save this definition file anywhere in the docset with any name, and reference it by its relative path to the root in the `zone_pivot_group_filename` metadata. For example, if you add a custom file called data-explorer-zone-pivot-groups.yml to the `articles/data-explorer` folder in azure-docs-pr, you'd add metadata as follows:

```
zone_pivot_group_filename: data-explorer/data-explorer-zone-pivot-groups.json
```

IMPORTANT

The extension is .json, not .yml, because you're technically referencing the published definition file, as opposed to the .yml source file.

To resolve this Warning, make sure the expected file is at the expected path, or update the `zone_pivot_group_filename` metadata to the correct path.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-group-missing

3/5/2021 • 2 minutes to read

Warning

```
Missing metadata attribute: 'zone_pivot_groups'. To use zone pivots in your file, you must specify the valid
pivot group that contains the pivot IDs you want to use.
```

The `zone_pivot_groups` metadata field specifies which zone pivot groups defined in a zone pivot definition file can be used in an article or included Markdown file. This metadata can be assigned:

- At the global or folder level in docfx.json.
- In the YAML header of a Docs Markdown article.
- In the `metadata` section of a Learn YAML file.

Each value of `zone_pivot_groups` must be a pivot group ID defined in the zone pivot definition file, such as zone-pivot-groups.yml.

Resolution

First, ensure that the pivot group you want to use is defined in zone-pivot-groups.yml (or in the custom zone pivot definition file if your docset doesn't use zone-pivot-groups.yml). Then add the pivot group ID as the value of `zone_pivot_groups`. If you're using a definition file other than zone-pivot-groups.yml, make sure the `zone_pivot_group_filename` metadata is also set to the path of the definition file.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-group-not-found

3/5/2021 • 2 minutes to read

Warning

```
Pivot group '{metadata-value}' isn't defined in '{definition-file}'. Make sure every pivot group you
reference in your content has been defined in your zone pivot definition file.
```

A pivot group ID specified in the `zone-pivot-groups` metadata isn't defined in the zone pivot definition file for your docset, usually zone-pivot-groups.yml.

Resolution

Check your zone pivot definition file. If your group is defined, update the `zone-pivot-groups` metadata with the correct pivot group ID. If the group isn't defined, add a new group as follows:

```
- id: pivot-group-id
  title: Pivot group label
  prompt: Prompt text for the group
  pivots:
    - id: pivot-1
      title: Pivot 1 label
    - id: pivot-2
      title: Pivot 2 label
    - id: pivot-3
      title: Pivot 3 label
```

Then add the `pivot-group-id` value as the value of `zone-pivot-groups` for your article or unit.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-id-not-found

3/5/2021 • 2 minutes to read

Warning

```
Pivot ID '{pivot-id}' is not defined in '{group-list}' in '{definition-file}'. You can only use pivots that
have been defined in the referenced pivot group.
```

You've specified one or more pivot group in the `zone-pivot-groups` metadata, but you've used a pivot ID that isn't included in any of the groups.

Resolution

Either add the desired pivot ID to the appropriate group, or use a different pivot ID. If you add a pivot ID to a group, make sure to add that pivot ID to every other article or unit that uses the group.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-id-unused

3/5/2021 • 2 minutes to read

Warning

```
Pivot ID '{id}' is defined in the '{pivot-group}' group in '{definition-file}', but not used. You must use
all pivot IDs in a pivot group to avoid blank tabs.
```

Pivot groups define a standard set of tabs. If you use a group in a file but don't use all pivots in the group, content for the pivot will be missing.

Resolution

Add the missing pivot. If it's not appropriate to use all the pivots defined in the group, use a different group or don't use zone pivots in this file. In the case that the pivot group is appropriate but some content is supplemental for one or more pivots, you can add an empty zone. For example, suppose you have an article that provides variations of instructions for Windows versus Linux. You might have a section where the information is the same except you need to add an extra note for Linux. To indicate that the note is intentionally left off for Windows, you can add an empty zone for Windows with a comment, such as:

```
:::zone pivot="windows":::
<!-- No corresponding content for Windws. -->
:::zone-end:::
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

preserve-view-not-set

4/8/2021 • 2 minutes to read

Applies to: All content types

Suggestion

`Did you mean to link to a specific version? If so, add &preserve-view=true to the URL: {URL}`

To specify a specific version for a link, you must specify both the `view` parameter, which specifies the version, and `&preserve-view=true`, which tells the system to always preserve the `view` value. Without `preserve-view` set to `true`, Docs can't tell the difference between a hard-coded `view` and a `view` that was autogenerated based on the user's selections.

For example, a content developer might add a link to the URL `https://docs.microsoft.com/sql/t-sql/functions/cast-and-convert-transact-sql?view=sql-server-ver15`, assuming that the link will always go to the SQL Server 2019 (v15) version of the target page. However, a user might get to this same URL by selecting SQL Server 2019 from the site version picker, and Docs can't tell the difference. So if you don't also add `&preserve-view=true` to your link, Docs will determine the view based on version defaults and user selections, potentially overriding the specified version.

This rule has been enabled for PR only.

Resolution

To ensure that your specified view is always preserved, add `&preserve-view=true` to the URL, such as

```
https://docs.microsoft.com/sql/t-sql/functions/cast-and-convert-transact-sql?view=sql-server-ver15&preserve-view=true
```

.

If it's not essential to link to a specific version of the target page, remove the `view` parameter from your link, such as `https://docs.microsoft.com/sql/t-sql/functions/cast-and-convert-transact-sql`.

For more information about versioned links, see Links to specific versions.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

quiz-multiple-answers

3/5/2021 • 2 minutes to read

Error

`Question must have one correct answer.`

Each question in a quiz can have only one answer.

Resolution

Review the possible answers for the question and indicate which one is correct via the `isCorrect` YAML key. Only one can be correct. The following is an example:

```
quiz:
    title: Logging in to the Azure CLI 2.0
    questions:
    - content: "Were you able to log in to the Azure CLI 2.0?"
      choices:
      - content: "Yes"
        isCorrect: true
        explanation: "Great! Proceed to the next step in the tutorial."
      - content: "No"
        isCorrect: false
        explanation: "Try to log in to your Azure account at [https://portal.azure.com]. If you can't, try resetting your password, then logging in again."
```

For more information, see Add a knowledge check in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

quiz-no-answer

3/5/2021 • 2 minutes to read

Warning

`Question must have one correct answer.`

Each question in a quiz must have one correct answer.

Resolution

Review the possible answers for the question and indicate which one is correct via the `isCorrect` YAML key. The following is an example:

```
quiz:
    title: Logging in to the Azure CLI 2.0
    questions:
    - content: "Were you able to log in to the Azure CLI 2.0?"
      choices:
      - content: "Yes"
        isCorrect: true
        explanation: "Great! Proceed to the next step in the tutorial."
      - content: "No"
        isCorrect: false
        explanation: "Try to log in to your Azure account at [https://portal.azure.com]. If you can't, try
resetting your password, then logging in again."
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

redirect-url-invalid

11/2/2020 • 2 minutes to read

Applies to: All content types

Warning

```
Can't preserve document ID for redirected file '{file-name}' because redirect_url '{URL}' is invalid or is is
in a different docset. Specify a redirect_url in the same docset, or set redirect_document_id to false in
.openpublishing.redirection.json.
```

This message means that either the `redirect_url` is an invalid link, or you're trying to redirect the document ID of the source file to a file in another docset. Document IDs can only be redirected within the current docset.

Resolution

First, determine which case is causing this Warning.

If your `redirect_url` is invalid, replace it with a valid relative path or absolute URL. If the `redirect_url` is a relative path to a file in the same docset, and no other redirection to that path specifies `redirect_document_id` as `true`, you can preserve the document ID.

If your `redirect_url` is a valid link to an article in a different docset, set `redirect_document_id` to `false`.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

redirected-file-not-removed

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

```
Redirected file '{file-name}' is still in the repo. After adding a file to the redirection JSON file, you
must delete the original file from the repo.
```

Resolution

Delete the redirected file. For more information about redirection, see How to redirect obsolete articles.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

rule-override-invalid

11/2/2020 • 2 minutes to read

Applies to: docfx.json

Warning

```
Rule '{error-code}' can't be overridden in docfx.json."
```

Some validation rules can be overridden for a file or set of files via the docfx.json config file, but most can't. This message means docfx.json contains override syntax for a rule that isn't overridable. The rule will still run on the excluded file(s).

Resolution

Remove the override syntax from docfx.json. If you believe you have a legitimate exception, file a bug via SiteHelp and the Docs team will evaluate your request.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

sandbox-not-mapped

3/5/2021 • 2 minutes to read

Applies to: Learn units

Error

```
Module '{parent-module}' is not mapped to the Azure sandbox. To use the sandbox, you must first submit an
onboarding request.
```

You've added `azureSandbox` metadata to a unit, but haven't registered its parent module with Azure Sandbox.

Resolution

To use the Azure Sandbox in an exercise unit, you must first register the parent module in the Sandbox database by filling out an onboarding form. Remove the `azureSandbox` metadata until mapping is confirmed. For more information, see Request an Azure Sandbox for a module in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

sensitive-language

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

```
Term '{term}' is sensitive and should not be used in content or code, or should be used with caution
depending on context.
```

Resolution

For a list of problematic terms and how to handle them, see the Terminology and inclusive language article.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

Tabbed conceptual

3/22/2021 • 2 minutes to read

Tab syntax

The syntax for tabs is as follows:

Single level tab:

```
# [Tab Display Name](#tab/tab-id)
```

Optional dependent tab:

```
# [Tab Display Name](#tab/tab-id/tab-condition)
```

Example of a single-level tab section with two tabs and the tab group terminator (---):

```
# [Linux](#tab/linux)

Content for Linux...

# [Windows](#tab/windows)

Content for Windows...

---
```

Tabs can optionally contain secondary tabs, or dependency tabs. This makes tabs dependent on the selection in another set of tabs. Here's an example:

```
# [Azure CLI](#tab/azure-cli/linux)

Azure CLI content for Linux...

# [Azure CLI](#tab/azure-cli/windows)

Azure CLI content for Windows...

# [PowerShell](#tab/azure-powershell/linux)

PowerShell content for Linux...

# [PowerShell](#tab/azure-powershell/windows)

PowerShell content for Windows...

---
```

The following validations apply to tab syntax:

- Tab syntax must be correct.
- Dependent tabs must have been defined in a previous tab group.
- Only one level of dependency is allowed.
- No fewer than two tabs are allowed.
- No more than four tabs are allowed.

- Tabs must be approved.
- Tab/ID pairs must be valid.
- Cannot have the same tab ID multiple times in one tab group.

Approved tabs

The following tab name/tab ID pairs are approved. Tabs names and IDs aren't validated, so you can currently use any tabs you want.

TAB NAME	TAB ID
`[.NET]`	`(#tab/dotnet)`
`[.NET Core 1.x]`	`(#tab/netcore1x)`
`[.NET Core 2.x]`	`(#tab/netcore2x)`
`[.NET Core 2.0]`	`(#tab/netcore20)`
`[.NET Core 2.1]`	`(#tab/netcore21)`
`[.NET Core 2.2]`	`(#tab/netcore22)`
`[.NET Core 3.x]`	`(#tab/netcore3x)`
`[.NET Core 3.0]`	`(#tab/netcore30)`
`[.NET Core CLI]`	`(#tab/netcore-cli)`
`[Azure CLI]`	`(#tab/azure-cli)`
`[Android]`	`(#tab/android)`
`[Ansible]`	`(#tab/ansible)`
`[Browser]`	`(#tab/browser)`
`[C#]`	`(#tab/csharp)`
`[C# Script]`	`(#tab/csharp-script)`
`[CentOS]`	`(#tab/centos)`
`[Command Line]`	`(#tab/command-line)`
`[Debian]`	`(#tab/debian)`
`[Docker Hub]`	`(#tab/docker-hub)`
`[F#]`	`(#tab/fsharp)`

TAB NAME	TAB ID
`[Fedora]`	`(#tab/fedora)`
`[iOS]`	`(#tab/ios)`
`[Java]`	`(#tab/java)`
`[JavaScript]`	`(#tab/javascript)`
`[Linux]`	`(#tab/linux)`
`[macOS]`	`(#tab/macos)`
`[Managed Kubernetes]`	`(#tab/kubernetes-managed)`
`[Maven]`	`(#tab/maven)`
`[Mint]`	`(#tab/mint)`
`[Node.js]`	`(#tab/nodejs)`
`[npm]`	`(#tab/npm)`
`[NuGet]`	`(#tab/nuget)`
`[openSUSE]`	`(#tab/opensuse)`
`[Other]`	`(#tab/other)`
`[Oracle Linux]`	`(#tab/oracle-linux)`
`[Package Manager]`	`(#tab/package-manager)`
`[PEAR]`	`(#tab/pear)`
`[pip]`	`(#tab/pip)`
`[Portal]`	`(#tab/azure-portal)`
`[PowerShell]`	`(#tab/azure-powershell)`
`[Private Registry]`	`(#tab/private-registry)`
`[Python]`	`(#tab/python)`
`[Resource Manager Template]`	`(#tab/azure-resource-manager)`
`[RHEL]`	`(#tab/rhel)`

TAB NAME	TAB ID
`[RubyGems]`	`(#tab/rubygems)`
`[SQL Server]`	`(#tab/sql-server)`
`[SQLite]`	`(#tab/sqlite)`
`[Terraform]`	`(#tab/terraform)`
`[TypeScript]`	`(#tab/typescript)`
`[Visual Basic]`	`(#tab/vb)`
`[Visual Studio]`	`(#tab/visual-studio)`
`[Visual Studio 15.6 and earlier]`	`(#tab/vs156)`
`[Visual Studio 15.7 and later]`	`(#tab/vs157)`
`[Visual Studio Code]`	`(#tab/visual-studio-code)`
`[Visual Studio for Mac]`	`(#tab/visual-studio-mac)`
`[Ubuntu]`	`(#tab/ubuntu)`
`[Unmanaged Kubernetes]`	`(#tab/kubernetes-unmanaged)`
`[Windows]`	`(#tab/windows)`

Example articles

Here are examples of articles with tabbed conceptual formatting:

- Handle SameSite cookie changes in Chrome browser
- Logging in MSAL applications
- Tutorial: Use dynamic configuration in an ASP.NET Core app

table-syntax-invalid

5/13/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Table syntax is invalid. Ensure your table includes a header and is surrounded by empty lines.`

Tables must have headers and be surrounded by blank lines to render correctly.

```
This table isn't valid because it has no header:

|This | table has |no header|at all |

This table isn't valid because it isn't surrounded by blank lines:
|Tables |need |space... |
|-------|-----|---------|
... to breathe!
```

Resolution

Make sure your table has a header and is surrounded by blank lines:

```
|This is   |a simple   |table header|
|----------|-----------|------------|
|table     |data       |here        |
|it doesn't|actually   |have to line up nicely!|
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

task-type-invalid

3/5/2021 • 2 minutes to read

Error

```
Task has invalid Azure resource type: '{resource-type}'.
```

Resolution

Verify that the specified `resourceGroup` matches a resource group that's available for your lab.

For more information, see Add task validation to an Azure Sandbox lab in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

title-missing

7/7/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

```
Missing attribute: title. Add a title string (19 - 59 characters) to show in search engine results.
```

Resolution

Add a title string to show in search results. In general, titles should be between 19 and 59 characters, should be distinct from the H1 of the article, and should include relevant branding words. Don't include " | Microsoft Docs" in your title - it's added automatically by Docs, and is ignored if you add it explicitly. If you want to add extra branding, like " - Azure", to all titles in a content set, set the `titleSuffix` metadata value globally or for a folder.

You can preview how your title will look in Google on https://moz.com/learn/seo/title-tag.

For more information about good titles and search engine optimization (SEO), see SEO Basics.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

title-suffix-missing

11/2/2020 • 2 minutes to read

This validation checks for the `titleSuffix` metadata, which appends branding information to Docs titles. Applying `titleSuffix` globally or at the folder level is recommended, but because manually added branding is still valid, this validation hasn't been enabled on a production ruleset for Docs at this time. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Missing attribute: titleSuffix. Add a titleSuffix value to append branding to your title. Using titleSuffix
is the recommended way to append a branding string to titles for SEO.
```

The `titleSuffix` metadata field applies a branding string, such as " - Xamarin", to all titles in scope.

Resolution

Apply `titleSuffix` globally or at the folder level via docfx.json. For example, the following JSON will append the string `- Xamarin` to all title values in the docset:

```
"globalMetadata": {
  "titleSuffix": "Xamarin",
   ...
}
```

Using `titleSuffix`, you can apply consistent branding to all articles in a docset or folder, without having to manually add it to every `title` value.

For more information, see SEO: Tips for writing titles.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

toc-breadcrumb-link-external

11/2/2020 • 2 minutes to read

Suggestion

```
TOC or breadcrumb link '{link}' goes to an external web site. TOCs and breadcrumbs should only link to pages
on docs.microsoft.com.
```

Resolution

Remove links to external sites from TOC and breadcrumb files.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

toc-missing

11/2/2020 • 2 minutes to read

Suggestion

```
Article '{file-name}' is not present in a toc.yml file. Every article must be in a TOC.
```

Resolution

Make sure the article or an article above it in its hierarchy is referenced in toc.yml file.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

token-not-found

3/5/2021 • 2 minutes to read

Warning

```
Token '{0}' can't be found in current repository.
```

Resolution

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-duration-missing

3/5/2021 • 2 minutes to read

Error

```
Missing attribute: DurationInMinutes. DurationInMinutes is required for each unit and must be greater than 0.
```

Resolution

Add the estimated number of minutes it will take a user to finish this unit.

For more information, see Create a unit in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-multiple-parents

3/5/2021 • 2 minutes to read

Error

```
Unit '{uid}' can't have two or more parents ('{parent-list}').
```

Resolution

Remove the unit UID from all but one parent module.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-no-module-parent

3/5/2021 • 2 minutes to read

Error

```
Unit '{uid}' must belong to a valid module.
```

This means either the unit doesn't have a parent or its parent isn't a module.

Resolution

Verify that the unit is the child of a valid module.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-task-and-quiz

3/5/2021 • 2 minutes to read

Error

```
A unit can't have both a quiz and a task.
```

Resolution

Remove either the quiz or the task.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-uid-missing

3/5/2021 • 2 minutes to read

Error

```
Missing attribute: uid. A valid uid is required for each unit.
```

Resolution

Add a unique unit UID in this format:

```
<prefix>.<module-folder-name>.<unit-name>

Such as:
learn.add-and-size-disks-in-azure-virtual-machines.introduction
```

Don't include file order prefixes in the unit name, such as "1-introduction". The unit *file name* will include a prefix to indicate its position in the module. The UID shouldn't in case the units get reordered in the future.

For more information, see Create a unit in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

validation-timeout

7/7/2021 • 2 minutes to read

Applies to: All content types

Warning

```
The call to the validation service timed out and validation was not completed. This happens when there's an
issue with the service and continuing to retry the call could cause build delays. You might have content
issues that were not reported. To retry validation, close and re-open your PR, or force a full build of your
branch via https://ops.microsoft.com. Note that forcing a full build requires admin permissions to the repo.
If you don't know who your repo admin is, or if you continue to see this message after a forced build, file
an issue via https://SiteHelp.
```

You get this Warning when there are transient issues in the validation service. For example, a server might be in a bad state. After several tries, the call times out and validation is canceled to avoid build delays and clogging the build pipeline.

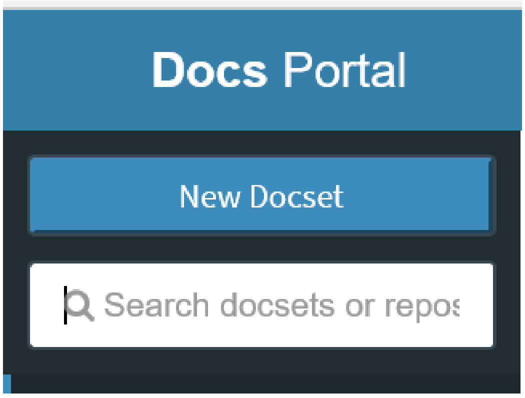
Resolution

Try closing and reopening your PR, or forcing a full build via Docs Portal. Often service issues clear themselves up after the initial retry.

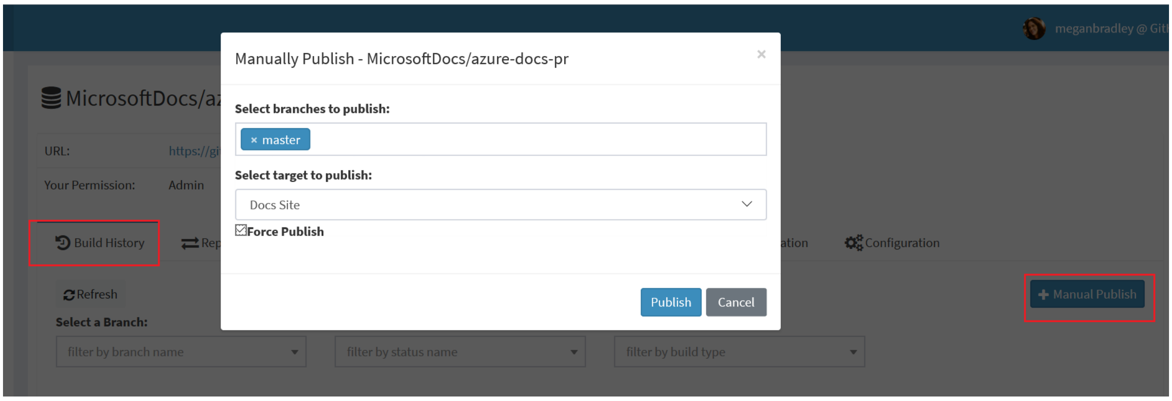
You must be a repo admin to force a build via Docs Portal. If you don't know who your repo admin is, or if you continue to get this message after a forced build, file an issue via https://SiteHelp.

If you're a repo admin, here's how you can force a full build:

1. Go to Docs Portal and sign in.
2. Find your repo by searching in the upper left corner, and select it.



3. On the **Build History** tab, select **+ Manual Publish**.
4. Select the branch that's getting the Warning, like **master**.
5. For target, keep the default, **Docs Site**.
6. Check the **Force Publish** checkbox.
7. Select **Publish**.



Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

xref-not-found

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

`Cross reference not found: '{uid}'.`

The referenced UID can't be found.

Resolution

Verify the UID and that XRef links are enabled for your docset. For more information, see XRef (cross reference) links and Xref Service.

> **TIP**
>
> To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

Error documentation not available

11/2/2020 • 2 minutes to read

No documentation is currently available for this validation issue. For help with the issue and to request documentation, please post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp.

Docs Local Validation

6/24/2021 • 2 minutes to read

The Docs Validation extension for Visual Studio Code allows you to perform build validations against your local Docs conceptual or Learn repository. This ensures your content is free of validation issues before making a pull request.

The Docs Validation extension is included in the Docs Authoring Pack.

Install Options

- Install as standalone
- Install as part of the Docs Authoring Pack

Prerequisites

- Install git.
- Clone your Docs or Learn repo locally in VS Code.
- All files in the repo must be saved.

Perform Local Validation

1. Clone your Docs or Learn repository locally.
2. Open the local repository in VS Code with the extension installed.
3. Make content changes, and **save** them locally.

   > **NOTE**
   > You do **NOT** have to commit them to your local repository yet.

4. If you're a Microsoft employee, optionally sign in by clicking **Docs Validation** from the VS Code status bar. You can still use the extension without signing in, but sign in is recommended to use the most recent build information for private repos.
5. Right-click anywhere on your folder/file structure and click on **Validate this repository**.

6. The process will start, and you can monitor the status in the output window

7. Once the process is complete, you can review any Validation issues identified in the "Problems" pane

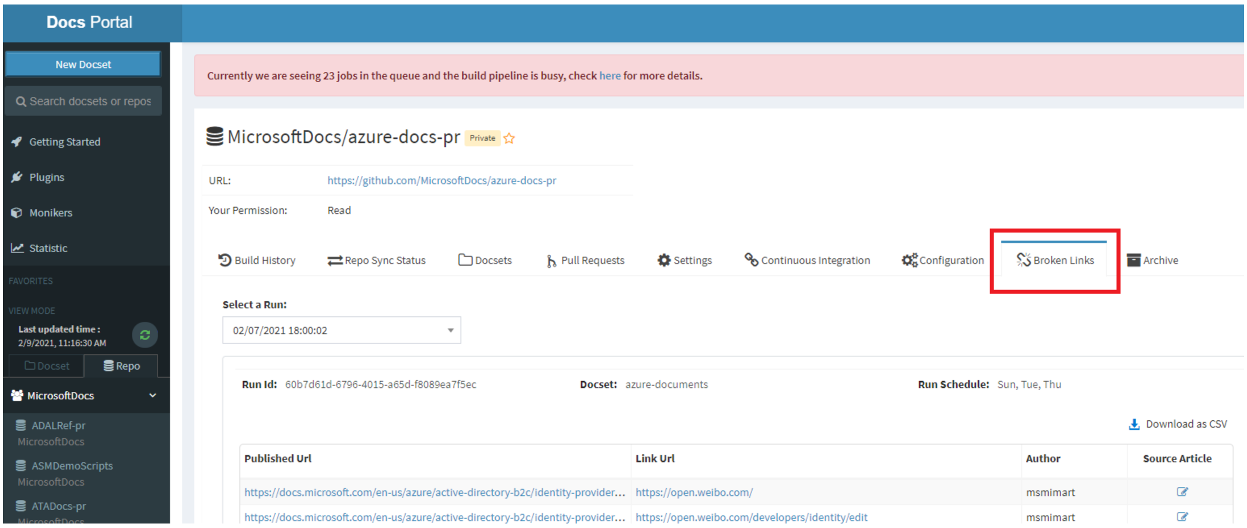
Details

From within VS Code a content creator can now perform the exact same Docs Build Validations against your locally cloned or forked repository **PRIOR** to submitting it into the hosted repository and waiting the Docs Build pipeline to evaluate your content.

Broken link report in Docs Portal

11/2/2020 • 5 minutes to read

A new broken link report is enabled on all repos that are live on Docs. The report is available through the Docs Portal. Any user with Read permissions for a repo can view the broken link report for that repo. Just find the repo in the portal and click the **Broken links** tab on the far right:



If the **Broken links** tab isn't available for your repo, it means the repo hasn't been published live since the broken link report was enabled. Publish the repo live to see broken link results and begin weekly runs.

When you click the **Broken links** tab, you'll see the results of the latest run. You can select a different run of the most recent three from the **Select a Run** drop-down box. By default, the report runs once per week. It can be run more frequently if needed; contact mbradley to request more frequent runs.

On the web view of the report, you'll see the **Published Url** of the article that contains the broken link, the **Link Url** of the broken link itself, the **Author** of the article by GitHub ID, and a **Source Article** edit button to open the article in GitHub or DevOps. Click **Download as CSV** at the top right corner of the results to download a sortable, filterable CSV with the following additional columns:

- **ms.author**: The author's Microsoft alias.
- **ms.service**: The ms.service value, if present.
- **ms.subservice**: The ms.subservice value, if present.
- **ms.prod**: The ms.prod value, if present.
- **ms.technology**: The ms.technology value, if present.
- **ms.topic**: The ms.topic value.

What types of issues are checked for?

The broken link report returns results for links that return 404 or otherwise unavailable when pinged by the broken link service. This includes links in conceptual Docs articles, reference articles, Learn modules, TOCs, and breadcrumbs. It includes links to web sites external to docs.microsoft.com, links to other docs.microsoft.com pages, images, videos, and samples. It doesn't include links to non-existent bookmarks on valid pages, because these links fall back to the beginning of the article and therefore return success.

The report also returns results for security issues.

The following are result categories:

CATEGORY	DESCRIPTION	USER ACTION
Broken	The link returns 404 or another HTTP status code that indicates that the link is permanently unavailable.	Replace the link with a valid one, or remove it altogether.
Security issue	The link has a security issue, such as an invalid certificate, and might cause a vulnerability.	We strongly recommend that you remove the link. If the link is essential, contact the site owner to fix the security issue.

Is the broken link report integrated into Docs Build?

Although the broken link results show in Docs Portal, the report isn't currently integrated into Docs Build. Some types of links - specifically Markdown links to other articles on Docs, image links, code links, and bookmarks - are checked at build time. However, many of these links can become broken after the fact if assets are moved or removed, so build checking isn't sufficient to ensure no 404s on the live site. Links to sites external to Docs aren't checked at build time because it would degrade performance. Therefore the broken link report is separate from Docs Build, although some integration is planned for the future.

What about CATS?

The Docs Portal broken link report replaces CATS, which was costly to maintain and didn't scale well to support all of Docs. The main differences between CATS and the new report are as follows:

	CATS	DOCS PORTAL
Scope	Ran only on demand or as manually scheduled via CATS.	Runs on all Docs repos that have been published live at least once since the report was enabled.
Cadence	Ran on demand or as scheduled.	Runs weekly by default. Can be run more frequently if needed.
Docs-to-Docs links	Checked.	Checked.
External site links	Checked.	Checked.
Bookmarks	Checked.	Not checked, because broken bookmarks fall back to the top of the target page and don't 404. Bookmarks are now checked as part of build validation. For more information, see bookmark-not-found.
TOC links	Checked as a separate content validation, "Broken TOC links".	Included in the broken link report.
Breadcrumb links	Checked as a separate content validation, "Broken breadcrumb links".	Included in the broken link report.
Images, videos, code samples	Not checked.	Checked, because missing assets return 404.

	CATS	DOCS PORTAL
Link allowlist	For URLs that weren't meant to be clickable links, such as domain names, namespace references, and example links like `www.contoso.com`, CATS maintained an allowlist to exclude results. This solution wasn't ideal, because the list was quite broad and sometimes cause real broken links to be missed. Links that weren't meant to be clickable also rendered as real links, which did in fact return 404 if clicked.	In the new report, we've eliminated the allowlist. To avoid broken link reports, links that aren't meant to be clickable should be styled as inline code (`like this`).

What do I do to exclude non-clickable links from results?

If your repo used the CATS link allowlist to exclude URLs from the report, you'll find results in the new report that seem to be false positives. The solution is to style URLs that aren't meant to be clickable links, or that only 404 under certain circumstances, as inline code (`like this`). In addition to excluding these URLs from the report, this will render them as non-clickable on the Docs site, so users won't be confused by the appearance of clickable links that actually return 404.

The following is an example:

```
`https://gallery.azure.com`
```

It renders like this on the Docs site:

`https://gallery.azure.com`

Because the old allowlist was quite broad, you might want to review these results to make sure they're true exceptions before you convert them to inline code. If you used the allowlist heavily and want a more efficient solution than converting each one manually, a script is available to find URLs in a repo that match the patterns specified in the allowlist and wrap them with inline code. This script will be run on the core Docs and Learn repos supported by the Developer Relations PR Review process/PRMerger as part of migration to the new report. If you want help scrubbing a repo that isn't supported by the PR Review process, send email to mbradley.

CATS deprecation

6/28/2021 • 6 minutes to read

The CATS site QA tool was developed over several years to test published Docs content for missing metadata, broken links, content rendering problems, and so on. Because CATS was supported by a large vendor team, it was often easier to get work done on CATS than within the Docs pipeline. Many content tests were implemented as site QA tests when implementation as build validation would have been more enforceable and more consistent with other validation experiences.

Now that Docs is a mature product, we've deprecated CATS and replaced it with functionality within the core Docs authoring platform. The main replacement features are new build-time validations, a new broken link report, and new or updated tools to provide author-time feedback on issues that aren't clear cut enough for build validation, such as some SEO guidelines.

CATS is fully deprecated as of June 30, 2020.

Final CATS deprecation: June 30, 2020

All priority CATS checks have now been enabled as build validations or alternative tooling, or will be turned on in Docs Build early in FY21. Some lower priority CATS tests are considered obsolete and will not be replaced.

The following CATS replacement validations were enabled on June 29, 2020:

- alt-text-missing
- column-header-missing
- duplicate-h1s
- duplicate-titles

The following CATS replacement validation is available for ad hoc runs, but has not been enabled in build because it isn't currently required by the Docs platform:

- alt-text-bad-length

The following CATS replacement validations are currently in testing/rollout and are expected to be enabled in build or made available as ad hoc validations at the end of July:

- alt-text-bad-value
- description-bad-length (ad hoc only - use Search Results Preview in the Docs Authoring Pack for VS Code to verify your titles and descriptions at author time)
- description-missing
- dev-language-missing
- duplicate-alt-text
- duplicate-descriptions
- duplicate-toc-entries
- path-duplication
- path-too-deep
- title-suffix-missing
- toc-breadcrumb-link-external
- toc-missing

To request an ad hoc run on your repo of any validations not currently enabled, or if you have questions about replacements for specific CATS tests, please contact mbradley.

Deprecated as of April 17, 2020: CATS broken link checks

CATS provided broken link checking on demand or on a schedule via the **Docs Content Validation** tab:

- **Broken internal links**
- **Broken bookmarks**
- **Broken TOC links**
- **Broken breadcrumb links**
- **Broken external links**
- **Broken video links**
- **Broken image links**
- **Broken sample links**

These tests have been replaced with a new broken link report that runs on all live Docs repos and can be accessed via Docs Portal. For more information about the new report, including how it differs from CATS, see Broken link report.

Deprecated as of November 20, 2019: Alt text length, insecure links, H1 headings

Alt text length test

The P2 **Alt text length test** under SEO rules gives outdated guidance, telling users to keep alt text between 30 and 70 characters, including spaces. The current maximum for alt text is 250 characters, with no specific minimum. For more information about alt text and other accessibility requirements for images, see Accessibility Guidelines for Multimedia.

Docs Build validation for alt text issues is coming early in 2020. Because the CATS test is P2 with incorrect results, we're removing it from CATS even though build validation isn't ready yet to avoid confusion and unnecessary work for authors.

Test for insecure links

The insecure-link validation has been added to the Docs Platform ruleset, which is enabled on all conceptual repos by default. This validation replaces the CATS **HTTPS Checking test**, which checks that links to Microsoft sites use http instead of https. The redundant CATS test is therefore deprecated.

H1 heading tests

The following validations have been added to the Docs Platform ruleset to replace the CATS **Missing or Extra H1 heading** test:

- h1-empty - checks for an H1 marker (#) with no text following it.
- h1-missing - checks for no H1 in an article.
- h1-no-space - checks for an H1 with no space after the hash (#), which Docs Build interprets as plain text.
- multiple-h1s - checks for more than one H1 in an article.

The redundant CATS test is now deprecated.

Docs Archived Content Validation

The Docs Archived Content Validation pane offered a set of tests specifically for migrating content from MSDN/TechNet to the Docs previous versions archive. These migrations are now complete, and the tests are no longer needed.

Deprecated as of October 11, 2019: Metadata tests and Side-by-side comparison tests

Required metadata populated by author

The following metadata must be present and valid: `author`, `ms.author`, `ms.date`, `ms.topic`, `title`, and `ms.service` OR `ms.prod`. `ms.subservice` is optional, but must be a valid pair with `ms.service` if specified. `ms.technology` is optional, but must be a valid pair with `ms.prod` if specified.

For **conceptual** content, these tests are already available as build validations and enabled for most repos that rely on these values for content reporting. For more information, see the Developer Relations/CGA reporting ruleset.

If you currently use CATS for metadata validation and do *not* yet have the full validation ruleset, file a request using this template.

For **reference** content, build validation for metadata is not yet available (ETA is FY19 Q3). However, there are other ways to verify your reference metadata. Author-added reference metadata is set in the docfx.json file, as described in this article. It's recommended that you set default values at the highest level possible, and that you use team aliases instead of individual aliases, to minimize churn in your values. Reference docsets are supported by the Metadata Health report, which will remain available until metadata build validation is enabled for reference. If you can't access the report or if your reference repo isn't shown, contact Khairun Jamal.

Missing or extra title

The `title` metadata field is also validated for conceptual content. It is required and can only be specified once. For reference content, `title` is autogenerated.

Required metadata populated by OPS

These tests were added in the early days of Docs when Docs component testing didn't cover system-generated metadata. The tests only fail if something goes wrong in the system, and there's nothing authors can do to fix failures. New component tests have been added to catch these issues so author-facing tests are no longer necessary in CATS.

APIScan metadata validation

APIScan metadata is automatically generated for .NET, UWP, and Java reference. This metadata is now covered by end-to-end component testing, so author-facing tests are no longer necessary in CATS.

Docs Side-by-side Comparison and MSDN/TechNet Migration Validation

These side-by-side comparison tests were created for use in Docs content migrations, but are no longer used by the migration team and will be removed from CATS. Future migration efforts will implement testing as part of the migration work.

FAQ

Will CATS tests go away without warning?

No. All deprecations will be announced two weeks before functionality is removed from CATS. To ensure you receive notifications, join the Docs Authoring Partner Announcements alias.

I depend on CATS test X. How do I know what you're planning to replace it with?

An overall CATS deprecation plan is posted here. You can review it for information about how we plan to replace various features of CATS tests. The plan is not finalized and some replacements may change before implementation. Check the linked features for current status of replacement functionality.

Is CATS supported during deprecation?

Yes, but at a lower priority than in the past. No new features will be accepted, and bugs will be prioritized based on how critical the functionality is. As features are replaced, they'll be removed from CATS.

DocsImageSorter

7/9/2021 • 2 minutes to read

The DocsImageSorter is a prototype tool that uses Azure Cognitive Services to identify issues with images. The prototype is a command-line tool that you can download and run on your repo. Currently the tool distinguishes between complex and standard images, and raises a failure if a complex image lacks a long description. More functionality, including identifying images with bad contrast and identifying images that contain sensitive terminology, are planned for the future.

To use the tool, first download it from one of the following locations and extract the .zip file:

- Windows: Download here
- Mac: Because of new MacOS security updates, DocsImageSorter is offline for Mac. We can address as resources and priorities allow.

Then, in a command prompt:

1. Navigate to the folder where you extracted the .zip.
2. Type the following command: `DocsImageSorter.exe -p "<repo-path>" -o "<csv-output-path>"`, such as:

```
DocsImageSorter.exe -p "C:\Users\mbradley\docs-help-pr" -o "C:\Users\mbradley\Desktop\image-output.csv"
```

3. Open the CSV output file. Review the columns:

COLUMN	DESCRIPTION
SOURCE	The path to the article that references the image.
IMAGE	The image file name.
MS.AUTHOR	The `ms.author` value for the source article.
REFERENCE TYPE	The type of Markdown image reference, as follows: - standard: A standard Markdown image in the format `![<alt text>.](<folderPath>)`. - complex: The Docs `:::image:::` extension with `type="complex"`. - content: The Docs `:::image:::` extension with `type="content"`. - icon: The Docs `:::image:::` extension with `type="icon"`.
ACTUAL TYPE	The actual type of the image, based on the machine learning model. Currently the model only distinguishes between `complex` (meaning an image is complex and requires a long description as well as alt text for accessibility) and `content` (all other images, including icons). `Undetermined` means the model was unable to confidently categorize the image.

COLUMN	DESCRIPTION
PASS/FAIL	Whether the image meets relevant guidelines. Currently, PASS means either that the image is complex and referenced as `:::image:::` with `type="complex"`, or the image is standard or icon and referenced appropriately in standard Markdown or the `:::image:::` extension. FAIL means the image is complex but isn't referenced as `:::image:::` with `type="complex"`. Undetermined means the model can't confidently categorize the image.
PROBABILITY	The confidence, between 0 and 1, in the image categorization. If PROBABILITY is less than 0.92, ACTUAL TYPE is returned as "Undetermined".

4. Fix any failures by updating the image reference in the source article. That is, if you have a complex image that isn't referenced as complex, update the reference to use the `:::image:::` syntax with `type="complex"` and a long description as described in Complex images with long descriptions.

Known issues

- DocsImageSorter doesn't work as expected on recent versions of MacOS. The Engineering team can't prioritize fixes at this time.

Automated archive solution

6/24/2021 • 4 minutes to read

This article describes how to setup the machine to use the automated archive tool. The tool will allow archiving the content from current repo to archive repo. The tool works both in GitHub and Azure DevOps repo. The tool will create two PRs one in current repo and second one in archive repo.

System preparation and requirement

Install Microsoft .NetCore SDK: Go to https://dotnet.microsoft.com/download/dotnet-core/thank-you/sdk-2.2.207-windows-x64-installer and install version .Net 2.2

> **NOTE**
> If you currently have a more recent version (3.0, 3.1, etc.), install both, as 2.2 is currently required for the Archive tool.

Install Git: Install latest Git version from https://git-scm.com/download.

Setup Git: Run following command.

```
git config -list
```

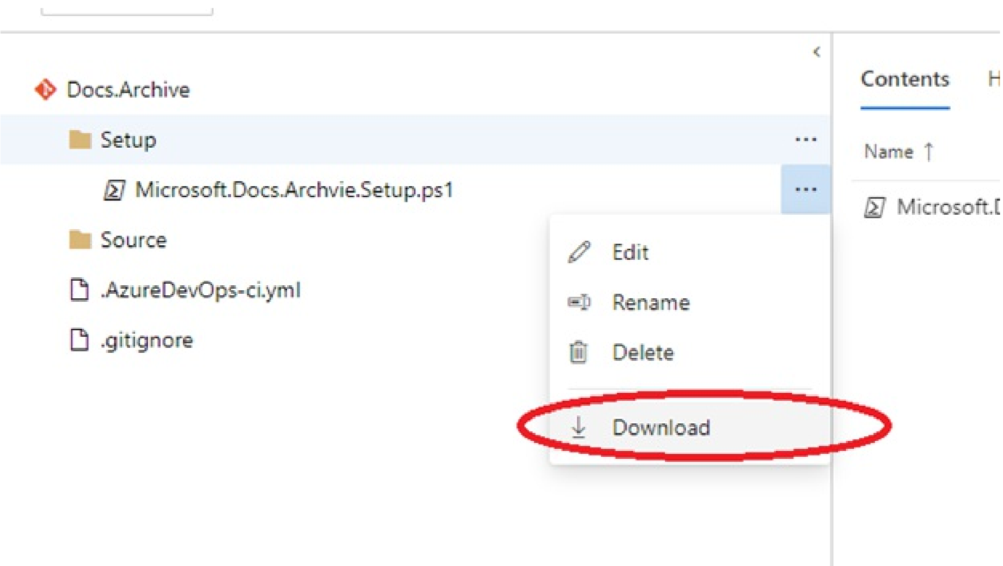
If "user email" and "user name" is not configured, run following command:

```
git config -global user.email "{your email address}"
```

```
git config-global user.name "{your user name}"
```

Download source files and setup scripts

Download Microsoft.Docs.Archive.Setup.ps1 from https://dev.azure.com/ceapex/Engineering/_git/Docs.Archive?path=%2FSetup%2FMicrosoft.Docs.Archive.Setup.ps1&version=GBmaster



Setup PowerShell execution policy Setup PowerShell execution policy to bypass, more information at https://docs.microsoft.com/powershell/module/microsoft.powershell.security/set-executionpolicy

Open PowerShell console as administrator

Run: Set-ExecutionPolicy Bypass

Unblock setup script

Open PowerShell console as administrator.

Unblock-File .\Microsoft.Docs.Archive.Setup.ps1

Installation

1. Open PowerShell window
2. Run the downloaded PowerShell script with "install" parameter.

```
.\Microsoft.Docs.Archive.Setup.ps1 install
```

How to create configuration file

To create a sample configuration file you can use following command.

```
PS C:\Users\username> docs.archive generateSampleConfiguration -o C:\Archive\sample
```

CONFIGURATION ELEMENTS	ELEMENT DETAILS
WorkingFolder	Setup the working folder. All file operations will happen under this folder. If not provided, current directory will be the working folder. E.g. D:\archive
Source	Current repo information
OpBuildUserToken	OpBuild User Token to operate on source repository. You can get the token from cookie in ops portal: https://ops.microsoft.com. The token lasts for 24 hour
PersonalAccessToken	Personal Access Token for AzureDevOps or Github. You can get the token from AzureDevOps or Github security settings.
SourceRepository	Archive source repository url. e.g. https://github.com/MicrosoftDocs/azure-docs-pr or https://cpubwin.visualstudio.com/_git/cortana.
DestinationBranch	The branch name that the archiving will be worked on and the final pull request targets. Live branch is not allowed.
DocsetName	Archiving source docset name.
Files	The file list to archive, example from Azure-docs-pr `articles\\storage\\common\\storage-import-export-tool-preparing-hard-drives-import-v1.md` . The tool also supports wild cards list.
TocNodes	Example from repo https://github.com/MicrosoftDocs/globalization "FilePath": "globalization\toc.yml", "TocPaths": "Globalization Overview\Documentation\Data Encoding"
Target	All information in this section is about archive repo

CONFIGURATION ELEMENTS	ELEMENT DETAILS
RepositoryType	AzureDevOps or Github
Account	Github or AzureDevOps account name. e.g. MicrosoftDocs, docs-archive
AzureDevOpsProject	AzureDevOps project name. e.g. docs-archive-project, For GitHub, leave empty.
RepositoryName	Target repository name, it can be an existing archive repository for example azure-docs-archive-pr or brand new one.
IsPrivate	the value can be "true" or "false"
DestinationBranch	The branch name that the archived content will be placed and the final pull request targets. Live branch is not allowed.
DocsetName	It can be an existing docset name for example Azure-docs-archive or new docset
BasePath	Base url path for the archiving target docset. e.g. /previous-versions/. Domain name https://docs.microsoft.com is not allowed. If the docset has already been provisioned, this is not required.
DocsetFolder	The folder name for archiving target docset. If the docset has already been provisioned, this is not required.
SubFolder	If SubFolder provided, the tool creates a sub folder in docset root and all archived items will be put into the sub folder. Not a required field.
GlobalMetadata	Any metadata that you want to add globally, you must enter "ROBOTS": "NOINDEX,NOFOLLOW" and "is_archived": "true"

Different run options

How to run archive tool

Open command window, run below command:

```
docs.archive archive -c <configuration file and path>
```

> **NOTE**
> One config maps one folder. If you run a specified archiving from working folder "A", you should not run other archive for different source repository or target repository on this "A" folder again. You should create a new folder or clean up "A" before you run another archive. But , if you run archiving against the same source and target repository again and again, you can still use the folder "A" and no need to clean it up.

How to get latest version of the tool

Run the downloaded Powershell script with parameter update `.\Microsoft.Docs.Archive.Setup.ps1 update`

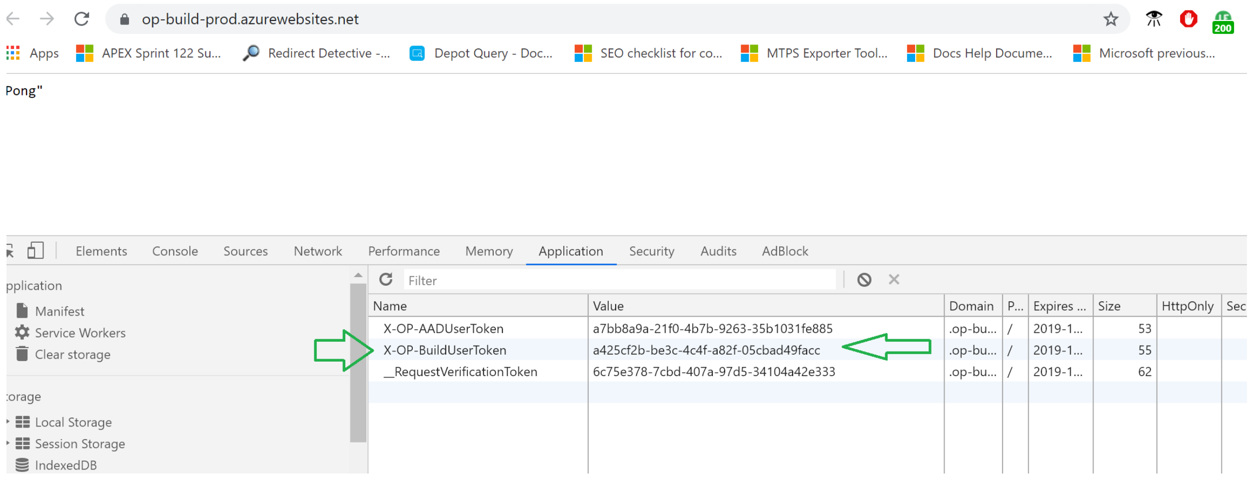
How to uninstall the archive tool

Run the downloaded Powershell script with parameter uninstall install

```
.\Microsoft.Docs.Archive.Setup.ps1 uninstall
```

How to get the OpBuildUserToken

1. Open Docs portal
2. Sign in with "GitHub" or "Azure DevOps" according to your repository type
3. Go to https://op-build-prod.azurewebsites.net/ in Chrome and press F12 to open Developer Tool, find the cookie in Application\Cookies\https://op-build-prod.azurewebsites.net)
4. The value of the cookie is OpBuildUserToken and is used both in source and target section of the config file.



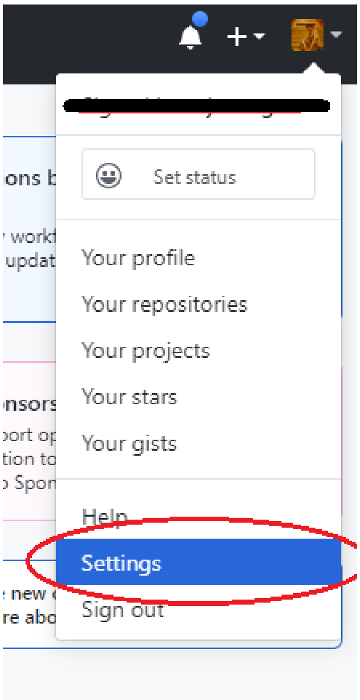
> **NOTE**
>
> The token will expire after 24 hours so make sure to get the new token every day otherwise you will get either 401 or you don't have access error.

How to get PersonalAccessToken

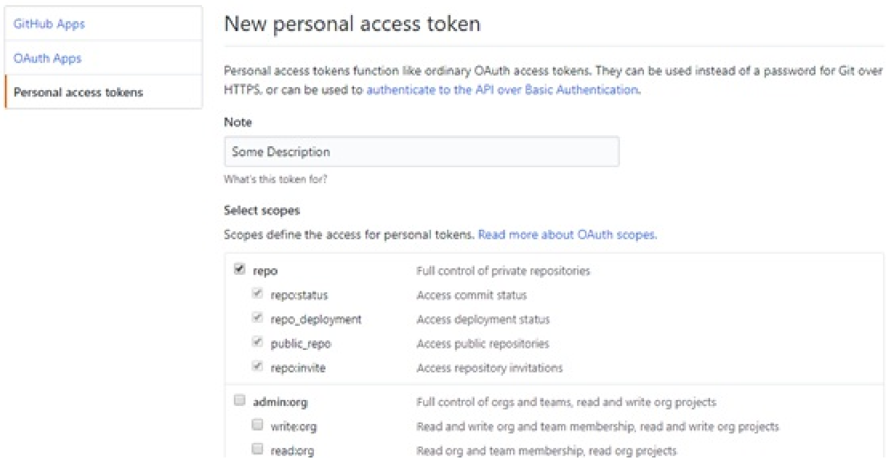
You will need to know this personal access token for both GitHub and Azure DevOps.

GitHub

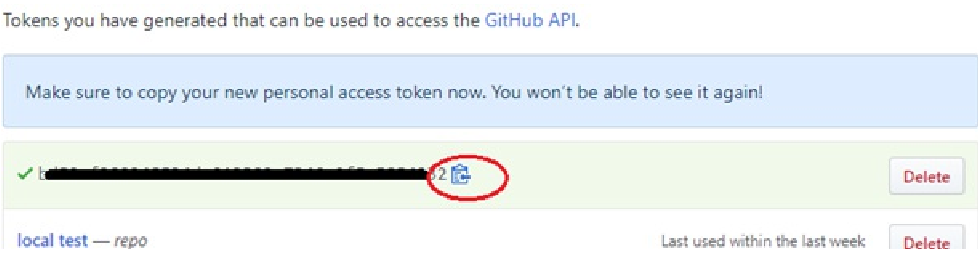
- First step is Logon to GitHub and go to settings.

- 

- Select "Developer settings"
- Select "Personal access tokens"
- Click on "Generate new token" (you may be required to enter your Github credential)
- Provide some description and make sure "repo" is selected, click "Generate token" button below the page.



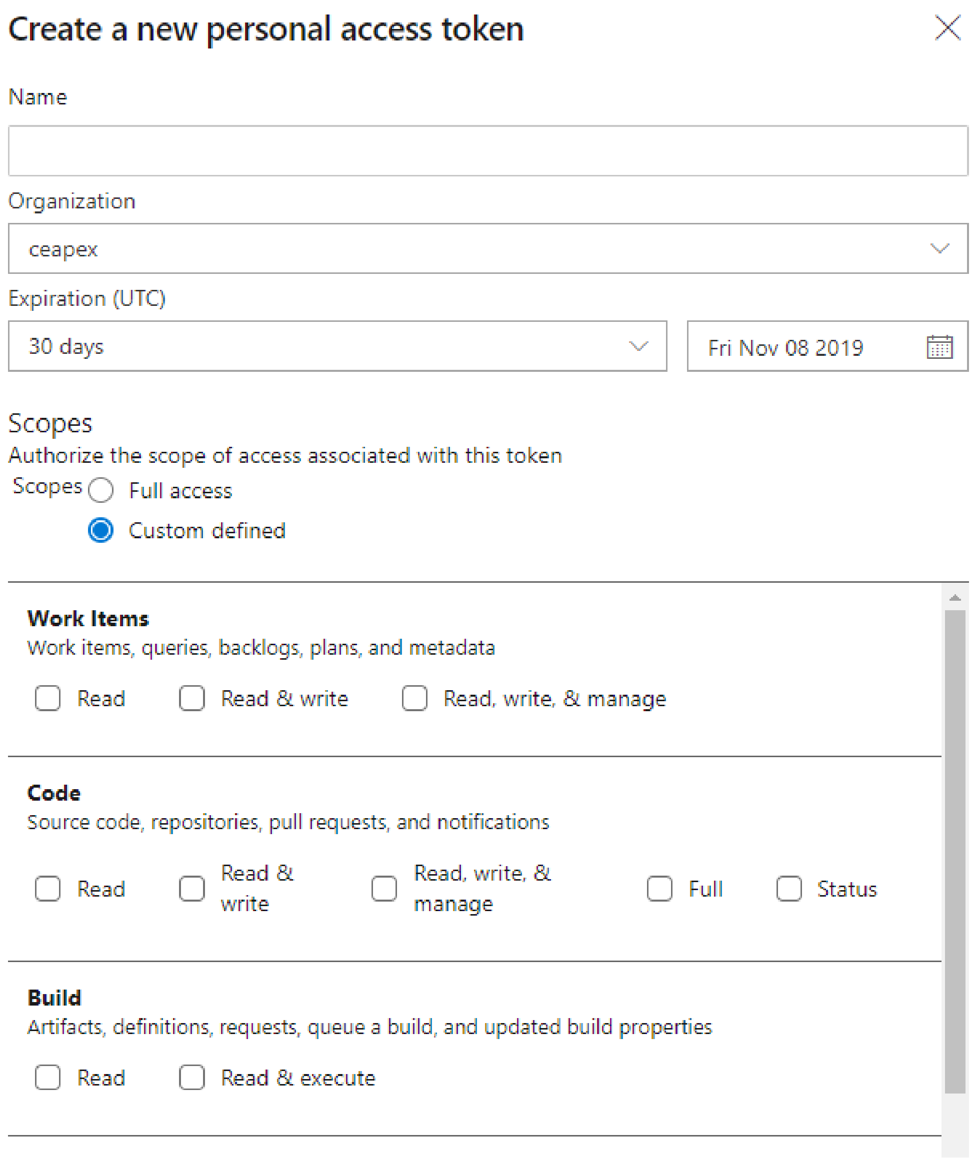
- Copy the token value and keep it in a safe location for future use, you will need this every time you will run archive tool.



Azure DevOps

- Open Azure DevOps project, e.g. https://dev.azure.com/ceapex/
- From Login drop down select security option.
- Select "Personal access tokens"
- Click on "New Token" button

- Provide token name, target organization and expiration.
- In Scopes section, make sure all the scopes under "Code" is checked.



- Click on "Create" button below
- Copy the token value

Use CodeFlow to review PRs

6/24/2021 • 3 minutes to read

Reviewers can use CodeFlow as a tool to review pull requests (PRs) in content repositories. CodeFlow is an alternative to the GitHub web UI PR review tools - users who already use CodeFlow may prefer to use a tool they're familiar with.

Get started with CodeFlow

If you decide to use CodeFlow as your PR review tool, you should review the articleWorking with GitHub Pull Requests. The article includes:

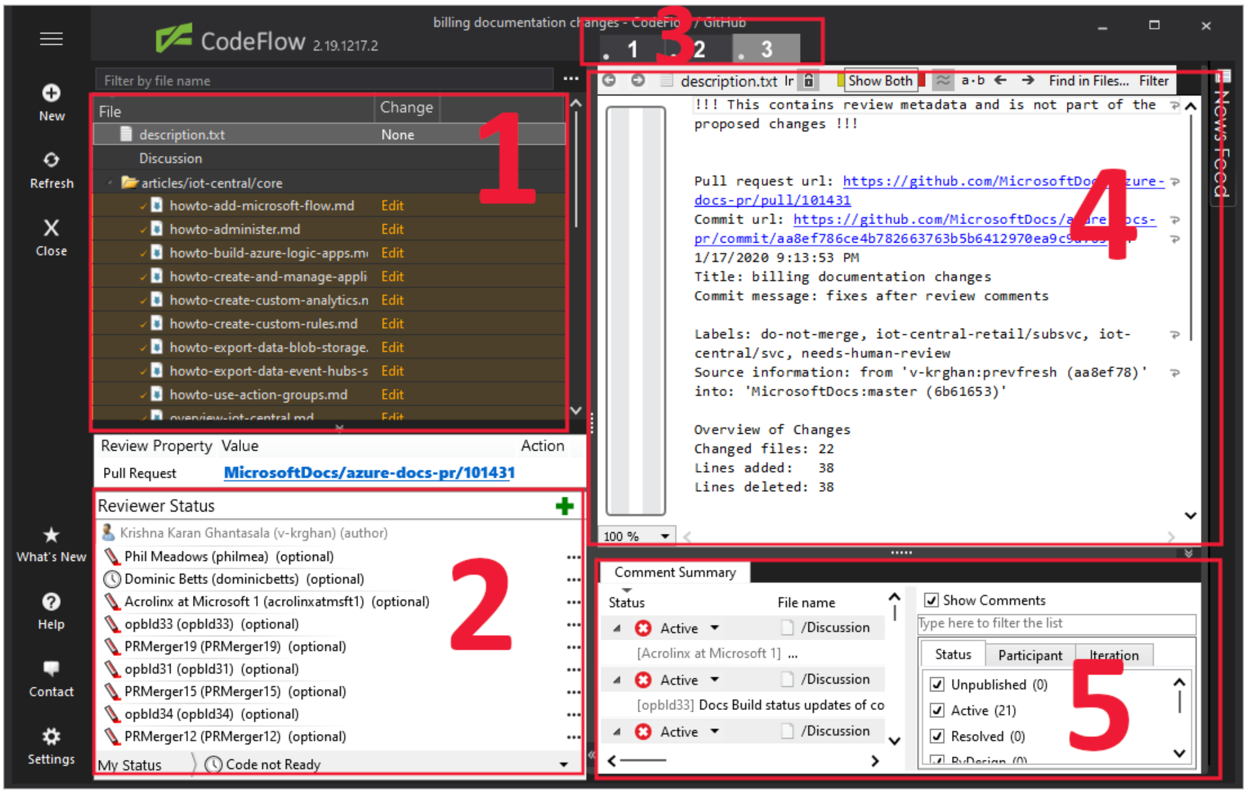
- One-time setup requirements to configure CodeFlow with your GitHub credentials.
- Opening pull requests using Chrome or Firefox extensions, or by using a console command.

> **TIP**
>
> The Edge browser can now install extensions from the Chrome Web Store.

View a PR in CodeFlow

The following screenshot shows the CodeFlow UI with an open PR from the **azure-docs-pr** GitHub repository:

The key areas of the UI are:

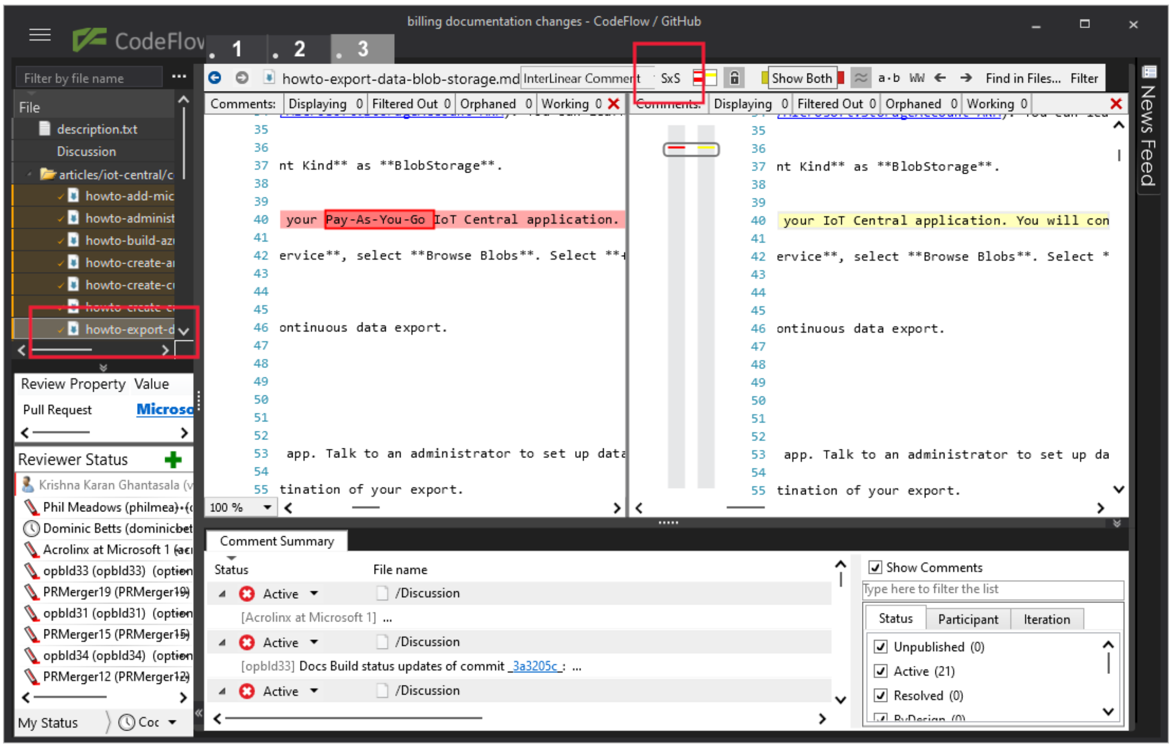


1. The list of modified files included in the PR. **Description.txt** contains generated metadata and isn't part of the proposed changes. **Discussion** includes the Acrolinx scorecard and staging links.
2. The list of reviewers associated with the PR. The PR author is listed first. Acrolinx and the validation bot

appear here when they have run. You can view the comments generated by Acrolinx and the validation bot if you choose **Discussion** in the file panel.

3. There's a tab for each commit in the PR that enables you to view the state of the files at that point. You can see the difference between two commits by Ctrl-clicking on two tabs. By default, CodeFlow displays the most recent commit with the final state of the files in the PR.

4. The file viewer/comment area enables you to view and comment on the files in the commit. Use the toolbar to control how the files, differences, and comments are displayed. The following screenshot shows the side-by-side *diff* view:



5. The **Comment Summary** area shows a list of the comments made in the PR. You can use the checkboxes on the right to filter by comment status, participant, and iteration. Iteration refers to the commit the comment was made against.

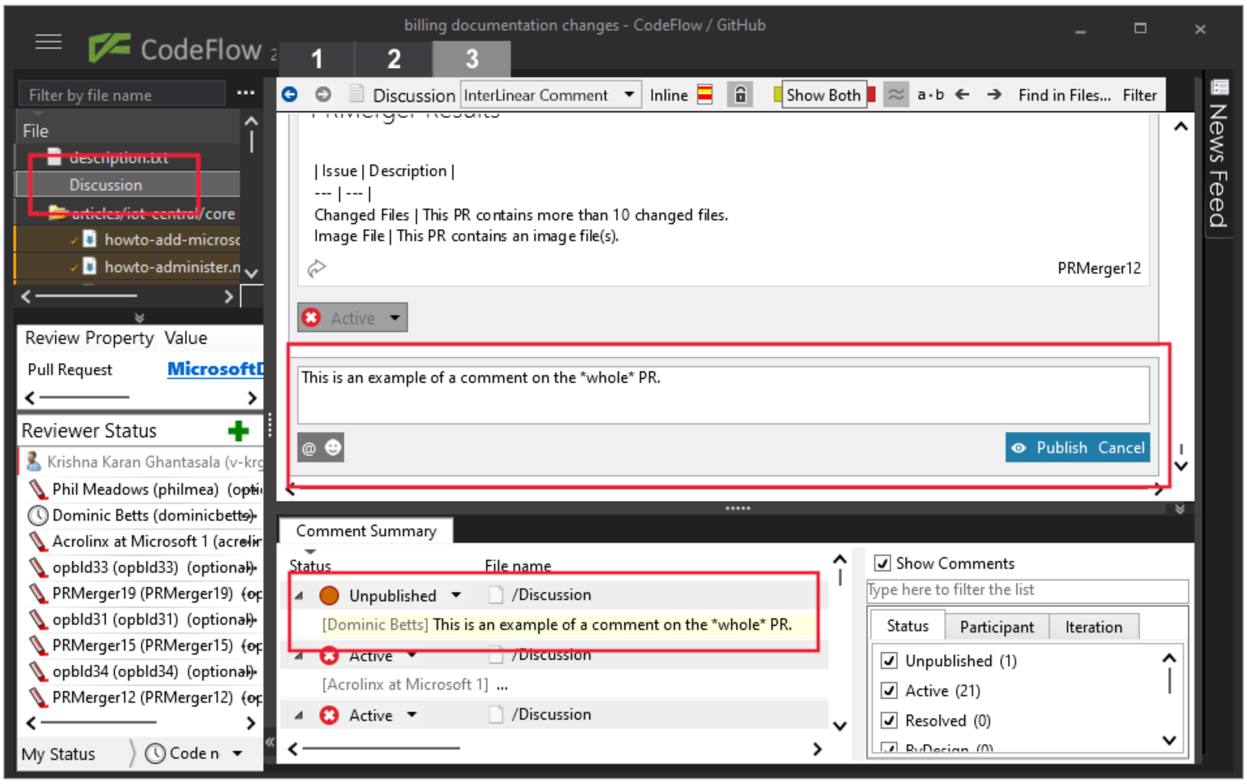
Add comments to a PR in CodeFlow

As a reviewer, you can add comments to a PR in CodeFlow in the following ways:

You can comment on the PR as a whole

1. Choose **Discussion** in the **File** panel. This panel is where the Acrolinx scorecard and validation links display.
2. Right-click in the file viewer/comment area and choose **Add comment**.
3. Enter your comment and choose **Publish**.
4. The comment appears in the **Comment summary** area in the **Discussion** iteration.

> **TIP**
>
> Use markdown to add formatting to your comments.

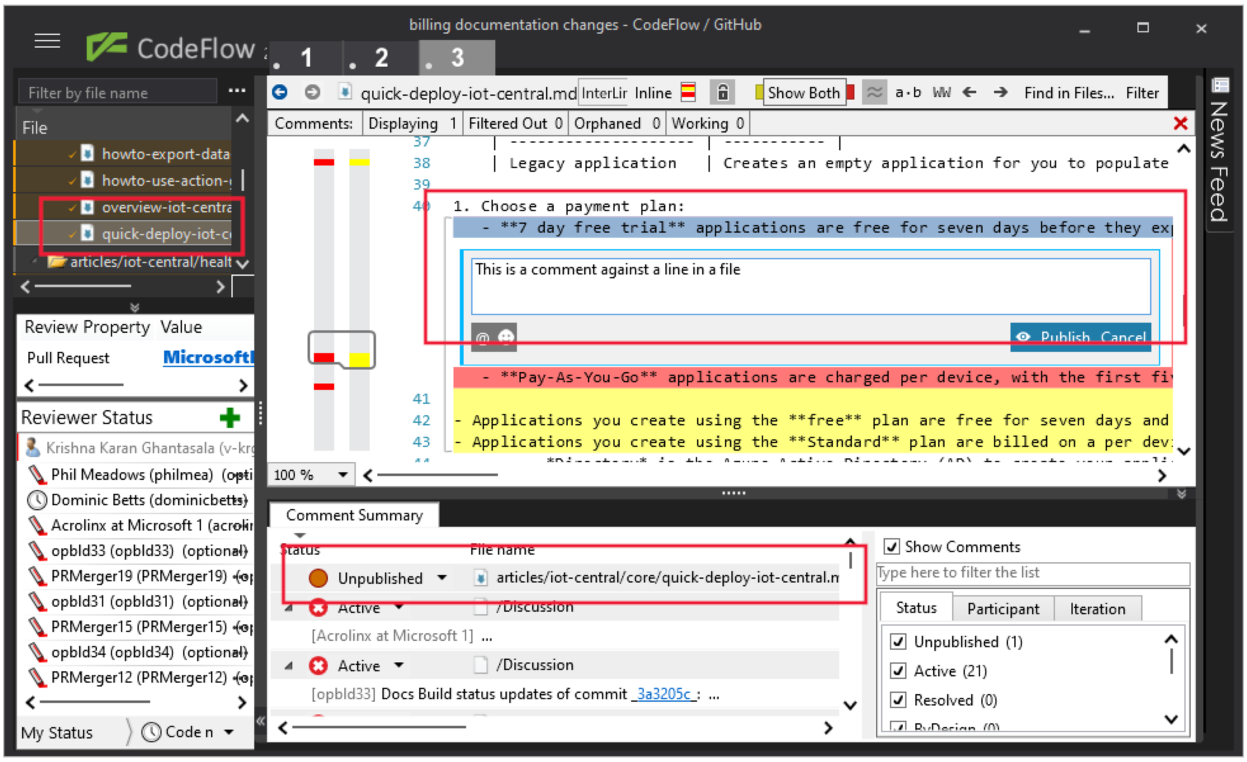


You can comment on a specific line in a file in the PR

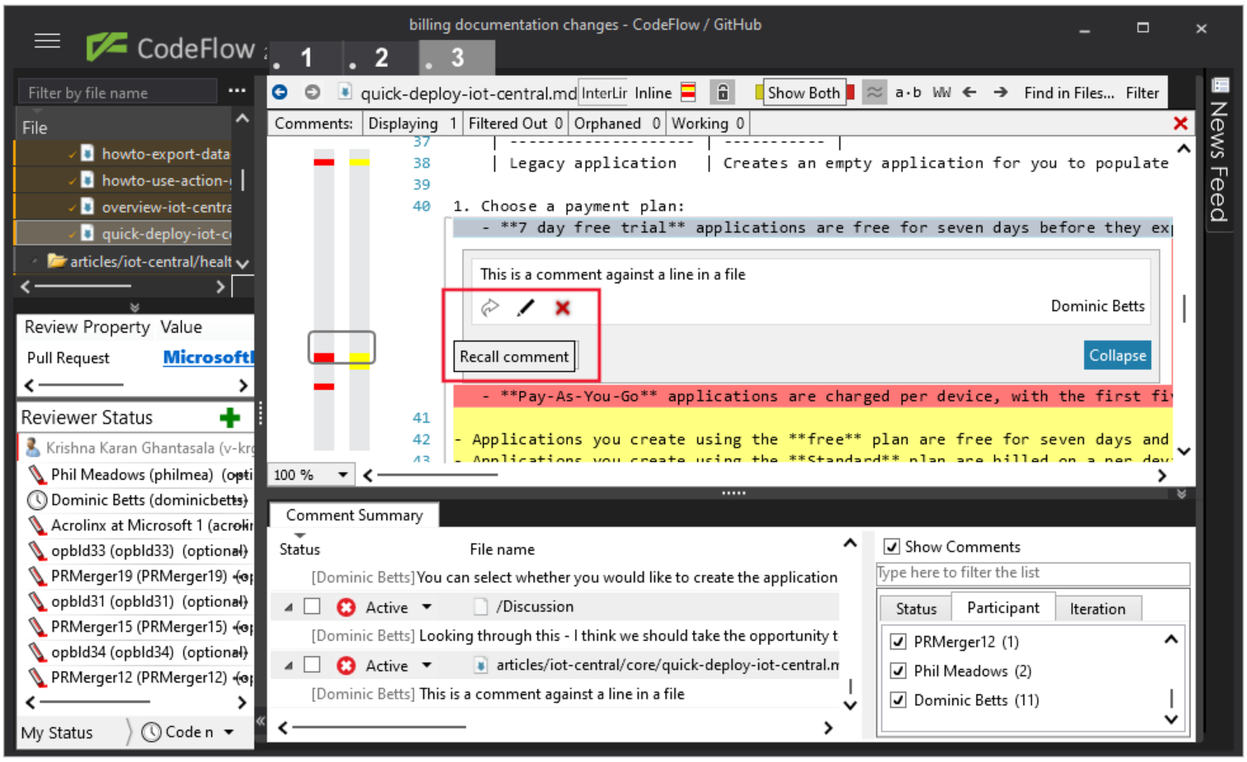
You can comment on any line in a file that's included in the PR, whether or not it was changed in the PR.

1. Choose the file you want to add a comment to in the **File** panel.
2. Right-click on a line in the file viewer/comment area and choose **Add comment**.
3. Enter your comment and choose **Publish**.
4. The comment appears in the **Comment summary** area in the **Discussion** iteration.

> **TIP**
>
> Use markdown to add formatting to your comments.

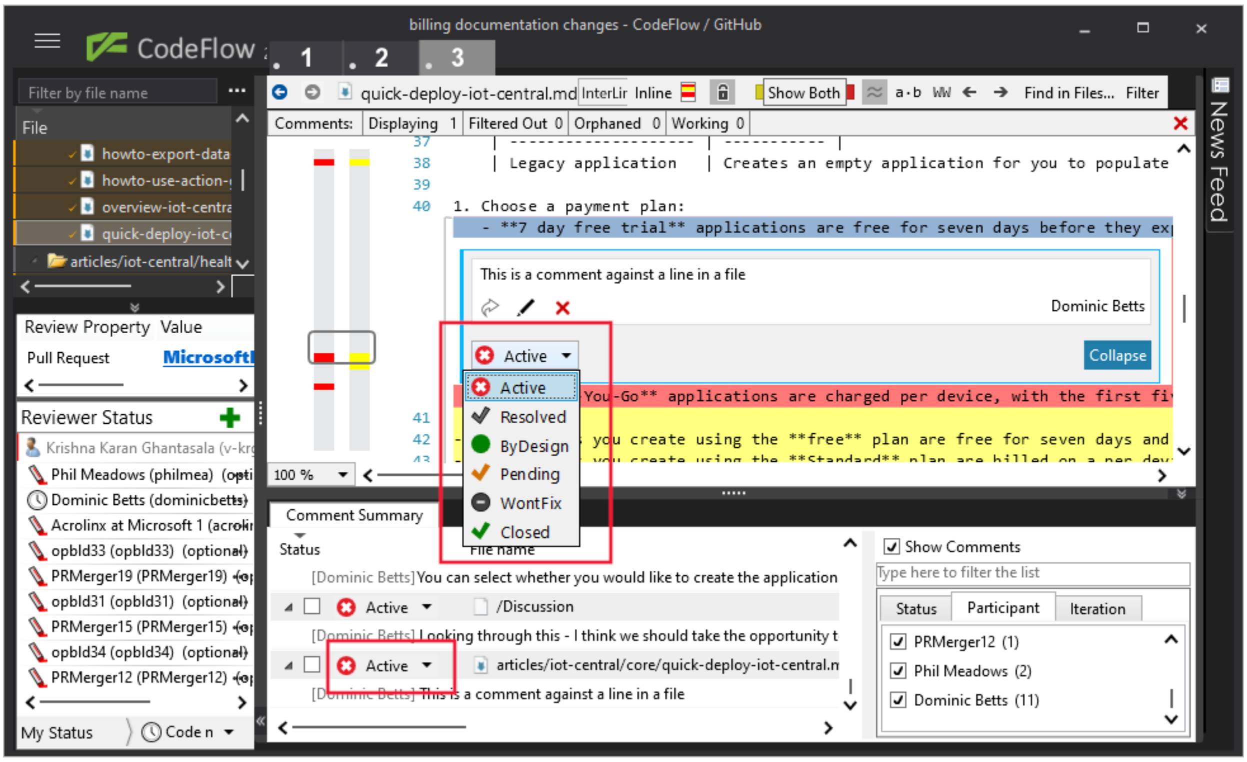


As a comment author, you edit or recall a comment you created. If the comment was created by another reviewer, you can post a reply:



All the comments are saved in the PR itself, CodeFlow doesn't use any external store when you use it to review a GitHub PR.

You can change the status of the comments you made in the PR:



You can change your status as reviewer in the **Reviewer Status** panel as follows:

- **Code not Ready** indicates that you expect the author to make changes before signing off the PR.
- **Approved** indicates that you approve the changes and the author can sign off the PR.

Manage the review process in CodeFlow

Invite reviewers

As the PR author, you can invite reviewers from within CodeFlow by adding a valid GitHub username in the **Reviewer status** area. As an alternative, you can email reviewers asking them to provide feedback on your PR.

In both cases, you must explain to users how to open the PR in CodeFlow. If you send an email, you can include the console command to run, for example:

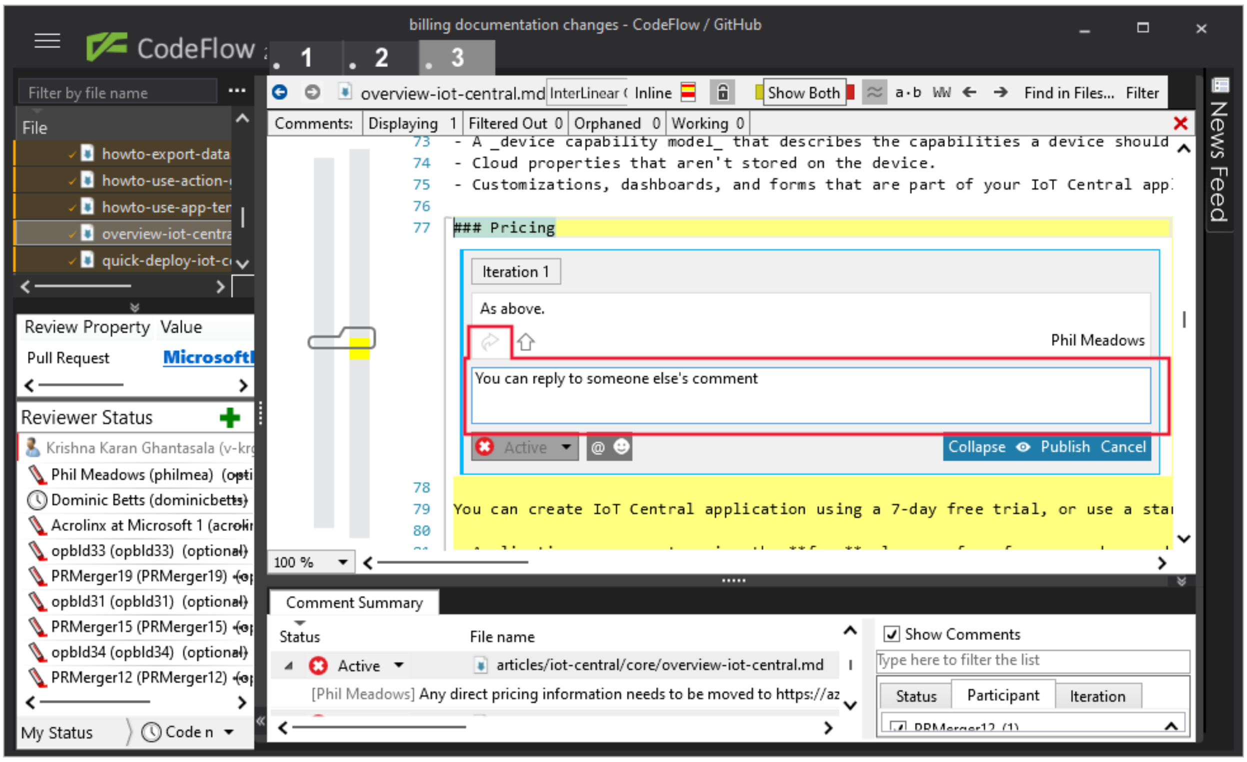
```
\\CodeFlow\public\cf.cmd openGitHubPR -webUrl https://github.com/MicrosoftDocs/azure-docs-pr/pull/16379
```

Manage comments

As the PR author, you can use the tools in CodeFlow to manage and track the comments added by your reviewers.

You can use the filters in the **Comment summary** area to find specific comments.

You can reply to comments made by your reviewers to request more information or a clarification:



Next steps

You can learn more about using CodeFlow to review GitHub PRs in Working with GitHub Pull Requests.

Acrolinx overview

7/13/2021 • 2 minutes to read

Acrolinx is editing software. It automates feedback on brand, terminology, grammar, spelling, punctuation, style, and voice. It's used by a partnership within Microsoft: Content & Learning (C&L), Microsoft 365 (MARVEL), Worldwide Learning (WWL), and the Business Applications Group (BAG). The goals are to help content contributors improve their writing and to make technical content on `docs.microsoft.com` clearer and more consistent.

- **GitHub**: You can use Acrolinx in GitHub. You get results when you submit a pull request (PR). Acrolinx runs automatically and writes results to the PR comments. Depending on your team, you must reach a minimum score before your PR is merged. For C&L, the minimum score is 80. You don't need to install anything to use Acrolinx in GitHub. If your team uses Acrolinx, they'll inform you of the process

- **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.

Criteria for using Acrolinx

Members of the partnership (C&L MARVEL, WWL, BAG) can add support to their repos since they've contributed to the budget. But their content must meet these criteria:

- Technical documentation for a Microsoft product.
- Conceptual in nature.
- In a GitHub repository and authored in Markdown.
- Published using the OPS publishing system.
- Published to `docs.microsoft.com`, `docs.microsoft.com/learn`, or to the official technical documentation library for a Microsoft product on another Microsoft-owned web property.
- Within the Microsoft Docs organization in GitHub. If the repository is in a different organization, a strong business justification must exist.
- The content team agrees to the minimum score set by their team.

Limitations:

- Acrolinx doesn't work with autogenerated reference content.
- Windows: Only these Windows technologies in C&L are covered by Acrolinx: Windows Developer (SDK), Windows Device Drivers (WDK), and Windows Server.
- Don't use Acrolinx for products outside the partnership because doing so violates our license.

Covered repositories

You can use Acrolinx in Visual Studio Code only in repositories that fit the criteria above and are part of the partnership at Microsoft:

- C&L.
- MARVEL
- WWL
- BAG

Onboarding Acrolinx

Request to enable Acrolinx on your repository.

Reporting bugs and false positive results

Report bugs and false positives at https://aka.ms/acrolinxbug. Issues are triaged once per week (and longer than that during holidays).

If you can't access the link above, request access to CEAPEX.

Acrolinx in Visual Studio Code

7/7/2021 • 4 minutes to read

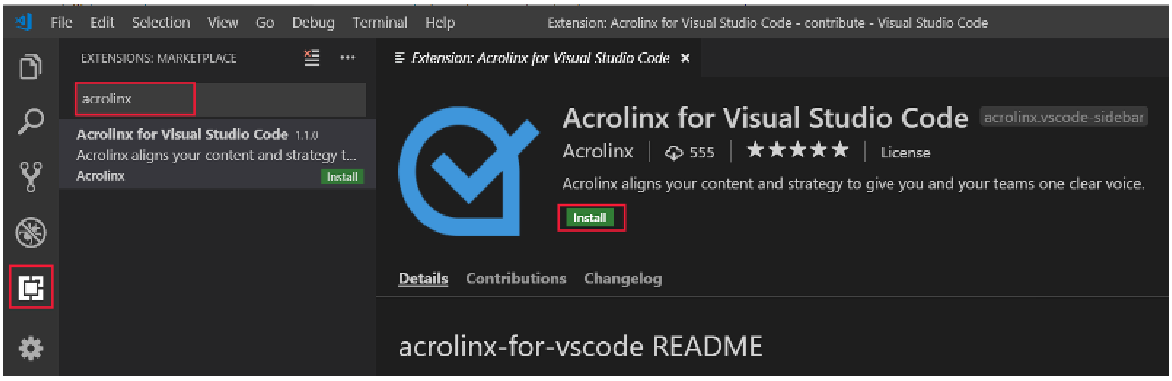
The Acrolinx extension for Visual Studio Code performs the same check that GitHub does: spelling, grammar, style, tone, clarity, and key terminology. This article explains how to install and run the Acrolinx extension in Visual Studio Code.

You can only use Acrolinx in Visual Studio Code in repositories that fit the criteria and are part of the partnership at Microsoft:

- Content & Learning (C&L)
- MARVEL
- Worldwide Learning
- Business Applications Group (BAG)

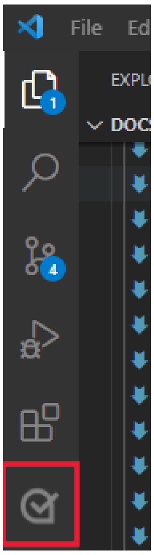
Install Acrolinx in Visual Studio Code

1. Start Visual Studio Code.
2. Update to the latest version if prompted.
3. Select the square icon in the left-hand bar to open the **Extensions** pane.
4. Type *acrolinx* in the **Search Extensions in the Marketplace** box, and select the **Acrolinx for Visual Studio Code** result.
5. Select the green **Install** button.



Configure Acrolinx

1. The first time you run Acrolinx, you need to configure the server details. To do so, open an *.md* or a *.txt* file to make the Acrolinx icon visible on the left sidebar:
2. Open an *.md* or a *.txt* file and select the Acrolinx icon from the left-hand menu:



An Acrolinx window appears, displaying the sign-in page.

3. For the **Acrolinx URL** paste in the following server address:

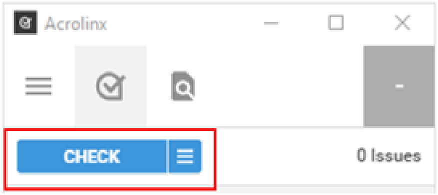
```
https://microsoft-ce-csi.acrolinx.cloud:443
```

4. Select **Connect** > **Sign In**.

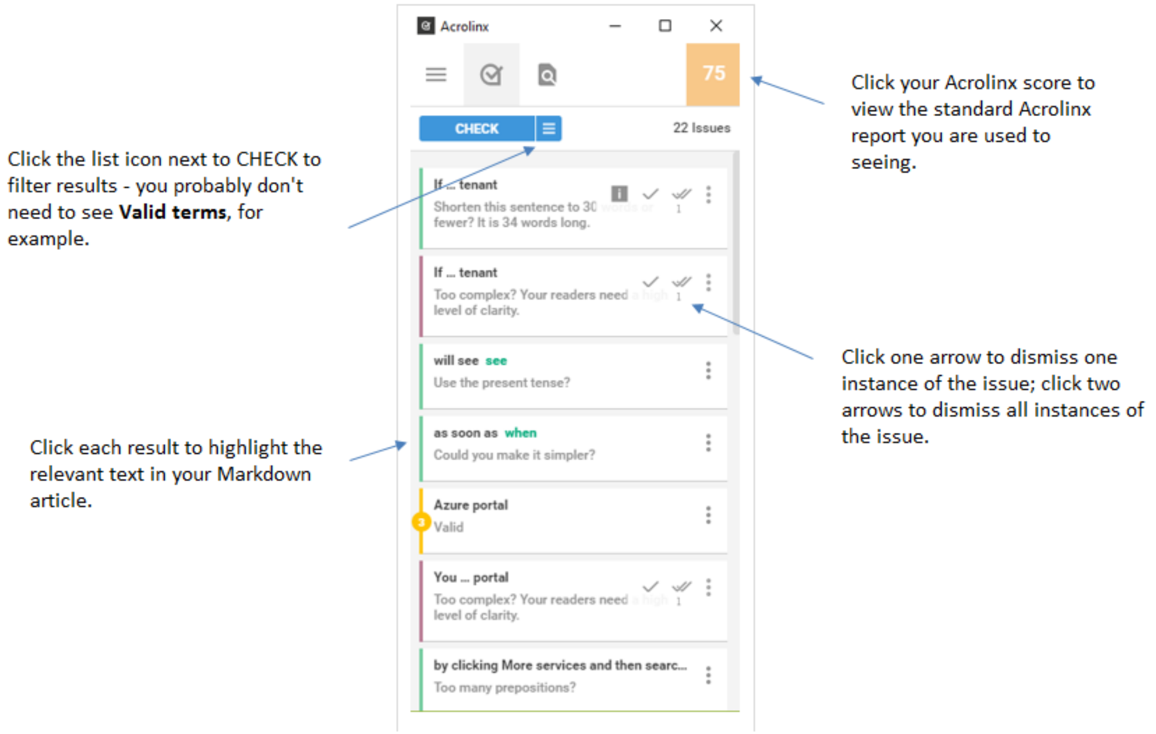
   Acrolinx uses Microsoft corpnet authentication, so sign in using your Microsoft credentials. As part of this process, you have to authorize Acrolinx for Visual Studio Code on the Acrolinx site - just follow the prompts.

5. Select **Options**. In the **Options** pane, select the correct guidance profile.

6. Select the **Check** button to check the document:



7. Review the results



Known issues with v1.1

- The Acrolinx extension version 1.1 for Visual Studio Code doesn't recognize Markdown files on a Mac. If you use a Mac, your only options are to rely on the GitHub integration, revert to version 1.0, or find a Windows computer.
- On Windows computers, you might receive a message that Acrolinx can't recognize the file type. The recommended workaround from Acrolinx is to select the Acrolinx sidebar "button" again before running a check. This needs to be done every time a different document is to be checked. These other tricks might also get you unblocked:
  - Open an XML file and try to run Acrolinx, then go back to the Markdown file and try again.
  - Change the editor layout to "split right" and back to "single."
- If you have multiple windows open, Acrolinx reports on the first window opened. To get it working, close all the editor windows that you don't want to analyze, then restart Visual Studio Code.

Install Version 1.0

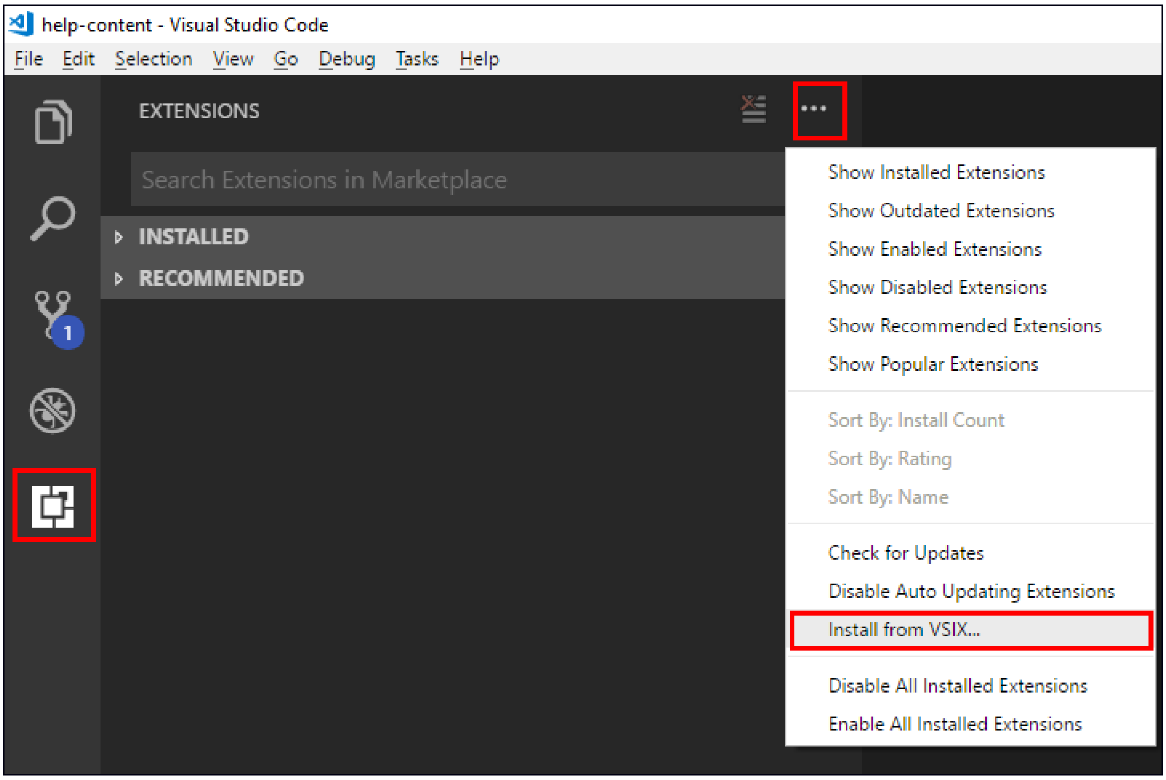
Prerequisite for Mac

On macOS, you need to copy the VSIX file locally using the SMB share before installation. To do this:

1. Open Finder and choose **Go** > **Connect to Server**.
2. Enter the **Server Address** of the file: `smb://dpspub/Acrolinx/`. Make sure you're connected to corpnet.
3. Enter your credentials in form `redmond\<alias>` (for `redmond` domain) and your password
4. Select **Connect**, and copy the file locally to your Mac.

Installation steps

1. Start Visual Studio Code.
2. Update to the latest version if prompted.
3. Select the square icon in the left-hand bar to open the **Extensions** pane.
4. Select the three dots for "More" and select "Install from VSIX..."



5. For Windows, paste the path `\\dpspub\Acrolinx\VisualStudioCodeV1.0.1_B87` into the **File** name box and select **Install**.

   On a Mac, locate the saved VSIX file and then select **Install**.

   > **NOTE**
   >
   > The extension is large and can take about a minute to install. There is no indication of activity during this time.

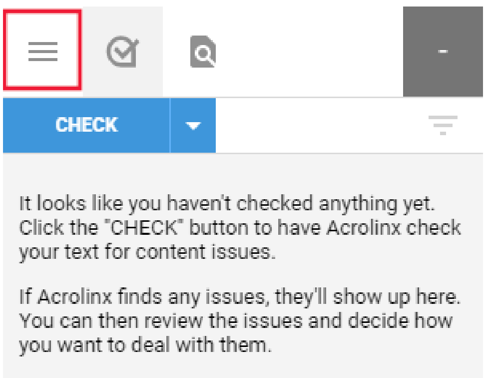
6. When installation is complete, Visual Studio Code may prompt you to reload. The first time you run Acrolinx, you need to configure it.

Change your guidance profile

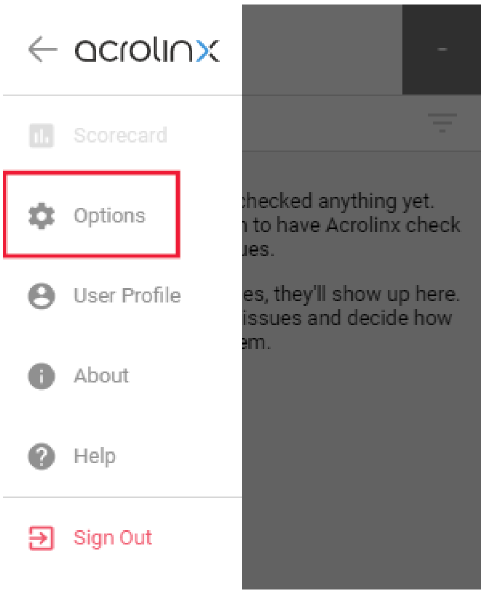
All content contributors must make sure they're using the correct guidance profile in Acrolinx. The guidance profile includes a specific term list and other guidelines that Acrolinx uses to score the content you're working on when you submit a pull request. Admins assign guidance profiles to each repo in GitHub.

If you work in repos for more than one content group, you must change your guidance profile each time you change orgs. Other guidance profiles may be added in the future.

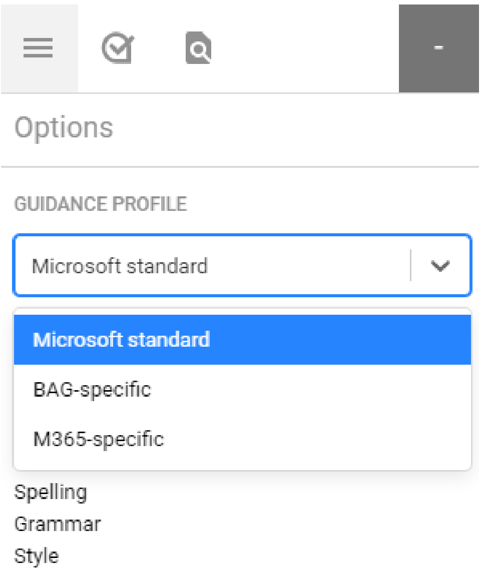
1. In Visual Studio Code, open an *.md* or a *.txt* file to make the Acrolinx icon visible on the menu bar. Open the Acrolinx sidebar and select the slide-out menu.



2. Select **Options**.



3. In the **Options** pane, select the correct guidance profile:



REPO	GUIDANCE PROFILE
C&L	Microsoft standard
MARVEL	M365-specific
BAG	BAG-specific

Report issues, bugs, or false positives

- Check the troubleshooting guide
- File a bug or suggestion using this link: https://aka.ms/acrolinxbug

Guidance for using Acrolinx and respecting the license

Use the Acrolinx extension only in repositories that are licensed for Acrolinx under the Content & Learning-Acrolinx contract.

- When you run Acrolinx locally using Visual Studio code, run it only on local repository clones for covered repositories.
- The path on your local computer must include the correct name of the authorized repository. For example, if you have a local clone of azure-docs-pr, the path on your computer must include "azure-docs-pr". The path is necessary for license compliance and accurate data on Acrolinx usage and scores. For repositories that have non-unique names ("docs", for instance), your local clone must be in a local path that includes the organization. The ASP.NET and .NET documentation repositories must be cloned into a location that includes aspnet/docs or dotnet/docs in the local file path.
- Don't put content from an unauthorized repo into an authorized repo just to get Acrolinx results. This screws up the data around scorecards and scoring for content in the repository you use to do this. It is outside the licensing agreement, and it corrupts the usage data.

Minimum Acrolinx scores

5/10/2021 • 3 minutes to read

For **Content & Learning**, in the repositories contributors work in, the pull request reviewers check all pull requests that require human review. They make sure each article in a pull request meets the minimum Acrolinx score before they merge the changes. All pull requests that require human review in C+L must meet the minimum Acrolinx score.

Note: Min score requirements for M365 and BAG will be added once we are all working from the same Acrolinx instance, scheduled for the beginning of November.

How the minimum score works for C+L

- The starting minimum score for C+L is 80. All the articles in a PR need to meet the minimum score for the PR review team to merge the PR.
- The minimum score applies on a going-forward basis - as you update content within your normal workflow, you will need to bring each article up to the minimum score (higher is better!). Everyone should address the spelling, tone of voice, and clarity feedback in Acrolinx reports. This feedback from Acrolinx is designed to make our content more trustworthy, easier to understand, and with a more conversational tone. The Acrolinx recommendations directly align with Microsoft's modern voice principles. We're doing this for our customers' benefit.
- You do have to fix pre-existing content as you make updates.

Dealing with false positive results

- We use the term "false positive" to refer to cases where Acrolinx flags something as incorrect or needing a fix, but where you know that the usage is correct for the context. The minimum score is intended to accommodate for some false positives because we have to live with some false positives. It's not possible or even desirable to add every term that generates a false positive to the system. You can request that terms be added when the terms are used broadly in a content set.
- **For Content & Learning,** if lots of false positives in a pull request scorecard put the score of an article below 80 and cause your PR to be blocked, take these steps:
  - File a bug for the false positives using this link: https://aka.ms/acrolinxbug. Make sure you include a link to the PR, a link to the scorecard, and a clear description of why the flags are false positives
  - Enter a comment in the pull request to let the PR reviewers know that a false positive in the results is causing the PR to have a score lower than 80. Include a link to the Acrolinx bug. As long as you have filed an Acrolinx bug for the false positives, the PR reviewers can review the PR against all other criteria and merge as appropriate.

Exceptions to the minimum score requirement

- **Bulk updates**: Bulk updates are exempt from the minimum score requirement. Examples of bulk updates include changing the author metadata across a group of articles or changing a product name across all the articles in a content set.
- **Localization fixes**: When a member of the localization team submits a fix, the localization PM is not responsible for updating the article to meet the Acrolinx min score requirement - that is outside the scope of their role. Usually, the comment in the PR makes it clear the change is a localization fix, and the

contributor's profile usually indicates they are affiliated with the localization team.

- **Emergency publishing/updates**: If you've been asked to make a high priority fix to an article to address a severe customer issue, or an executive, red zone, security, or privacy issue, you can request an exception to the score requirement if the issue is truly urgent and you cannot dedicate time to the fixes. This guidance isn't about false positives – this is about not fixing valid Acrolinx flags. (The process for handling false positives is described earlier in this article.) The content lead needs to approve the exemption. This will be handled on a case-by-case basis.

Acrolinx reports

7/19/2021 • 2 minutes to read

The Acrolinx report has been restored to the SkyEye site thanks to work by the CGA team and Acrolinx. This report gives scores and related details for articles with pull requests. To access the report, go to https://aka.ms/skyeye and select the **Acrolinx Report** tile.

What's in the Acrolinx report

All content that has been run through the GitHub integration should be in the report. VS Code doesn't write results to the report. So, the report contains no info for repos that are public only and where VSCode is the only Acrolinx tooling.

Baseline reports

Baseline reports give Acrolinx scores for all articles in a repo, not just articles with recent pull requests. This report is useful if you're evaluating the quality of an existing repo that you inherited from another team.

- Baseline reports are available only for private repos currently.
- The reports are run automatically when Acrolinx support is added to a repo.
- You can request a baseline report for a repo with existing Acrolinx support.

To request a baseline report file a request at https://aka.ms/acrolinxbug.

Questions

If you have questions, contact monicar.

Troubleshooting issues with Acrolinx

6/24/2021 • 2 minutes to read

If you have problems authenticating to Acrolinx or if you receive an error message, this info will help you resolve the issue.

Authentication

- **When you select a scorecard in GitHub or VS Code for the first time**, you need to authenticate as a Microsoft user. You'll be directed to the Acrolinx site and asked to sign in. Just use your Microsoft alias and select **Sign in**.
- **Occasionally you have to authenticate again**. This error can happen if you change to another repo or browser. Just sign in again.
- **If you still can't authenticate**, file a bug here: https://aka.ms/acrolinxbug

Outages

- If Acrolinx is down, file a ticket at the SiteHelp portal or sign in to the Docs Support Channel in Teams. (In the SiteHelp portal, select **Submit request or issue** > **Select Service** > **OPS** > **Select Service Category** > **Acrolinx down or Acrolinx not accessible**)

Error messages

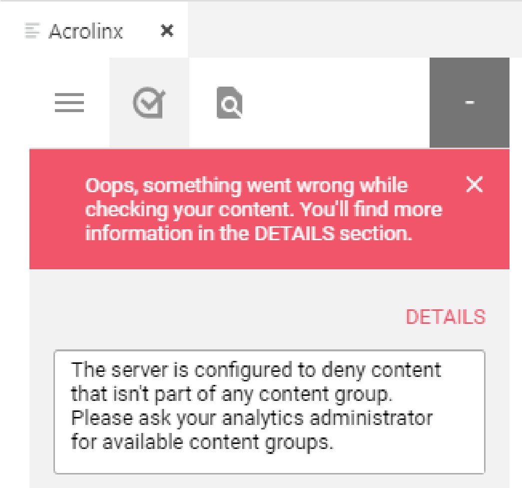
The most common error message appears if you're trying to run Acrolinx on content that doesn't have Acrolinx support. The error may also appear on content that isn't in the correct local path when using Acrolinx in VS Code.

When you're using Acrolinx in VS Code, after you select the **Check** button, you might receive the following message:

> *Oops, something went wrong while checking your content. You'll find more information in the DETAILS section.*

When you select **DETAILS**, you get the following additional information:

> *The server is configured to deny content that isn't part of any content group. Please ask your analytics administrator for available content groups.*



This problem occurs because you're trying to run Acrolinx on content that is not covered by the Acrolinx program. You can only use Acrolinx in VS Code in repositories that fit the criteria and are part of the partnership at Microsoft:

- Content & Learning
- MARVEL
- Worldwide Learning
- Business Applications Group

To resolve the problem:

- If you're working on content for a listed repository, you need to work on it within your local clone of the repository. The local clone's name must match the name of the upstream repository.
- If you're trying to run Acrolinx on content for a repository outside the Acrolinx program, you need to stop. Acrolinx can only be used for content in a covered repository. See the Acrolinx in Visual Studio Code article for details about how to use Acrolinx in alignment with our license.
- It's possible that you're using Acrolinx for content in a repository where Acrolinx is configured. The content group for your repository might be missing or have a problem. For this case, file a bug using https://aka.ms/acrolinxbug.

Twitter App (aka.ms/tweet) (Preview)

6/24/2021 • 6 minutes to read

The Twitter App allows for the simplified use and management of public-facing Twitter accounts. Once a Twitter account is added to the app by the account manager, anyone from the company can propose tweets. Doing so adds the tweet to the account manager's **Tweet queue**, letting the account manager decide which tweets publish to Twitter.

This app allows multiple people to tweet from the same Twitter account without sharing credentials, but still controls all tweets in a single person's hands. You can use this app for your own personal Twitter account or use it for larger accounts, such as @docsmsft or @SQLDocs.

> **IMPORTANT**
>
> The strength of this app is its simplicity, along with both Microsoft and DevRel-specific feature integration. However, Twitter accounts with more than 1000 followers should also be using Sprinklr, a broader and more complex feature set.

First-time setup

Add a Twitter account to allow other users to tweet to it, pending your approval. Nothing is posted to this account without your consent. You can remove the account at any time either in the app or via your Twitter third-party application permissions.

The account that you're currently signed into Twitter with is the account added to the app. If you're already logged into an account, but want to set up a different one, then you need to go to Twitter, sign out of the current account, and log into the account you want to add.

Follow these steps to add a Twitter account to the Twitter app:

1. Navigate to the **Management portal** from aka.ms/tweet after signing in. You can also select **Account** from the top navigation menu.
2. Select the **Sign-in with Twitter** button and follow the screens to allow the TwitterApp to access your Twitter account.
3. Select **Return to portal** once your account has been added successfully.

> **NOTE**
>
> The account that you're currently signed into Twitter with is the one that gets added to the portal. If you need to add a different account, be sure to sign directly into Twitter, sign out of the one account, and then sign into the account you want to add to the app.

Account Management

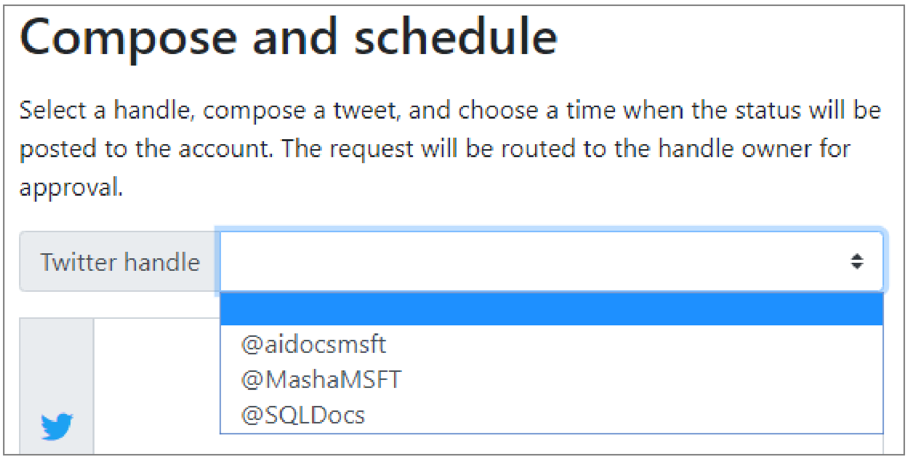
As a Twitter account manager, you first need to configure your Twitter account for first use. After that, you can approve, edit, and delete tweets using the tweet queue.

To manage your twitter account, expand **Settings** from the **Management portal**.

Here you can set your account to be either public or private, and enable automatic retweets of mentions.

Privacy

Setting your account to private means that only you can tweet to that Twitter handle, but setting it to public means that anyone can tweet to the private handle from within the app. Only your own private account and accounts set to public are visible in the **Twitter handle** drop-down in the Tweet portal:



How to compose a tweet

Follow the steps below to tweet to any available account:

1. Sign in to your Twitter account.
2. Navigate to the **Tweet portal** from aka.ms/tweet.
3. Select the Twitter account you want to use from the drop-down.
4. Select a date and time you like your tweet to go live (optional).
5. Compose a tweet and then select **Create Tweet** to add it to the account manager's tweet queue.

> **NOTE**
>
> The character limit matches that of Twitter. You have 280 characters per Tweet. URL character counts are calculated at a constant value of 23 characters per URL regardless of the URL length. This means that a URL of 500 characters long is perfectly valid in a Tweet, as it only counts as 23 characters. You may also have multiple URLs in a Tweet. If you have two URLs with a single space between them, your Tweet would be 47 characters long.

Tweet guidelines

Add a URL in a tweet

If you are adding a URL to your tweet, then please follow the guidelines listed below.

1. Remove the locale (ex: en-us) from the URL.
2. Add one of the following dashboard extensions to the URL. The dashboard extensions are used to help collect data.
   a. For @docsmsft add: ?WT.mc_id=docsmsft-twitter
   b. For @WindowsDocs add: ?WT.mc_id=windowsdocs-twitter

For example: https://docs.microsoft.com/rest/api/?WT.mc_id=windowsdocs-twitter

Evangelize your tweets

- Tag any appropriate PM or team that could help amplify the tweet.
- Use relevant hashtags.
- Include an image if possible.

Ideas to tweet about and some sample tweet formats:

Some sample ideas to tweet.

- New or updated topic on cool tech.
- An interesting blog post.

- Quote tweeting an existing tweet.
- Poll about what more info users want.
- An interesting piece of history about our technology.
- A question about how devs are using technology.
- Ideas on how to use UI design.
- Tiny 4-6 step tutorials in a series of tweets.

Here are some sample tweets.

- "Did you know you can do a cool thing with X? Here's how..."
- "Great news! X just got updated - now it has Y, and here's a list of all the other new stuff."
- "Have you ever found yourself wishing for X? Lucky you - we just released X. Here's more information."
- "This is how to do X, which is a cool thing, in four steps."
- "What do you think of X? How could we improve it?"

For more information about tweet submissions, visit SocAmp Submssions Onenote

For more information about guidlines, visit Social Media Guidlines Onenote.

Setting a time to send a tweet

If you leave the date blank, the tweet goes live as soon as the account manager approves it.

If you provide a date and time, the tweet goes live at the designated time frame, as long as it's been approved by the account manager.

The account manager for a specific handle can edit your tweet.

If you @mention another account that's been added to the app (such as @sqldocs), and that account has the retweet mentions feature enabled, your tweet is added as a retweet to that account managers queue. The account manager can't edit it.

Retweet mentions

Twitter gives you the ability to mention another Twitter handle, such as @sqldocs. If you enable automatic retweet of mentions, then anytime *anyone* mentions your twitter account (either through the app or through Twitter), that tweet is added to your queue as a retweet. Since this functionality uses Twitter's underlying retweet API, these tweets are uneditable. When you approve the tweet, it shows up as a retweet on your Twitter page and displays the original user's Tweet.

For example, this screenshot shows the @sqldocs account retweeting @MashaMSFT's tweet, who mentioned @sqldocs in the tweet:

The original tweet and the retweet were done through the Twitter App.

Tweet queue

The queue in the **Management portal** is where the Twitter account manager can approve, edit, or delete tweets. Select **Tweet Queue** from the top navigation menu to access the queue for all of the Twitter accounts associated with your TwitterApp account.

Remarks

- You can approve a tweet by selecting the **Approve** button.
- You can edit the content of a tweet by selecting the **Edit** button.
- Once a tweet is approved, it goes live immediately or at its scheduled time.
- Use the trash can to delete a tweet, either before or after it's been approved.
- You can't currently modify the date and time of a scheduled tweet.
- You can't edit the content of a retweet.

Sprinklr

Sprinklr is the platform of record for managing Microsoft's social media accounts and is required for all publishing and engagement for any social account with more than 1,000 fans/followers.

Sprinklr has a robust feature set and is geared towards social media interaction and engagement. You can schedule tweets, follow trends, interact with customers, and run comprehensive social media presence analysis.

Although Sprinklr allows multiple people to tweet to the same account, it doesn't have the same unique request/approval process as the Twitter app. Additionally, there are plans to integrate GitHub functionality into the Twitter app, such as automatically tweeting when adding new articles and other DevRel specific things. However, the TwitterApp and Sprinklr aren't mutually exclusive and can be used in conjunction, with Sprinklr bringing added complex functionality.

Contact

You can contact the SocAmp V-team team for questions and comments.

- Twitter App info
- Feedback / feature request.
- See what's currently in development via the DevOps dashboard.

- Email: TwitterApp@microsoft.com

Next steps

- Submit feedback and feature updates.

Banners on docs.microsoft.com

7/7/2021 • 4 minutes to read

Banners are used on `docs.microsoft.com` to promote events or announcements. They're limited to a specific set of documentation related to the notice or promotion, with the exception being large-scale events like Ignite and Build. These banners have been proven to be successful in reaching Docs users.

This article details the guidelines for when a banner may be used and instructions for requesting a new banner. Requests are approved and processed by the PM team, and then the Dev team creates the banner.

Primary uses and scenarios

Banners are run at the discretion of the Docs product team, and every request will require approval. These situations and events are the primary uses for banners on `docs.microsoft.com`:

- Emergency situations (for example, Security issues)
- Tier 1 events (for example, Build or Ignite)
- Marquee/Learn TV events (for example, events run by DevRel or cloud advocates)

How to fill out the banner request form

To submit a request, fill out the banner request form at aka.ms/BannerRequest. Read through the instructions on the form carefully and be as specific as you can when filling out the form.

> **IMPORTANT**
>
> The minimum lead time for banners is 10 days prior to the desired go-live date. Requests must be submitted at least 10 days before the banner is expected to go live. This gives the PM team the necessary time to review the request and pass it on to the Dev team, if approved.

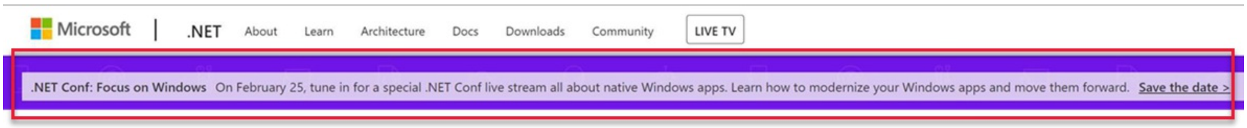
You'll need the following information to fill out the form:

- Name of requester.
- Reason for banner.
- Banner information:
  - Date/time: Provide the date/time that the banner should go live and the date/time it should end.
  - Copy title: Provide the title of the banner.
  - Copy description: Provide the text of the banner.
  - CTA text: At the end of the banner text, there is a CTA (Call To Action) link for users to select. In this field, provide the text of the link.
  - CTA destination URL: Provide the URL that the link should go to.
- All Docs/Learn URL paths where the banner should be displayed.
- Any additional questions, comments, or information that you want to communicate.

After you've finished filling out the form, click **submit**. The form will send an email to the PM team in charge of banners.

Example banner and form entries

An example of a typical Learn TV event banner and the banner information that was submitted for this request:



- **Copy title:** .NET Conf: Focus on Windows
- **Copy Description:** On February 25, tune in for a special .NET Conf live stream all about native Windows apps. Learn how to modernize your Windows apps and move them forward.
- **CTA text:** Save the date
- **CTA destination URL:** aka.ms/TrackingLinkExample

Timing of banners

Allow two full business days after a request is submitted to hear back from the PM team. If the request was received before the required 10-day lead time, the Dev team will meet that deadline.

Depending on the type of promotion, and at the discretion of the product team, a banner will typically be live on the site for 2-4 weeks.

> **NOTE**
>
> If the request is submitted after the 10-day lead time of when the banner will go live, please submit the form and contact Isa Hoban (ihoban) immediately. The PM team will do their best to meet the deadline, but cannot guarantee the request will be fulfilled if the form is not submitted 10 days prior to the desired go-live date.

Evaluating banner performance

For banners with a CTA, use a specific tracking URL so you can observe how the links from `docs.microsoft.com` performed compared to other referrers to the destination page. You'll share this information with the PM team after the banner is removed and be tagged in an Azure DevOps work item, along with instructions for the data you're expected to report.

FAQ

Can I include images?

No. Specialized banners are currently reserved for Tier 1 events. Otherwise, expect a single-colored banner with black text.

Can I use a banner to promote on-demand content following an event?

No. We don't approve the use of banners for promoting on-demand content following an event. The team is beginning discovery on potential new features for notifications and promotions within a docset, outside of banner usage, and these may be a more appropriate option for promoting on-demand content in the future.

Are banners persistent or dismissible? Will a banner reappear after being dismissed?

Banners for events must be dismissible. If a user selects the banner close button, the banner will be hidden, and the user action is stored in browser local storage. This means that the banner won't reappear unless a user removes the key "dismissed-banner" from local storage.

Are banners localized?

Yes, if needed, the banners will be localized.

Can I run a second banner on a page that already has a banner?

No. Two event banners are never run at the same time. There are some cases where an event banner may appear with a security/emergency banner, but these will be decided and prioritized at the discretion of the PM team.

Questions and feedback

Contact Isa Hoban (ihoban) for further questions or concerns.

Use AutoHotkey automation for Git Issues and DevOps templates

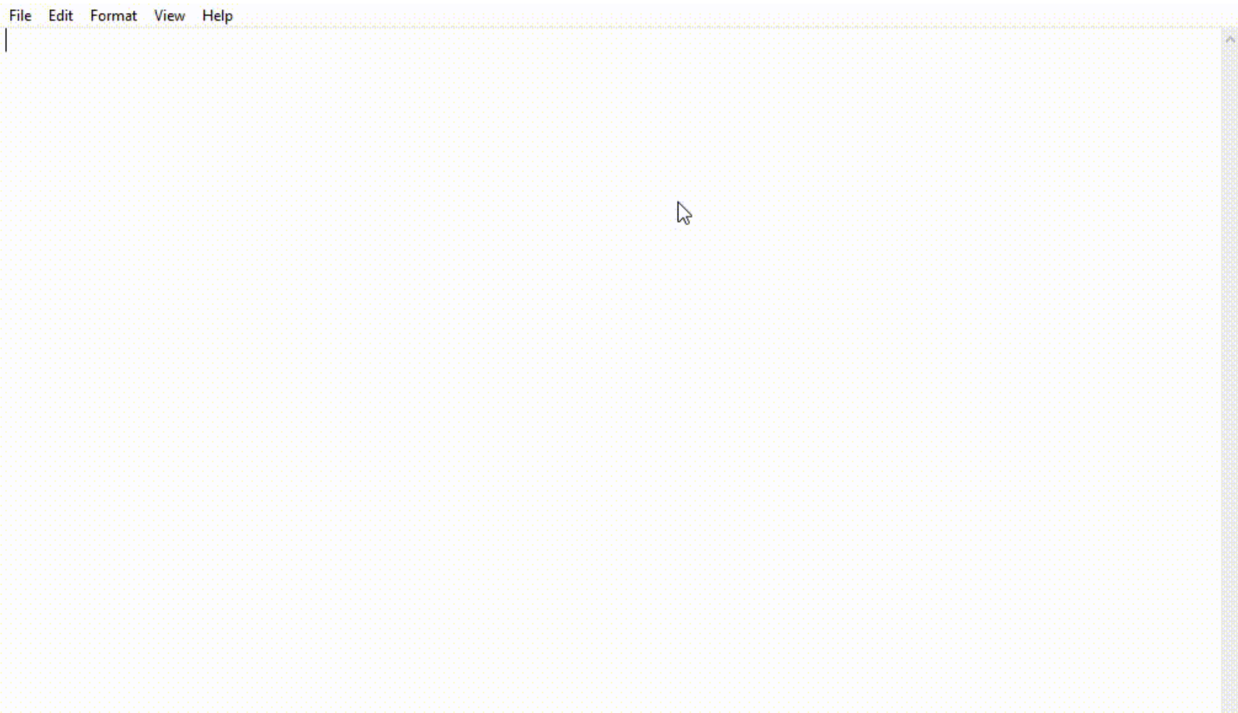
6/24/2021 • 23 minutes to read

AutoHotkey is a third-party OSS approved open-source automation tool that allows you to create templates from self-selected "keywords". The keywords are then automatically replaced with predefined text blocks and/or scripts in most text editors and browsers.

This topic describes how to use AutoHotkey to create automatic templates for responding to customers via Git Issues, and for populating Azure DevOps items. However, this tool can be used within most text editors, and scripts can be added manually to suit any other needs.

This topic is meant to be used as an introduction and foundation for configuring preliminary scripts. The scripts provided in this article are **completely modifiable**, and can be edited to suit your needs.

Video Example



When to use

AutoHotkey is an automation tool that facilitates the output of canned saved text. It can be used within anything that accepts text input, such as Azure DevOps items, web browsers, Word, Notepad, Notepad ++, etc. (except VS Code). We recommend using it to reply to Git Issues, and populate the "Description" section of an Azure DevOps item, such as within Azure Boards. This article is meant to be just a starting point to introduce you to AutoHotkey. You can create any type of script you want that suits any need you may have.

The scripts and Git Issue replies listed in this topic are just suggestions - they are by no means mandatory to use in either Git Issues or Azure DevOps items. However, they are likely to save you time and increase your efficiency.

How is AutoHotkey different?

- **VS Code / Docs authoring pack templates**: It is different than a VS Code template in that those are built into VS Code and just populate templates based on the content model, such as a quickstart. Autohotkey is not tied to a single program, and it populates text based on what you've configured it to.

- **Azure Boards Templates**: Templates used in Azure Boards create work items, and fill out fields such as the area and iteration path. AutoHotkey can be used to populate text within an Azure Board item, but it does not create new work items.

Installation

To use AutoHotkey, do the following:

1. Download and install AutoHotkey from: https://autohotkey.com/download/.
2. Create a new text file and change the extension from `.txt` to `.ahk`.
3. Copy various scripts from this topic into your `.ahk` file - modify keywords and templates as necessary. These templates were originally created for the SQL docs team so they've been generalized with underscores. You can either pick and choose individual script snippets, or you can copy the entire full script located at the end of the article.
4. Save the `.ahk` file.
5. Place a copy of the `.ahk` in your startup folder located at `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp`. Security settings may prevent you from making changes to this file in this location, so you may need to have a working copy elsewhere that you use to replace the file in the startup folder after making changes to it.
6. Double-click the `.ahk` file to initiate the script - this step only needs to be done if you want to manually initiate the script (such as the first time you configure this, or if you're changing and testing the script). Otherwise, this step is done automatically for you each time your computer starts up as long as the .ahk file is in the startup folder.
7. Type in any of the keywords in any text editor to test the script. Unfortunately, AutoHotkey does not work in VS Code - for that, you'll need to create snippets within VS Code.

Azure DevOps scripts

Description section

The following script can be used as a template for the description section in an Azure Board item.

KEYWORD	OUTPUT
vnotee	Status ====== To do ====== Done ====== Git ====== Contacts ====== Details ======

Place the following script in your .ahk file to enable this keyword script:

```
:r0:vnotee::
(
Status
======

To do
======

Done
====

Git
====

Contacts
========

Details
=======
)
```

Discussion section

The following script can be used as a title for notes appended in the Discussion section. It will populate the current date and the word sprint - you'll need to manually add which sprint it is.

KEYWORD	OUTPUT
dnotee	`2019-05-01 | Sprint`

Place the following script in your .ahk file to enable this keyword script:

```
:r0:dnotee::

FormatTime, Time,, yyyy-MM-dd
Send %Time%
send {Space}| Sprint
return
```

Random tasks DevOps item

This is a useful template for a single monthly "Random Tasks" Azure Boards item meant to track Public PRs, Git Issues, and random tasks that are not substantial enough for their own user story. The top "To do" portion shows the tasks, Public PRs, and Git Issues still pending. Once an item is complete, it's moved down to the "Done" section.

KEYWORD	OUTPUT
rnotee	

KEYWORD	OUTPUT
====== To do ====== Tasks ------ 1. 2. 3. Public PRs ----------- 1. 2. 3. Git Issues ---------- 1. 2. 3. ====== Done ====== Tasks ------ Public PRs ----------- Git Issues ----------	

Place the following script in your .ahk file to enable this keyword script:

```
:r0:rnotee::
(
====================================================
                    To do
====================================================

Tasks
------
1.
2.
3.


Public PRs
-----------
1.
2.
3.


Git Issues
----------
1.
2.
3.


====================================================
                   Done
====================================================

Tasks
------

Public PRs
-----------

Git Issues
----------
)
```

Git Issue template scripts

For a new Git Issue

KEYWORD	OUTPUT

KEYWORD	OUTPUT
ishnew	Thanks for contacting Microsoft with feedback about our product documentation. Depending on the complexity, it could take from a few business days to a few weeks to look further into your question, issue, or suggestion. We'll contact you if we need further information, and you'll receive an email message each time we "@mention" you in a comment. If you're suggesting a simple change to the documentation, it might be faster to submit the suggestion yourself by creating a GitHub pull request (PR). Here's how: 1. On the page you have feedback about, select the **Edit** link at the top right. 1. On the next page, select the **Pencil** icon at the top right. 1. On the next page, in the **Edit file** text window, make your edits directly to the text you want to change. If you need help with formatting the new or changed text, see our Markdown Cheatsheet After you've made your edits, under **Commit changes**: a. In the first text box, enter a brief description of the change you've made. b. In the **Add an optional extended description** box, paste the link to your GitHub issue. 1. Select **Propose file change**. 1. On the **Comparing changes** page, select **Create pull request**. 1. On the **Open a pull request** page, select **Create pull request**. 1. On the **Open a pull request** page, select **Create pull request**. Allow a day or more for the article's author to review and approve your change or offer an alternative solution. By opening a pull request, you're helping the author add your approved change directly to the documentation. After the updated article is published, GitHub lists you as an article contributor. If your issue is more complex than a simple change in the documentation, that's okay. We'll continue to research your suggestion and then get back to you with our response. Thanks again for submitting your feedback. Your suggestions help improve our documentation.

Place the following script in your .ahk file to enable this keyword script:

```
:r0:ishnew::
(
Thanks for contacting Microsoft with feedback about our product documentation. Depending on the complexity,
it could take from a few business days to a few weeks to look further into your question, issue, or
suggestion. We'll contact you if we need further information, and you'll receive an email message each time
we "@mention" you in a comment.

If you're suggesting a simple change to the documentation, it might be faster to submit the suggestion
yourself by creating a GitHub pull request (PR). Here's how:

1. On the page you have feedback about, select the **Edit** link at the top right.
1. On the next page, select the **Pencil** icon at the top right.
1. On the next page, in the **Edit file** text window, make your edits directly to the text you want to
change.
    If you need help with formatting the new or changed text, see our [Markdown Cheatsheet]
(https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet).
1. After you've made your edits, under **Commit changes**:
    a. In the first text box, enter a brief description of the change you've made.
    b. In the **Add an optional extended description** box, paste the link to your GitHub issue.
1. Select **Propose file change**.
1. On the **Comparing changes** page, select **Create pull request**.
1. On the **Open a pull request** page, select **Create pull request**.

Allow a day or more for the article's author to review and approve your change or offer an alternative
solution.

By opening a pull request, you're helping the author add your approved change directly to the documentation.
After the updated article is published, GitHub lists you as an article contributor.

If your issue is more complex than a simple change in the documentation, that's okay.  We'll continue to
research your suggestion and then get back to you with our response.

Thanks again for submitting your feedback. Your suggestions help improve our documentation.
)
```

For a Git Issue intended for support

KEYWORD	OUTPUT
ishport	Thanks for contacting Microsoft about your issue. As we understand it, you're describing a problem with the product itself and not with the product documentation. To get assistance or support for your product issue, we recommend that you engage with the product community or open a ticket with Microsoft Support. For more information, see SQL Server help and feedback. To provide product suggestions or ideas for improvement, go to https://aka.ms/sqlfeedback. If you're submitting feedback about the product documentation, please reply to this comment to clarify your issue further. Otherwise, we'll proceed with closing out this Git issue within a few business days. We're sorry that the SQL Docs team was unable to assist you further with your issue, but we appreciate your reaching out to us. If you experience other issues in the future, we encourage you to contact us again.

Place the following script in your .ahk file to enable this keyword script. Replace the underscores with values that apply to your specific team:

```
:r0:ishport::
(
Thanks for contacting Microsoft about your issue. As we understand it, you're describing a problem with the product itself and not with the product documentation.

To get assistance or support for your product issue, we recommend that you engage with the product community or open a ticket with Microsoft Support. For more information, see __________________.

To provide product suggestions or ideas for improvement, go to ___________.

If you're submitting feedback about the product documentation, please reply to this comment to clarify your issue further. Otherwise, we'll proceed with closing out this Git issue within a few business days.

We're sorry that the _________ Docs team was unable to assist you further with your issue, but we appreciate your reaching out to us. If you experience other issues in the future, we encourage you to contact us again.
)
```

To close a Git Issue

KEYWORD	OUTPUT
ishclose	Thanks again for contacting Microsoft. We've submitted a documentation change based on your suggestion, and the updated article should be live by tomorrow. Please let us know if there’s anything more we can do for you. To follow up on this issue, leave us a comment with an @mention.

Place the following script in your .ahk file to enable this keyword script:

```
:r0:ishclose::
(
Thanks again for contacting Microsoft. We've submitted a documentation change based on your suggestion, and the updated article should be live by tomorrow. Please let us know if there’s anything more we can do for you. To follow up on this issue, leave us a comment with an @mention.
)
```

Templates for editing content

Editing a doc in azure-docs-pr

KEYWORD	OUTPUT

KEYWORD	OUTPUT
azprpr	It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following: 1. Navigate to the page of interest 2. Select "Edit" button at the top right 3. Modify the URL so that 'azure-docs' becomes 'azure-docs-pr' 4. Press enter to navigate to the newly-changed URL and reload the page 5. Select the pencil to edit the document 6. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet 7. Scroll down to the bottom, name your file change, and select "Propose file change" 8. Select "Create pull request" on the 'Open a pull request' page 9. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link within the PR 10. Once you're satisfied with your changes, either type #sign-off in the comments to start the merge/publish process, or if you'd like me to review the changes, then type @MashaMSFT in the comments so that I can review and sign off If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs organization. More information can be found here: https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master Feel free to ping me directly on Teams or over email if you run into any issues.

Place the following script in your .ahk file to enable this keyword script - modify the underscore to your GitHub alias:

```
:r0:azprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following:

1. Navigate to the page of interest
2. Select  "Edit" button at the top right
3. Modify the URL so that 'azure-docs' becomes 'azure-docs-pr'
4. Press enter to navigate to the newly-changed URL and reload the page
5. Select the pencil to edit the document
6. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
7. Scroll down to the bottom, name your file change, and select "Propose file change"
8. Select "Create pull request" on the 'Open a pull request' page
9. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link within the PR
10. Once you're satisfied with your changes, either type #sign-off in the comments to start the merge/publish process, or if you'd like me to review the changes, then type @______ in the comments so that I can review and sign off

If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs organization. More information can be found here:
https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master

Feel free to ping me directly on Teams or over email if you run into any issues.
)
```

Editing a doc in sql-docs-pr

KEYWORD	OUTPUT
sqlprpr	It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following: 1. Navigate to the page of interest 2. Select the "Edit" button at the top right 3. Modify the URL so that 'sql-docs' becomes 'sql-docs-pr' and 'live' becomes 'master' 5. Press enter to navigate to the newly-changed URL and reload the page 6. Select the pencil to edit the document 7. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet 8. Scroll down to the bottom, name your file change, and select "Propose file change" 9. Select "Create pull request" on the 'Open a pull request' page 10. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link within the PR 11. Once you're satisfied with your changes, either type #sign-off in the comments to start the merge/publish process, or if you'd like me to review the changes, then type @MashaMSFT in the comments so that I can review and sign off If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs organization. More information can be found here: https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master Feel free to ping me directly on Teams or over email if you run into any issues.

KEYWORD	OUTPUT

Place the following script in your .ahk file to enable this keyword script - modify the underscore to your GitHub alias:

```
:r0:sqlprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following:

1. Navigate to the page of interest
2. Select the "Edit" button at the top right
3. Modify the URL so that 'sql-docs' becomes 'sql-docs-pr' and 'live' becomes 'master'
5. Press enter to navigate to the newly-changed URL and reload the page
6. Select the pencil to edit the document
7. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
8. Scroll down to the bottom, name your file change, and select "Propose file change"
9. Select "Create pull request" on the 'Open a pull request' page
10. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link within the PR
11. Once you're satisfied with your changes, either type #sign-off in the comments to start the merge/publish process, or if you'd like me to review the changes, then type @______ in the comments so that I can review and sign off

If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs organization. More information can be found here:
https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master

Feel free to ping me directly on Teams or over email if you run into any issues.
)
```

Editing a doc in a public repo

KEYWORD	OUTPUT
pubprpr	It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following: 1. Navigate to the page of interest 2. Select the "Edit" button at the top right of the page 3. Select the "Pencil" icon on the right 4. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet 5. Name your file change and provide a description, if necessary 6. Select "Propose file change" 7. Select "Create pull request" on the 'Comparing changes' page 8. Select "Create pull request" on the 'Open a pull request' page 9. Feel free to type @MashaMSFT in the comments so I can review the change as well if I'm not the author of the page Feel free to ping me directly on Teams or over email if you run into any issues.

Place the following script in your .ahk file to enable this keyword script - modify the underscore to your GitHub alias:

```
:r0:pubprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following:

1. Navigate to the page of interest
2. Select the "Edit" button at the top right of the page
3. Select the "Pencil" icon on the right
4. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
5. Name your file change and provide a description, if necessary
6. Select "Propose file change"
7. Select "Create pull request" on the 'Comparing changes' page
8. Select "Create pull request" on the 'Open a pull request' page
9. Feel free to type @_____  in the comments so I can review the change as well if I'm not the author of the page

Feel free to ping me directly on Teams or over email if you run into any issues.
)
```

Miscellaneous templates

The current date

KEYWORD	OUTPUT
datep0	05/01

Place the following script in your .ahk file to enable this keyword script:

```
:r0:datep0::
  Date := A_Now
  FormatTime, nDate, %Date%, MM/dd
  Send, %nDate%
Return
```

The date two days from now

KEYWORD	OUTPUT
datep2	05/03

Place the following script in your .ahk file to enable this keyword script:

```
:r0:datep2::
  Date := A_Now
  Date += 2, Days
  FormatTime, nDate, %Date%, MM/dd
  Send, %nDate%
Return
```

FAQ

How do I modify keywords?

The keyword is the second word between the colons. So in `:r0:datep0::`, the keyword is datep0. If you want to modify the keywords, just modify the text between the colons for the script you're interested in changing.

How are scripts managed?

All the scripts are located within your .ahk file, in text format. To review or edit your existing scripts, or add new scripts, simply open the .ahk file, make your changes, save the .ahk file, and then double-click it to initiate the script to enforce your new changes. If your .ahk file lives in your startup folder and you made your changes elsewhere, you'll need to replace the existing .ahk file in your startup folder with the one you just changed.

Do I need to recreate the scripts on every machine?

Theoretically, yes. But in practice, keep the same .ahk file on your Onedrive so it's the same file on every machine, and then just place this file in the startup folder of each machine. If you need to make changes, make changes to the file on Onedrive, and then replace the existing file in the startup folder. Autohotkey does need to be installed manually on each machine.

Is it possible to make the scripts more complex?

Yes. Check out the scripting showcase for more information: https://autohotkey.com/docs/scripts/.

Full script

Instead of copying each individual script snippet, you can also bulk copy the entire script and save it in your `.ahk` file.

Place the following script in your .ahk file to enable all the keywords found within this topic. Don't forget to modify the underscores to relevant values.

```
:r0:vnotee::
(
Status
======

To do
======

Done
====

Git
====

Contacts
========

Details
=======
)

:r0:dnotee::

FormatTime, Time,, yyyy-MM-dd
Send %Time%
send {Space}| Sprint
return

:r0:rnotee::
(
====================================================
                    To do
```

```
====================================================

Tasks
------
1.
2.
3.


Public PRs
-----------
1.
2.
3.


Git Issues
----------
1.
2.
3.


====================================================
                   Done
====================================================

Tasks
------

Public PRs
-----------

Git Issues
----------
)


:r0:ishnew::
(
Thanks for contacting Microsoft with feedback about our product documentation. Depending on the complexity,
it could take from a few business days to a few weeks to look further into your question, issue, or
suggestion. We'll contact you if we need further information, and you'll receive an email message each time
we "@mention" you in a comment.

If you're suggesting a simple change to the documentation, it might be faster to submit the suggestion
yourself by creating a GitHub pull request (PR). Here's how:

1. On the page you have feedback about, select the **Edit** link at the top right.
1. On the next page, select the **Pencil** icon at the top right.
1. On the next page, in the **Edit file** text window, make your edits directly to the text you want to
change.
    If you need help with formatting the new or changed text, see our [Markdown Cheatsheet]
(https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet).
1. After you've made your edits, under **Commit changes**:
    a. In the first text box, enter a brief description of the change you've made.
    b. In the **Add an optional extended description** box, paste the link to your GitHub issue.
1. Select **Propose file change**.
1. On the **Comparing changes** page, select **Create pull request**.
1. On the **Open a pull request** page, select **Create pull request**.

Allow a day or more for the article's author to review and approve your change or offer an alternative
solution.

By opening a pull request, you're helping the author add your approved change directly to the documentation.
After the updated article is published, GitHub lists you as an article contributor.

If your issue is more complex than a simple change in the documentation, that's okay.  We'll continue to
research your suggestion and then get back to you with our response.
```

```
research your suggestion and then get back to you with our response.

Thanks again for submitting your feedback. Your suggestions help improve our documentation.
)

:r0:ishport::
(
Thanks for contacting Microsoft about your issue. As we understand it, you're describing a problem with the
product itself and not with the product documentation.

To get assistance or support for your product issue, we recommend that you engage with the product community
or open a ticket with Microsoft Support. For more information, see __________________.

To provide product suggestions or ideas for improvement, go to ___________.

If you're submitting feedback about the product documentation, please reply to this comment to clarify your
issue further. Otherwise, we'll proceed with closing out this Git issue within a few business days.

We're sorry that the _________ Docs team was unable to assist you further with your issue, but we appreciate
your reaching out to us. If you experience other issues in the future, we encourage you to contact us again.
)

:r0:ishclose::
(
Thanks again for contacting Microsoft. We've submitted a documentation change based on your suggestion, and
the updated article should be live by tomorrow. Please let us know if there’s anything more we can do for
you. To follow up on this issue, leave us a comment with an @mention.
)

:r0:azprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the
following:

1. Navigate to the page of interest
2. Select  "Edit" button at the top right
3. Modify the URL so that 'azure-docs' becomes 'azure-docs-pr'
5. Press enter to navigate to the newly-changed URL and reload the page
6. Select the pencil to edit the document
7. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-
here/wiki/Markdown-Cheatsheet
8. Scroll down to the bottom, name your file change, and select "Propose file change"
9. Select "Create pull request" on the 'Open a pull request' page
10. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View"
link within the PR
11. Once you're satisfied with your changes, type @_________ in the comments so that I can review and sign
off to approve the changes

If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs
organization. More information can be found here:
https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master

Feel free to ping me directly on Teams or over email if you run into any issues.
)

:r0:sqlprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the
following:

1. Navigate to the page of interest
2. Select the "Edit" button at the top right
3. Modify the URL so that 'sql-docs' becomes 'sql-docs-pr' and 'live' becomes 'master'
4. Press enter to navigate to the newly-changed URL and reload the page
5. Select the pencil to edit the document
6. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-
here/wiki/Markdown-Cheatsheet
7. Scroll down to the bottom, name your file change, and select "Propose file change"
8. Select "Create pull request" on the 'Open a pull request' page
9. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link
```

```
9. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link
within the PR
10. Once you're satisfied with your changes, type @________ in the comments so that I can review and sign
off to approve the changes

If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs
organization. More information can be found here:
https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master

Feel free to ping me directly on Teams or over email if you run into any issues.
)


:r0:pubprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the
following:

1. Navigate to the page of interest
2. Select the "Edit" button at the top right of the page
3. Select the "Pencil" icon on the right
4. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-
here/wiki/Markdown-Cheatsheet
5. Name your file change and provide a description, if necessary
6. Select "Propose file change"
7. Select "Create pull request" on the 'Comparing changes' page
8. Select "Create pull request" on the 'Open a pull request' page
9. Feel free to type @_____  in the comments so I can review the change as well, if I'm not the author of
the page

Feel free to ping me directly on Teams or over email if you run into any issues.
)

:r0:datep0::
  Date := A_Now
  FormatTime, nDate, %Date%, MM/dd
  Send, %nDate%
Return

:r0:datep2::
  Date := A_Now
  Date += 2, Days
  FormatTime, nDate, %Date%, MM/dd
  Send, %nDate%
Return
```

Docs Authoring Pack for VS Code

6/24/2021 • 7 minutes to read

The Docs Authoring Pack is a collection of VS Code extensions to aid with Markdown authoring for docs.microsoft.com. The pack is available in the VS Code Marketplace and contains the following extensions:

- Docs Markdown: Provides Markdown authoring assistance for docs.microsoft.com (Docs) content, including basic Markdown support and support for custom Markdown syntax in Docs, such as alerts, code snippets, and non-localizable text. Now also includes some basic YAML authoring assistance, such as inserting TOC entries.
- markdownlint: A popular Markdown linter by David Anson to help make sure your Markdown is valid.
- Code Spell Checker: A fully offline spell checker by Street Side Software.
- Docs Preview: Uses the docs.microsoft.com CSS for more accurate Markdown preview, including custom Markdown.
- Docs Article Templates: Allows users to scaffold Learn modules and apply Markdown skeleton content to new files.
- Docs YAML: Provides Docs YAML schema validation and auto-complete.
- Docs Images: Provides image compression and resizing for folders and individual files to help authors of docs.microsoft.com.

Prerequisites and assumptions

To insert relative links, images, and other embedded content with the Docs Markdown extension, you must have your VS Code workspace scoped to the root of your cloned Open Publishing System (OPS) repo. For example, if you have cloned the docs repository to `C:\git\SomeDocsRepo\`, then open that folder or a subfolder in VS Code: **File** > **Open Folder** menu, or `code C:\git\SomeDocsRepo\` from the command line.

Some syntax supported by the extension, such as alerts and snippets, are custom Markdown for OPS. Custom Markdown will not render correctly unless published via OPS.

How to use the Docs Markdown extension

To access the **Docs Markdown** menu, type ALT+M. You can click or use the up and down arrows to select the command you want. Or you can type to start filtering, then hit ENTER when the function you want is highlighted in the menu.

See the Docs Markdown readme for an up-to-date list of commands.

How to generate a master redirect file

The Docs Markdown extension includes a script to generate or update a master redirection file for a repo, based on the `redirect_url` metadata in individual files. This script checks every Markdown file in the repo for `redirect_url`, adds the redirection metadata to the master redirection file (*.openpublishing.redirection.json*) for the repo, and moves the redirected files to a folder outside the repo. To run the script:

1. Select F1 to open the VS Code command palette.
2. Start typing "Docs: Generate..."
3. Select the command `Docs: Generate master redirection file`.
4. When the script finishes running, the redirection results will show in the VS Code output pane, and the

removed Markdown files will be added to the Docs Authoring\redirects folder under your default path.

5. Review the results. If they are as expected, submit a pull request to update the repo.

How to assign keyboard shortcuts

1. Type `CTRL+K` then `Ctrl+S` to open the **Keyboard Shortcuts** list.
2. Search for the command, such as `formatBold`, for which you want to create a custom key binding.
3. Click the plus that appears near the command name when you mouse over the line.
4. After a new input box is visible, type the keyboard shortcut you want to bind to that particular command. For example, to use the common shortcut for bold, type `Ctrl+B`.
5. It's a good idea to insert a `when` clause into your key binding, so it won't be available in files other than Markdown. To do this, open *keybindings.json* and insert the following line below the command name (be sure to add a comma between lines):

   `"when": "editorTextFocus && editorLangId == 'markdown'"`

   Your completed custom key binding should look like this in *keybindings.json*:

```
[
    {
        "key": "ctrl+b",
        "command": "formatBold",
        "when": "editorTextFocus && editorLangId == 'markdown'"
    }
]
```

> **TIP**
>
> Place your key bindings in this file to overwrite the defaults

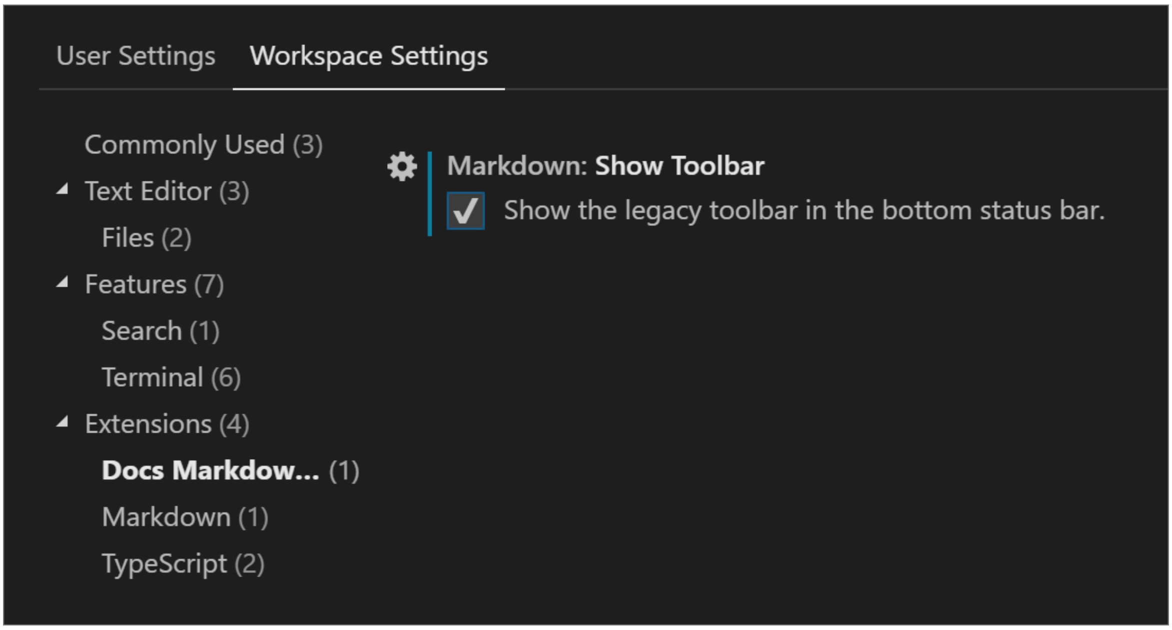
6. Save *keybindings.json*.

For more information on key bindings, see the VS Code docs.

How to show the legacy "Gauntlet" toolbar

Former users of the extension code-named "Gauntlet" will notice that the authoring toolbar no longer appears at the bottom of the VS Code window when the Docs Markdown Extension is installed. This is because the toolbar took up a large space on the VS Code status bar and did not follow best practices for extension UX, so it is deprecated in the new extension. However, you can optionally show the toolbar by updating your VS Code settings.json file as follows:

1. In VS Code, go to **File** > **Preferences** > **Settings** or select `Ctrl+,`.
2. Select **User Settings** to change the settings for all VS Code workspaces or **Workspace Settings** to change them for just the current workspace.
3. Select **Extensions** > **Docs Markdown Extension Configuration**, and then select **Show the legacy toolbar in the bottom status bar**.



Once you've made your selection, VS Code updates the *settings.json* file. You will then be prompted to reload the window for the changes to take effect.

Newer commands added to the extension will not be available from the toolbar.

How to use Docs templates

The Docs Article Templates extension lets writers in VS Code pull a Markdown template from a centralized store and apply it to a file. Templates can help ensure that required metadata is included in articles, that content standards are followed, and so on. Templates are managed as Markdown files in a public GitHub repository.

> **TIP**
> The core Docs templates described in the Docs template finder article are available in the Docs Article Templates extension.

To apply a template in VS Code

1. Ensure the Docs Article Templates extension is installed and enabled.
2. If you don't have the Docs Markdown extension installed, click `F1` to open the command palette, start typing "template" to filter, then click `Docs: Template`. If you do have Docs Markdown installed, you can use either the command palette or click `Alt+M` to bring up the Docs Markdown QuickPick menu, then select `Template` from the list.
3. Select the desired template from the list that appears.

To add your GitHub ID and/or Microsoft alias to your VS Code settings

The Templates extension supports three dynamic metadata fields: author, ms.author, and ms.date. That means that if a template creator uses these fields in the metadata header of a Markdown template, they will be auto-populated in your file when you apply the template, as follows:

METADATA FIELD	VALUE
`author`	Your GitHub alias, if specified in your VS Code settings file.
`ms.author`	Your Microsoft alias, if specified in your VS Code settings file. If you are not a Microsoft employee, leave unspecified.

METADATA FIELD	VALUE
`ms.date`	The current date in the Docs-supported format, `MM/DD/YYYY`. The date is not automatically updated if you subsequently update the file - you must update it manually. This field is used to indicate the "article freshness".

To set author and/or ms.author

1. In VS Code, go to **File** > **Preferences** > **Settings** or select `Ctrl+,`.
2. Select **User** settings to change the settings for all VS Code workspaces, or **Workspace** settings to change them for just the current workspace.
3. In the Default Settings pane on the left, find **Docs Article Templates Extension Configuration**, click the pencil icon next to the desired setting, then click Replace in Settings.
4. The **User** settings pane will open side by side, with a new entry at the bottom.
5. Add your GitHub ID or Microsoft email alias, as appropriate, and save the file.
6. You might need to close and restart VS Code for the changes to take effect.
7. Now, when you apply a template that uses dynamic fields, your GitHub ID and/or Microsoft alias will be auto-populated in the metadata header.

To make a new template available in VS Code

1. Draft your template as a Markdown file.
2. Submit a pull request to the templates folder of the MicrosoftDocs/content-templates repo.

The docs.microsoft.com team will review your template and merge the PR if it meets docs.microsoft.com style guidelines. Once merged, the template will be available to all users of the Docs Article Templates extension.

Demo several features

Here's a short video that demonstrates the following features of the Docs Authoring Pack:

- **YAML files**
  - Support for "Docs: Link to file in repo"
- **Markdown files**
  - Update “ms.date" Metadata Value context menu option
  - Code auto-completion support for code-fence language identifiers
  - Unrecognized code-fence language identifier warnings / auto correction support
  - Sort selection ascending (A to Z)
  - Sort selection descending (Z to A)

Contribution expectations

The Docs Authoring Pack extension is open source and available for contributions to anyone with a GitHub account. There is a dedicated internal Microsoft team that actively works on this project. This team monitors issues and pull requests. The service level agreement (SLA) and expectation of getting a pull request reviewed is currently one week. The team is undergoing automation efforts to improve this turn around time.

Next steps

Explore the various features available in the Docs Authoring Pack, Visual Studio Code extension.

- Dev lang completion

- Image compression
- Metadata updates
- Reformat table
- Smart quote replacement
- Sort redirects
- Sort selection

Dev lang completion

6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

Contributors need assistance determining the valid language identifiers (dev langs) that can follow triple-backticks (code fence openings) in a Markdown file. Unfortunately, build-time validation of dev langs doesn't exist. The result is disparate representations of a single language within the same conceptual docset.

Consider C# as an example. Contributors have used `c#`, `C#`, `cs`, `csharp`, and others as dev lang representations of the language. Which of the preceding representations is correct?

The *Dev lang completion* feature dispels the confusion by displaying a list of known dev langs. Upon selecting a dev lang name from IntelliSense:

- The code fence is closed.
- The caret is positioned in the code fence.

Preferences

It's not possible to disable this feature. The following settings are available:

- Display commonly used dev langs
- Display all known dev langs

Display commonly used dev langs

Only a subset of the valid dev langs will be used in a single docset. To enhance the user experience:

1. In Visual Studio Code, open the docset to the root directory.
2. Select **File** > **Preferences** > **Settings** and filter by *Docs Markdown Extension*.
3. Click the **Edit in settings.json** link in the **Markdown: Docset Languages** section.
4. Add the following `markdown.docsetLanguages` property to the *settings.json* file:

   ```
   {
       "markdown.docsetLanguages": [

       ]
   }
   ```

5. Position your caret in the property's empty array, and activate IntelliSense (via `Ctrl` + `Space`). A list of known dev lang names appears.
6. Add dev lang names to the array until you're satisfied with the list. For example, the following list will display four dev lang names to the user after typing triple-ticks:

```json
{
    "markdown.docsetLanguages": [
        ".NET Core CLI",
        "C#",
        "Markdown",
        "YAML"
    ]
}
```

7. Save your changes to the *settings.json* file.

> **WARNING**
> An empty `markdown.docsetLanguages` array causes all known dev langs to display.

Display all known dev langs

By default, all known dev lang names are displayed in IntelliSense. This setting overrides the `markdown.docsetLanguages` property described in Display commonly used dev langs.

To change this setting:

1. Select **File** > **Preferences** > **Settings** and filter by *Docs Markdown Extension*.
2. Toggle the setting in the **Markdown: All Available Languages** section.

In action

Below is a brief demonstration of this feature:

Image compression

6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, Visual Studio Code meta extension is comprised of multiple sub extensions. This feature is included in the Docs Images extension. The Docs Markdown extension is part of the Docs Authoring Pack, there is no need to install it separately.

Summary

All documentation is provided via the web, with the exception of PDF versions of docs. When serving static content, it is best to minimize the number of bytes sent over the wire. One way to do that is to compress images at rest.

The Docs Authoring Pack extension includes image compression context menu items. The following image types / extensions are supported:

- **.png*
- **.jpg*
- **.jpeg*
- **.gif*
- **.svg*
- **.webp*

The lossless image compression algorithms are used, where applicable.

Compress image

From the **Explorer** navigation pane, right-click on an image file - then select the **Compress image** option. The image is then compressed.

Compress images in folder

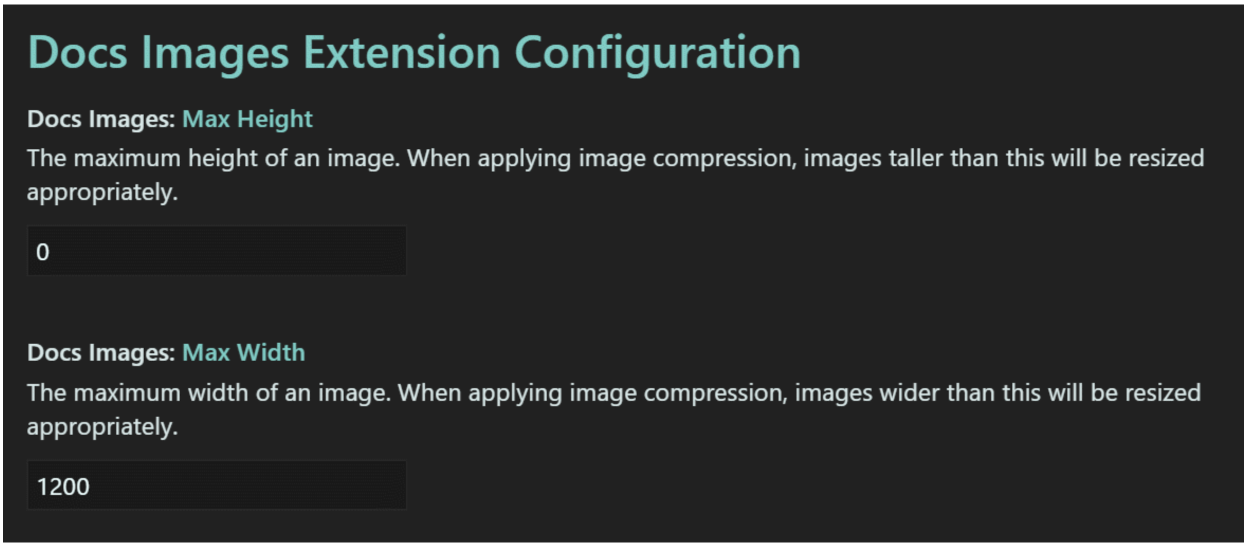
From the **Explorer** navigation pane, right-click on a folder containing images - then select the **Compress images in folder** option. All images in the folder are compressed.

Considerations

Large resolution images are implicitly resized. The maximum dimensions are based on the platform suggested max width, suggested here. The max is only used when images are larger than they are recommended to be, and they will maintain the aspect ratio when automatically resized.

Preferences

The maximum dimensions are configurable, but a default max width of `1200` pixels exists. To configure the max dimensions, select **File -> Preferences -> Settings** and filter by `"Docs Image Extension"`.

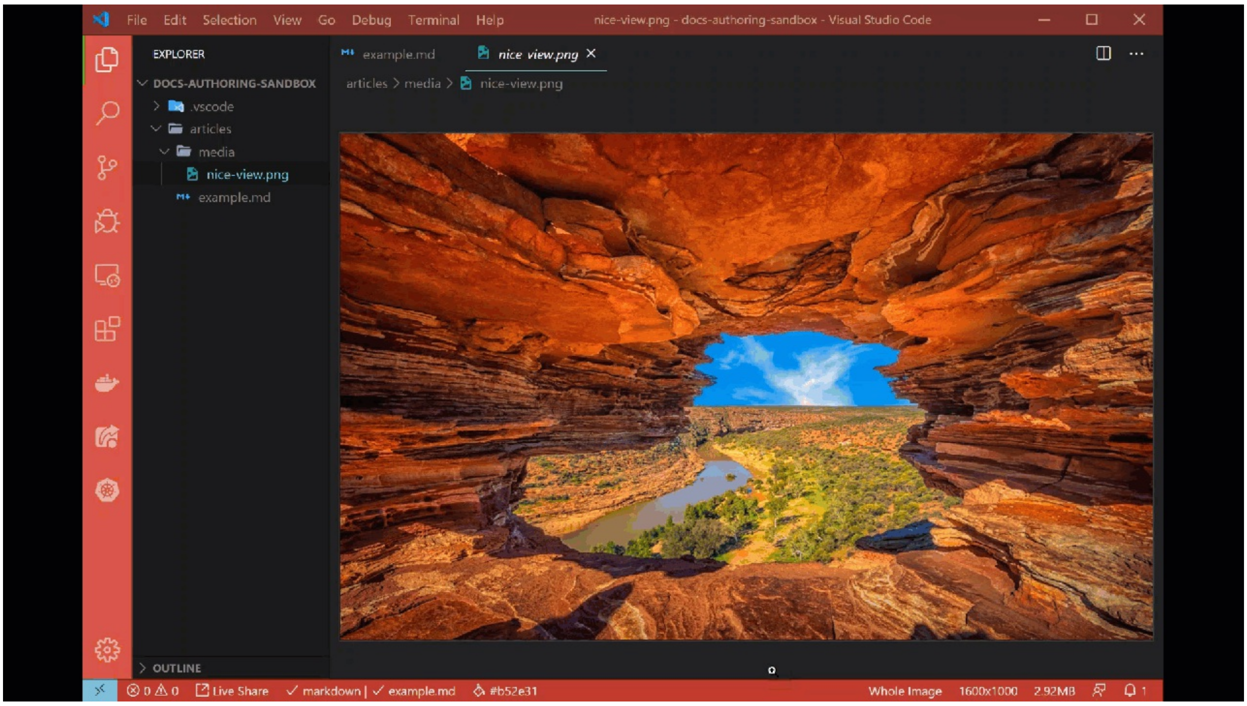


NOTE

A value of `0` in either the **Max Width** or **Max Height** will simply ignore resolution variances.

In action

Below is a brief demonstration of this feature.



Metadata explorer

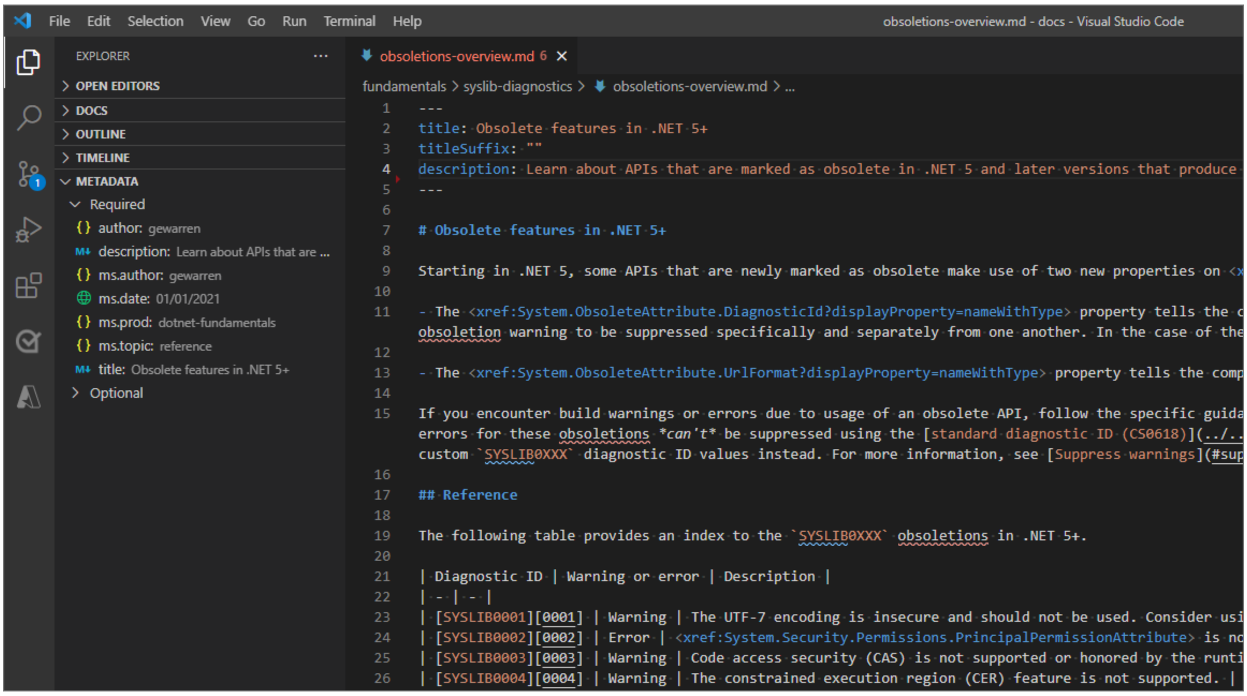
6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

The Metadata explorer appears automatically in the Explorer side bar of Visual Studio Code when you open a Docs Markdown file. It has two main sections: required metadata and optional metadata.



Metadata sources

The Metadata explorer shows metadata from three sources: the YAML front matter of the Markdown file that's open in the editor, and the `fileMetadata` and `globalMetadata` sections of the docset's *docfx.json* file. Icons help to quickly identify the source of the displayed metadata. If your article is missing any required metadata, that metadata key appears with a warning icon.

ICON	METADATA SOURCE
	YAML front matter of active Markdown file
	`fileMetadata` section of *docfx.json* file
	`globalMetadata` section of *docfx.json* file
	This required metadata is *missing completely*

Precedence

The Metadata explorer only shows unique metadata keys. If more than one value can apply for the same key, the explorer uses the same precedence settings as DocFx does. The order of precedence in decreasing priority is:

1. YAML front matter
2. `fileMetadata`
3. `globalMetadata`

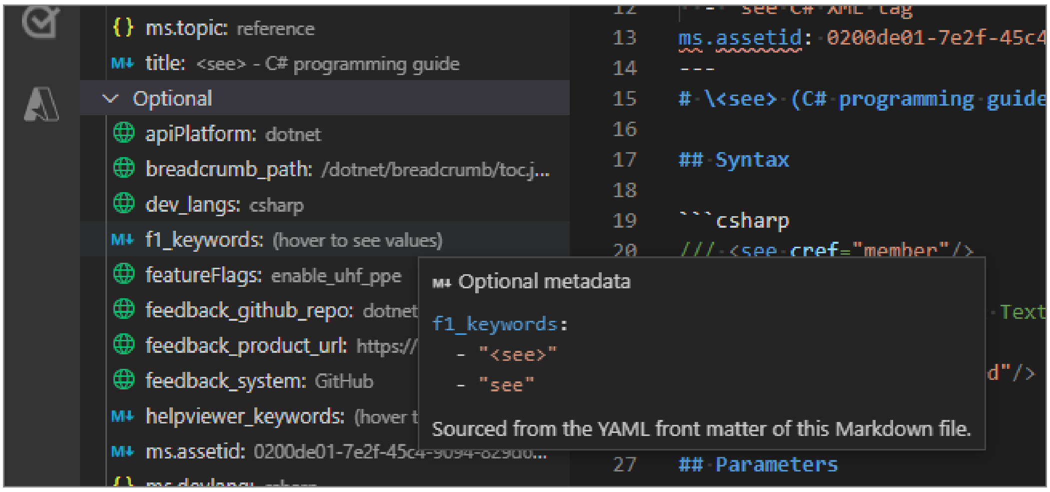
If multiple folder-based glob patterns apply to a single article, the entry that appears last in the *docfx.json* file is selected. For example, consider the file at *C:/docs/csharp/reference/keywords.md* and the following entries in the *docfx.json* file:

```
"ms.topic": {
  "csharp/**/*.md": "conceptual",
  "csharp/reference/*.md": "language-reference"
}
```

In this case, while both glob patterns include the path to the file, the second value of `language-reference` is selected.

Tooltip

You can hover over a value to see an expanded display in the tooltip. This is especially useful for keys that have multiple values in an array. The information also includes the source of the metadata.



Refresh

If you have auto-save disabled in Visual Studio Code, the Metadata explorer refreshes automatically when you save your Markdown file. If you have auto-save enabled, you can refresh the displayed metadata by clicking the **Refresh** button at the top of the explorer.

When you open a different Markdown file, the explorer refreshes automatically.

See also

- How to apply metadata
- Metadata attributes

Update metadata

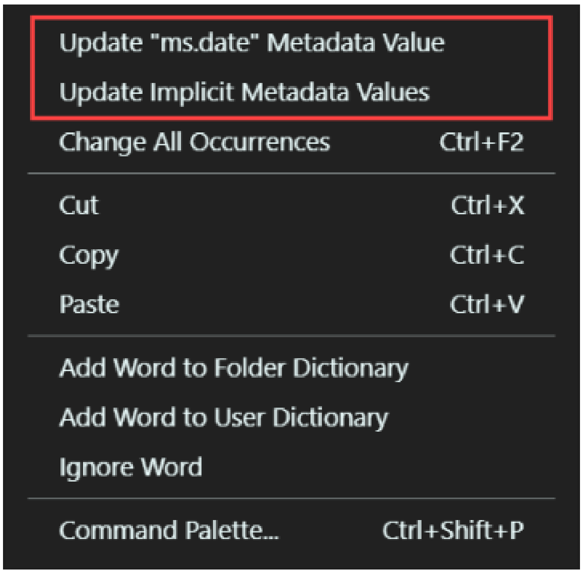
6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

In a Markdown (*.md*) file, there are two contextual menu items specific to metadata. When you right-click anywhere in the text editor, you will see something similar to the following menu items:



Update `ms.date` metadata value

Selecting the **Update** `ms.date` **Metadata Value** option will set the current Markdown files `ms.date` value to today's date. If the document does not have an `ms.date` metadata field, no action is taken.

Update implicit metadata values

Selecting the **Update implicit metadata values** option will find and replace all possible metadata values that could be implicitly specified. Metadata values are implicitly specified in the *docfx.json* file, under the `build/fileMetadata` node. Each key value pair in the `fileMetadata` node represents metadata defaults. For example, a Markdown file in the *top-level/sub-folder* directory that omits the `ms.author` metadata value could implicitly specify a default value to use in the `fileMetadata` node.

```
{
    "build": {
        "fileMetadata": {
            "ms.author": {
                "top-level/sub-folder/**/**.md": "dapine"
            }
        }
    }
}
```

In this case, all Markdown files would implicitly take on the `ms.author: dapine` metadata value. The feature acts on these implicit settings found in the *docfx.json* file. If a Markdown file contains metadata with values that are explicitly set to something other than the implicit values, they are overridden.

Consider the following Markdown file metadata, where this Markdown file resides in **top-level/sub-folder/includes/example.md**:

```
---
ms.author: someone-else
---

# Content
```

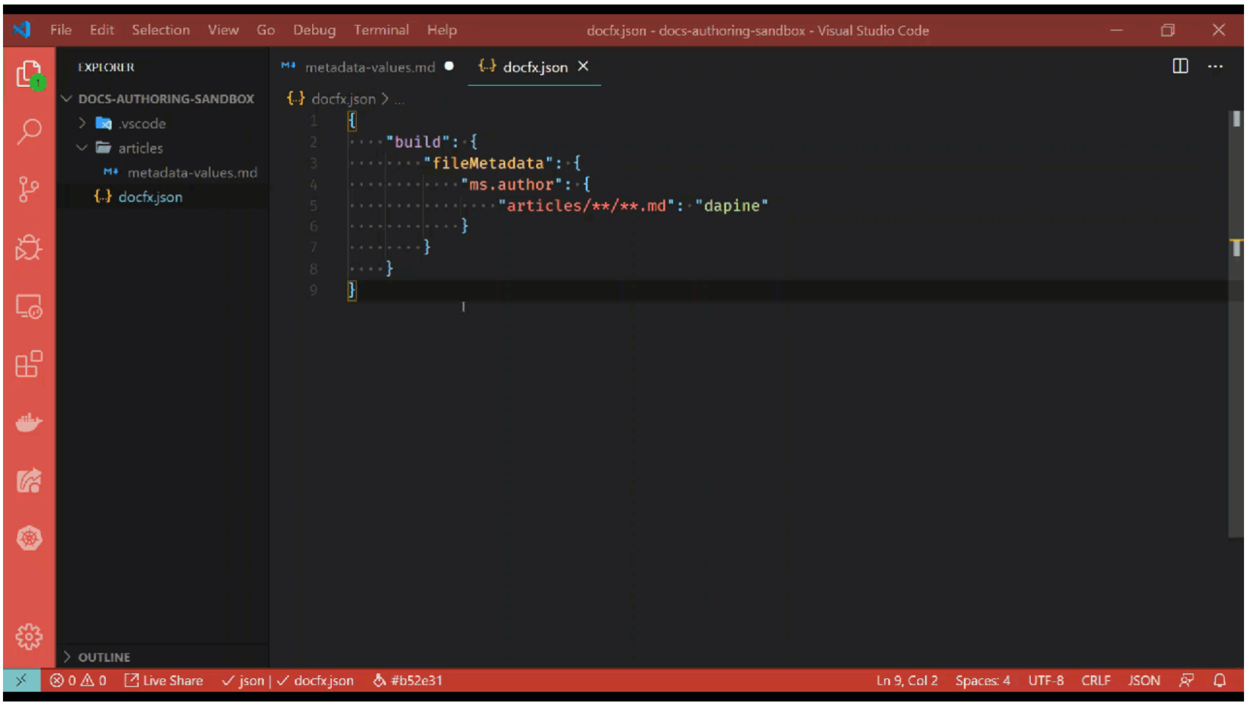
If the **Update implicit metadata values** option was executed on this file, with the assumed *docfx.json* content from above the metadata value would be updated to `ms.author: dapine`.

```
---
ms.author: dapine
---

# Content
```

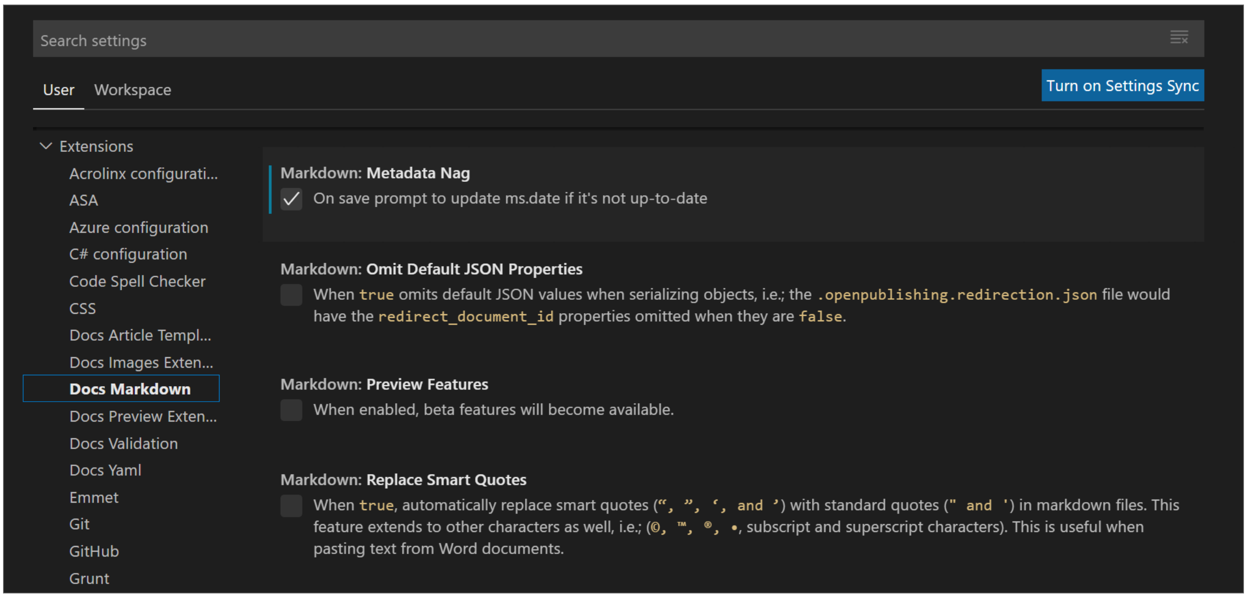
In action

Below is a brief demonstration of this feature.

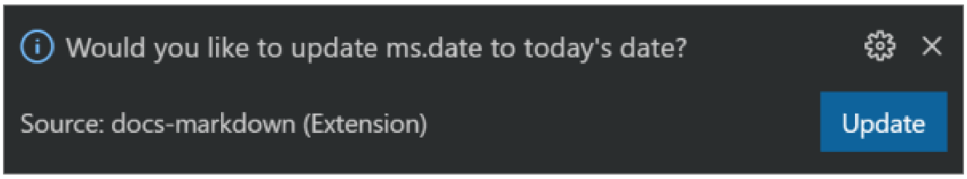


Get prompted to update `ms.date`

You can be prompted to update the date automatically by enabling the **Markdown: Metadata nag** setting:



When a file is saved, the following prompt appears if `ms.date` has not been updated:



To update the date to the current date, select **Update**. If you're making small changes that don't warrant an updated date, select the **x** to close the prompt. The prompt only appears once per file, per Visual Studio Code session.

Reformat Markdown tables

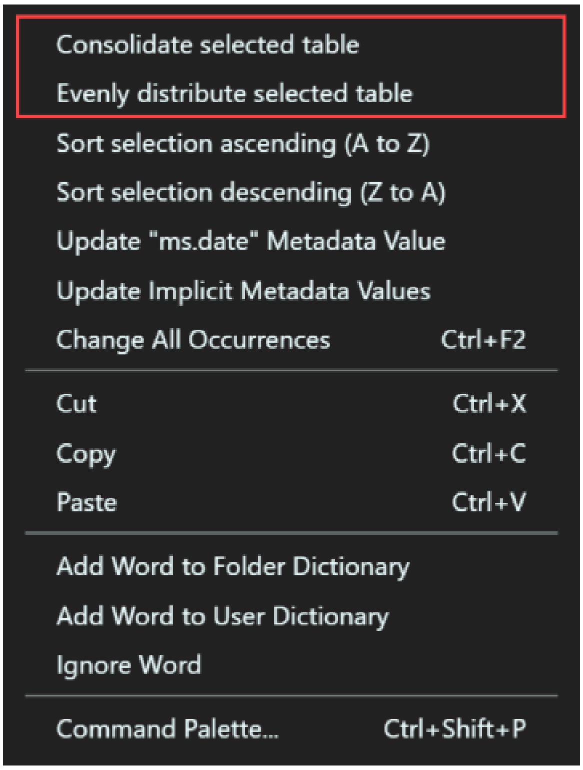
6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

In a Markdown (*.md*) file, when you select a complete table - two table formatting context menu items are now available. Right-click on the selected Markdown table to open the context menu. You will see something similar to the following menu items:



TIP

This feature **does not** work with multiple table selections, but rather is intended for a single Markdown table. You must select the entire table, including headings for desired results.

Consolidate selected table

Selecting the **Consolidate selected table** option will collapse the table headings and contents with only a single space on either side of each value.

Evenly distribute selected table

Selecting the **Evenly distribute selected table** option will calculate the longest value in each column and evenly distribute all the other values accordingly with space.

Considerations

The feature will not impact the rendering of the table, but it will help to improve the readability of the table - thus making more maintainable. The reformatting table feature will keep column alignment intact.

Consider the following table:

```
| Column1 | This is a long column name | Column3 |  |
|--:|---------|:--:|:----|
||         |  |         |
|     |  |         |   a value      |
||         |         |         |
|     |         | This is a long value |       but why? |
|     |         |         |         |
|     |                                         |         | Here is something |
|  |         |   |         |
```

After being "evenly distributed":

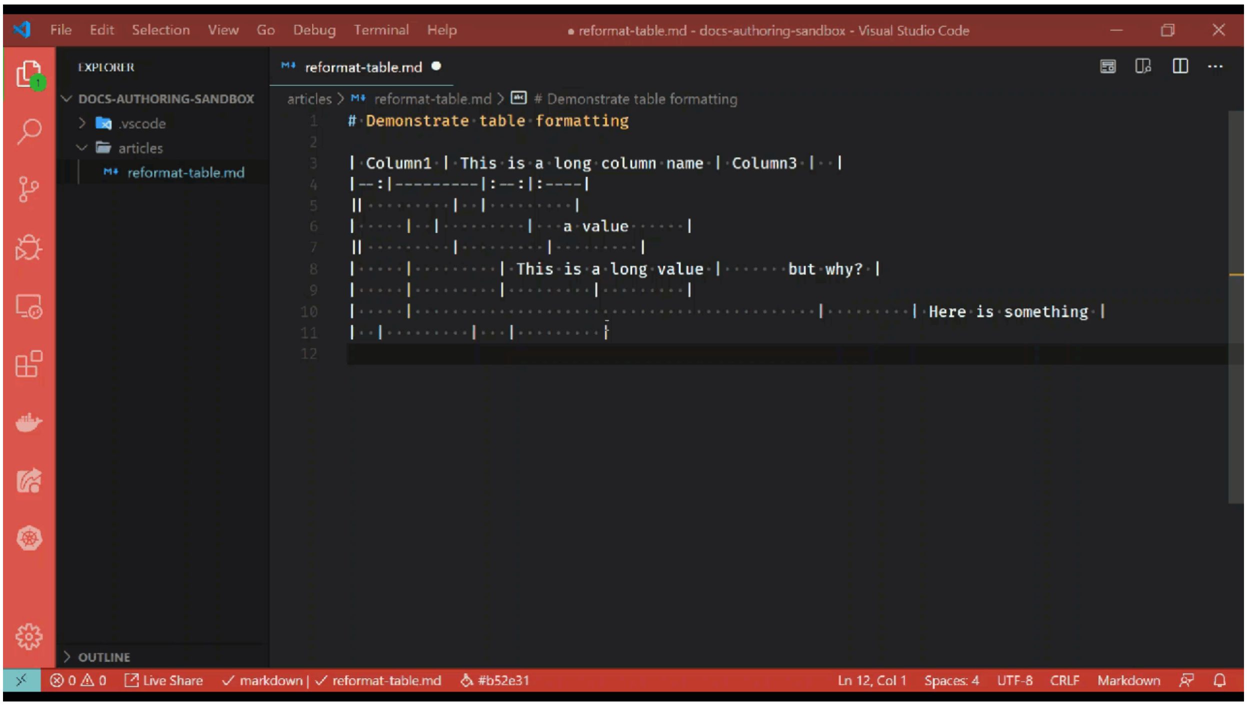
```
| Column1 | This is a long column name | Column3              |                   |
|--------:|----------------------------|:--------------------:|:------------------|
|         |                            |                      |                   |
|         |                            |                      | a value           |
|         |                            |                      |                   |
|         |                            | This is a long value | but why?          |
|         |                            |                      |                   |
|         |                            |                      | Here is something |
|         |                            |                      |                   |
```

After being "consolidated":

```
| Column1 | This is a long column name | Column3 |  |
|-:|--|:-:|:-|
|  |  |  |  |
|  |  |  | a value |
|  |  |  |  |
|  |  | This is a long value | but why? |
|  |  |  |  |
|  |  |  | Here is something |
|  |  |  |  |
```

In action

Below is a brief demonstration of this feature.


File Edit Selection View Go Debug Terminal Help
● reformat-table.md - docs-authoring-sandbox - Visual Studio Code
EXPLORER
DOCS-AUTHORING-SANDBOX
.vscode
articles
reformat-table.md
OUTLINE
reformat-table.md ●
articles > reformat-table.md > # Demonstrate table formatting
Demonstrate table formatting
Column1	This is a long column name	Column3	
			a value
		This is a long value	but why?
			Here is something
0 0 Live Share ✓ markdown | ✓ reformat-table.md #b52e31
Ln 12, Col 1 Spaces: 4 UTF-8 CRLF Markdown

Smart quote replacement

6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

Content developers are responsible for authoring some of the most advanced features of modern technology and intelligence, yet our tooling today prefers the usage of "dumb quotes" in Markdown. The antonym of course being "smart quotes". Smart quotes are common with ideal typographies, but not preferred with Markdown and rendered HTML.

When working on a Microsoft Word document for example, you may have noticed that when you hold the `Shift` and type a " Microsoft Word quickly replaces the `"` character with a smart quote equivalent `“` character.

DESCRIPTION	UNICODE	SMART QUOTE	REPLACEMENT
Double left quote	`\u201c`	`“`	`"`
Double right quote	`\u201d`	`”`	`"`
Single left quote	`\u2018`	`‘`	`'`
Single right quote	`\u2019`	`’`	`'`

In a Markdown (**.md*) file, when you paste in text or as you update content - this feature will actively evaluate the content and automatically replace values accordingly.

NOTE

The smart quote replacement feature also replaces other characters, such as but not limited to; (`©, ™, ®, •`, subscript and superscript characters). This is useful when pasting text from Word documents.

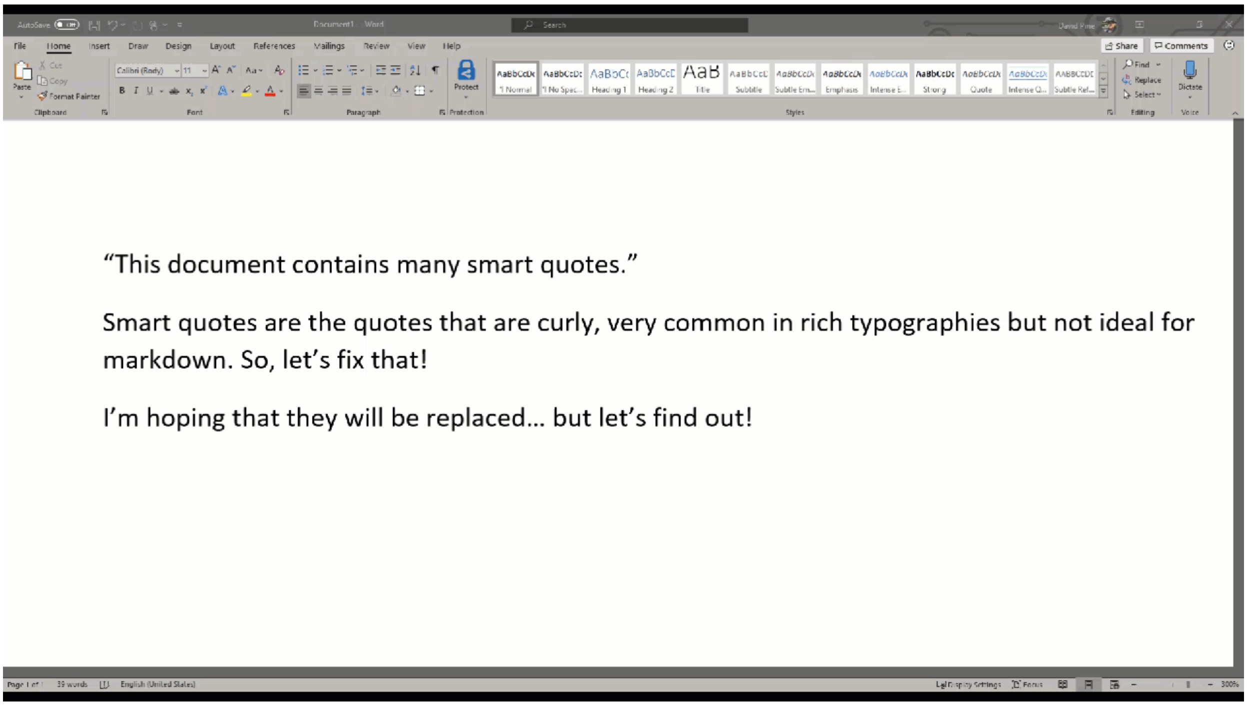
Preferences

This feature is optional, but defaults to `true`. To toggle this feature on or off:

1. Select **File** > **Preferences** > **Settings** and filter by *Docs Markdown Extension*.
2. Toggle the setting in the **Markdown: Replace Smart Quotes** section.

In action

Below is a brief demonstration of this feature.



“This document contains many smart quotes.”

Smart quotes are the quotes that are curly, very common in rich typographies but not ideal for markdown. So, let’s fix that!

I’m hoping that they will be replaced… but let’s find out!

Sort redirects

6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

With the evolution of a docs.microsoft.com docset, some Markdown files eventually will be deleted. When a Markdown file is deleted, we're required to provide a redirect so that any reference to the deleted article is properly resolved via the redirect. Redirections are specified in the *.openpublishing.redirection.json* file.

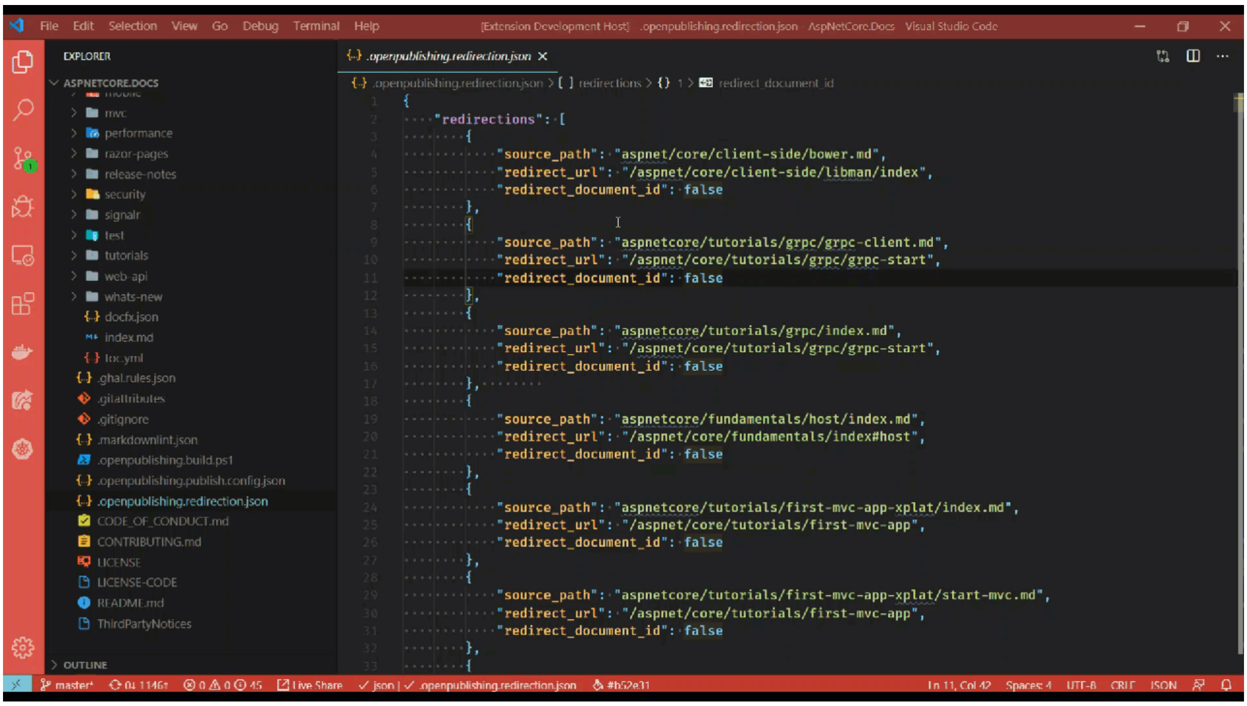
1. Open the Command Palette, press `F1` (or ⇧⌘P on macOS)
2. Type: `Docs: Sort master redirection file`
3. Select the command to execute it
4. Observe changes to *.openpublishing.redirection.json* file

Considerations

The concept of "daisy chaining" exists with how the *.openpublishing.redirection.json* file was originally designed. Over time, files added as a redirect will eventually become stale. This happens when file A is deleted and needs a redirect to file B, then later file B is deleted and then redirects to file C. Ideally, both entries would point to C - so that A redirects to C, and B remains the same. This is a minor performance gain, and the feature is actively being worked on.

In action

Below is a brief demonstration of this feature.



Sort selection

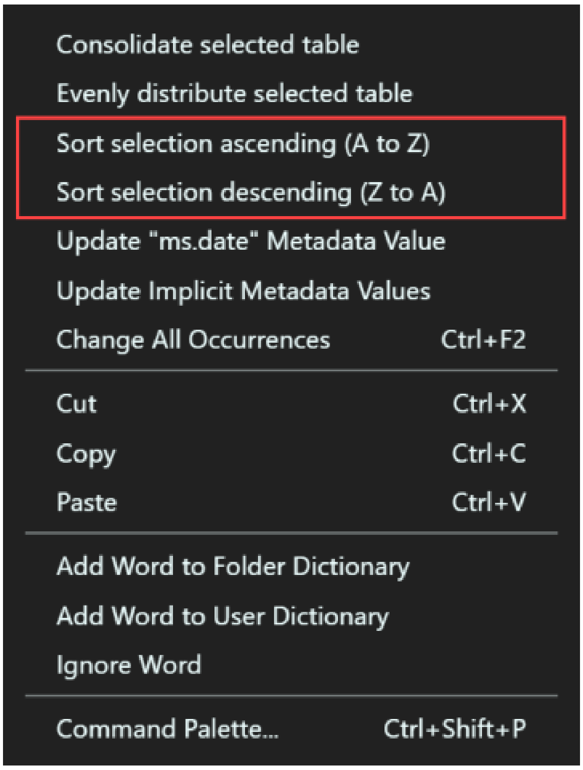
6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

In a Markdown (**.md*) file, when you've made a selection - two sorting context menu items are now available. Right-click on the selected text to open the context menu. You will see something similar to the following menu items:



TIP

The sorting context menu items are hidden until there is a valid selection in the Visual Studio Code text editor.

Sort selection ascending (A to Z)

Selecting the **Sort selection ascending (A to Z)** option will sort the entire selection ascending, alphabetically from A to Z.

Sort selection descending (Z to A)

Selecting the **Sort selection descending (Z to A)** option will sort the entire selection ascending, alphabetically from Z to A.

Considerations

The underlying sorting mechanism uses *natural language* sorting. This makes it more powerful and comprehensive than standard sorting. Consider the following table:

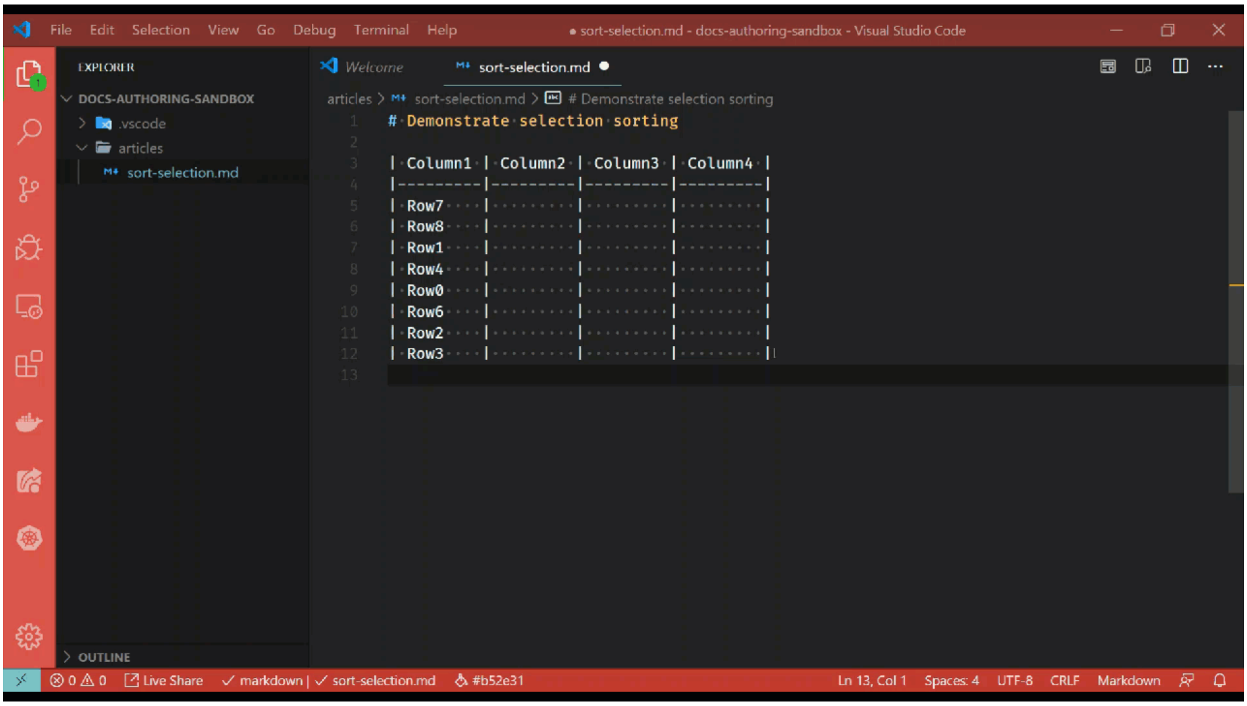
```
| Column1 | Column2                               |
|---------|---------------------------------------|
| 1       | Number 1                              |
| Aa      | The first letter in the alphabet      |
| Ab      | The first letter in the alphabet      |
| C       | The a letter after A in the alphabet  |
| M       | Somewhere in the middle?              |
| 2       | Number 2                              |
| X       | The alphabet letter is towards the end |
| Z       | The last letter in the alphabet       |
| 11      | Number 11                             |
```

Without natural language sorting, the order for `Column1` would have been 1, 11, 2, etc. but instead it understands that 11 is greater than 2 - resulting in the following ascending order:

```
| Column1 | Column2                               |
|---------|---------------------------------------|
| 1       | Number 1                              |
| 2       | Number 2                              |
| 11      | Number 11                             |
| Aa      | The first letter in the alphabet      |
| Ab      | The first letter in the alphabet      |
| C       | The a letter after A in the alphabet  |
| M       | Somewhere in the middle?              |
| X       | The alphabet letter is towards the end |
| Z       | The last letter in the alphabet       |
```

In action

Below is a brief demonstration of this feature.



Insert and update content from a Jupyter notebook

6/24/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

IMPORTANT

Insert and update from a Jupyter notebook functionality does not currently work on a Mac.

Summary

Jupyter Notebooks are a standard interactive way of creating and sharing code in the Python world. The notebook contains a combination of Python code, markdown, and optionally output from the code.

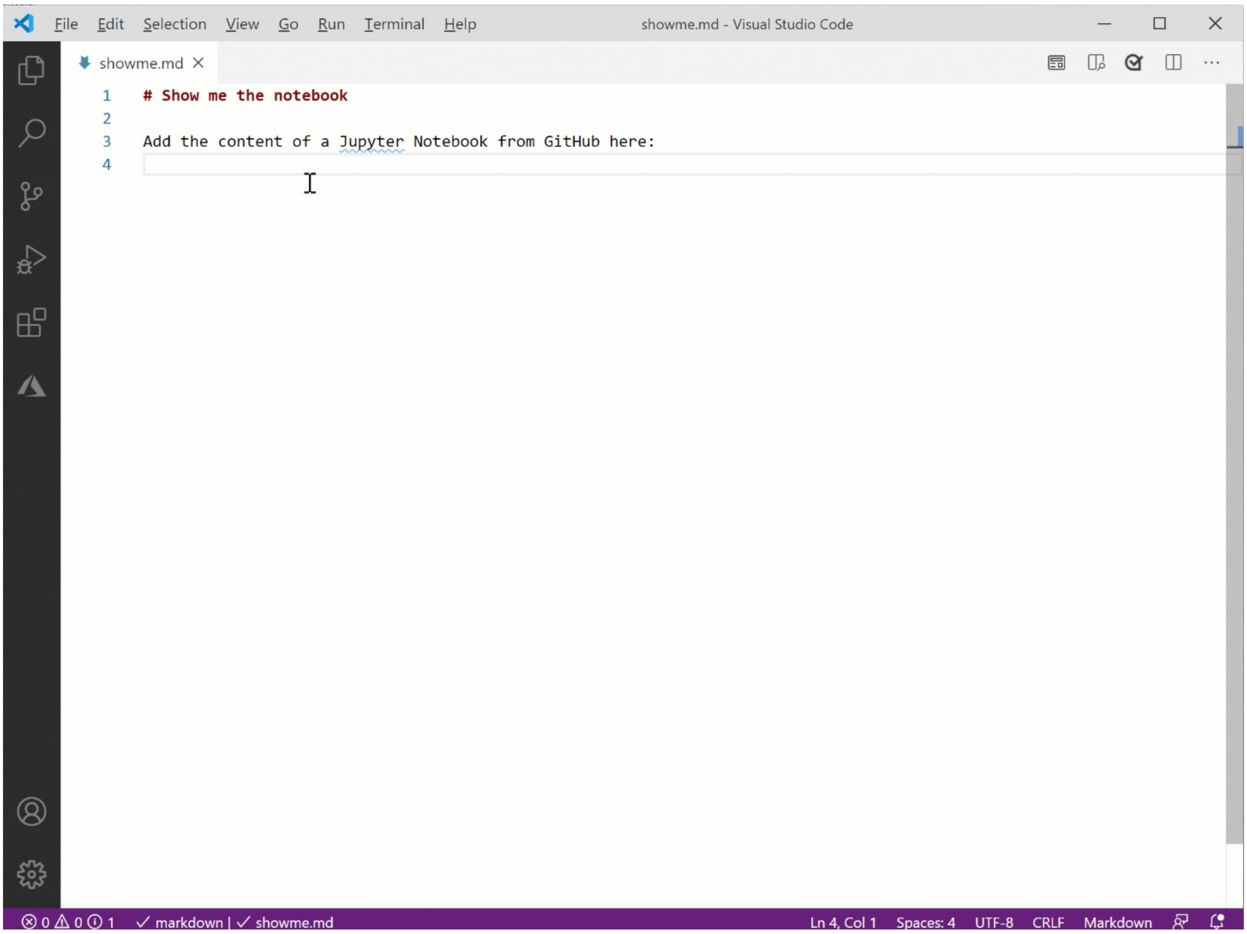
The Docs Authoring Pack extension includes functionality to put a static markdown version of a Jupyter notebook into your document:

Docs: Insert Jupyter notebook: Enter the URL of the notebook. A markdown version of the notebook is added to the document at the position of your cursor. Do not modify the start or end tags; it's what the next function uses to update the notebook.

Docs: Update Jupyter notebook: This function will replace the previously inserted notebook content between start and end with the latest version. No need to enter the URL, it's recorded in the start tag. Update functionality assumes there is a single notebook in the document. Don't add multiple notebooks to a single document.

In action

Below is a brief demonstration of this feature.



Troubleshooting

> **IMPORTANT**
>
> Insert and update from a Jupyter notebook functionality does not currently work on a Mac.

These functions need Python, `jupyter`, and `nbconvert` installed on your machine.

To see if Python is installed, open a VS Code terminal and run:

- Windows - `where python`
- Linux/Mac - `which python`

If the return is one or more paths, Python is installed. If not, install Python now.

Next make sure `jupyter` is installed:

- Windows - `where jupyter`
- Linux/Mac - `which jupyter`

If a path is not returned, install `jupyter`:

```
pip install --upgrade jupyter
```

Once both Python and `jupyter` are installed, install `nbconvert`:

```
pip install --upgrade nbconvert
```

Repo cleanup tool (CleanRepo.exe)

5/8/2021 • 3 minutes to read

The CleanRepo.exe command-line tool helps you clean up your repo. To run the tool, you must have a local copy (clone) of the repo you want to run it against. It can:

- Find articles that aren't linked from a TOC file.
- Find, and optionally delete, orphaned media (.png, .jpg, .gif, and .svg) files.
- Map images to the files that reference them.
- Find, and optionally delete, orphaned "shared" Markdown files (include files).
- Format the redirection file using Serialize with pretty printing.
- Find and replace links to redirected files in markdown files.
- Replace "site-relative" links with file-relative links in markdown files.

Installation

To install the tool, either clone the cleanrepo repo and build it yourself, or download and unzip the executable files to a folder on your computer. Then, open a command prompt and change to the directory where you created or copied the executable files.

Usage examples

The next sections show you the command-line options to use with the tool for different functions. You must have a local copy (clone) of the repo you want to "clean" using the tool.

For the functions that modify files, for example, replacing links or deleting orphaned files, don't worry too much about doing something irreversible. It's simple to open up Git Bash (or Visual Studio Code, if you use that for Git operations) and undo the changes. For example, `git checkout -- .` undoes all the changes, including file deletions. To undo file changes in a specific directory, use a command like `git checkout -- pipelines/.`.

> **TIP**
> Consider the execution order of commands. For example, if you intend to delete orphaned images and orphaned articles, you should delete orphaned articles first, then delete orphaned images. The reason for this is that deleting articles could potentially orphan further images.

Orphans

The tool has the ability to delete orphaned markdown files and orphaned media files.

Orphaned articles

This option finds Markdown files that aren't back linked from a TOC.yml or TOC.md file. It ignores Markdown files in directories named *includes* or *_shared*.

Find and delete orphaned articles recursively (that is, in the specified directory and any subdirectories):

```
CleanRepo.exe --orphaned-topics --start-directory c:\repos\visualstudio-docs-pr\docs\ide
```

Orphaned images

Find and delete orphaned .png/.jpg/.gif/.svg files (recursive):

```
CleanRepo.exe --orphaned-images --start-directory c:\repos\visualstudio-docs-pr\docs\ide
```

Orphaned include-type files

Find and delete shared Markdown files that are orphaned (recursive):

```
CleanRepo.exe --orphaned-includes --start-directory c:\repos\visualstudio-docs-pr\docs\ide
```

> **NOTE**
>
> This option considers markdown files in a directory named *includes* or *_shared* to be shared markdown files. It ignores markdown files in directories that aren't named *includes* or *_shared*.

Cataloging

This option generates a JSON file that lists each image (.svg/.jpg/.png/.gif) along with a list of all the files that reference the image.

```
CleanRepo.exe --catalog-images --start-directory c:\repos\visualstudio-docs-pr\docs
```

For example, the output might look like this:

```
{
  "c:\\users\\gewarren\\test-repo\\docs\\debugger\\dbg-tips.png": [
    "c:\\users\\gewarren\\test-repo\\docs\\dual-stack.md",
    "c:\\users\\gewarren\\test-repo\\docs\\dual-stack.md"
  ],
  "c:\\users\\gewarren\\test-repo\\docs\\media\\stratus.png": [],
  "c:\\users\\gewarren\\test-repo\\docs\\media\\two-forest.png": [
    "c:\\users\\gewarren\\test-repo\\docs\\landing.yml"
  ],
  "c:\\users\\gewarren\\test-repo\\docs\\media\\vs.png": [
    "c:\\users\\gewarren\\test-repo\\docs\\configure-ldaps.md"
  ],
  "c:\\users\\gewarren\\test-repo\\docs\\debugger\\dbg-tips.jpg": []
}
```

Redirects

These options relate to redirect links and the redirection JSON file.

Format redirection file

Formats the redirection JSON file by deserializing and then reserializing it using "pretty printing" indentation.

```
CleanRepo.exe --format-redirects --docset-root c:\users\gewarren\dotnet-docs\docs
```

Replace redirected links

Find articles with backlinks to redirected articles and replace the links with their target URL:

```
CleanRepo.exe --replace-redirects --start-directory c:\repos\visualstudio-docs-pr\docs\ide
```

Site-relative links

Replace any site-relative links (that is, those that start with `/<docsetname/`) with file-relative links (for example, `../misc/index.md`). File-relative links are validated at build time, whereas site-relative links are not.

```
CleanRepo.exe --relative-links --start-directory c:\repos\dotnet-docs\docs\core
```

> **TIP**
>
> The Docs Authoring Pack provides functionality to sort and remove daisy chains from a redirection file. For more information, see Docs Authoring Pack: Sort redirects.

See also

- Readme file for CleanRepo.exe
- Docs Authoring Pack: Sort redirects

> **NOTE**
>
> CleanRepo is an "in-house" tool written by a content developer, so you may find bugs or issues with the tool. If you do, please file an issue or submit a PR to fix the problem.

SpineEdit

7/19/2021 • 2 minutes to read

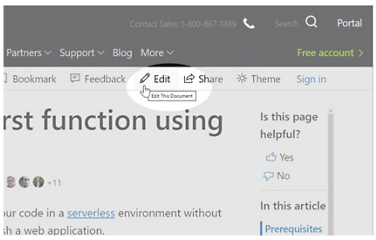
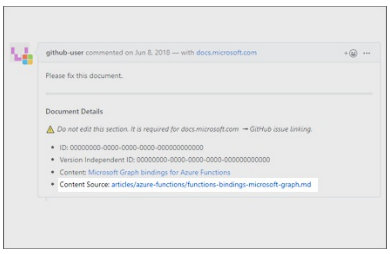
SpineEdit is a browser extension that makes it easy for you to edit docs in the private repository - all in the browser.

Prerequisites

SpineEdit requires you to have read permissions in the private MicrosoftDocs repository you want to access. For information about requesting access to the repository, see this GitHub account setup article.

Make an edit

With the extension installed, you can open an editor in the article's private repository.

WEBSITE	SELECT THE...	EXAMPLE
https://docs.microsoft.com/	*Edit* button	
http://www.github.com	*Content Source* link in an article issue	

> **NOTE**
>
> This workflow only works if you have access to the destination repository.

Installation

Install the extension via the Google webstore.

Feedback and support

To request a feature or report a bug, create an issue in the GitHub repository.

Get to know the Global Experiences Content Localization team

7/20/2021 • 2 minutes to read

DevRel Global Experiences team is localizing all software, sites, technical and Learn content for the Cloud & AI division at Microsoft. The products we support range from Azure to SQL, Visual Studio, .NET, Intune, System Center, to PowerBI and more. In addition to owning the localization of products and services, we are the product owners of the localized technical and Learn content published to docs.microsoft.com with the goal of promoting our content usage.

General Localization Process

C+AI docs localization follows the Continuous Publishing model (CPUB). Most loc projects run in the regular cadence automatically.

For a new localization request, the DevRel Content Localization Production PM team defines the loc cadence, localized language set and loc file set. It takes us one week to onboard the new repository or new docset to the localization system. The initial translation time varies depending on the repo scope and language set. Afterwards, the localization turns to CPUB mode. Contact your project PM owner about your project.

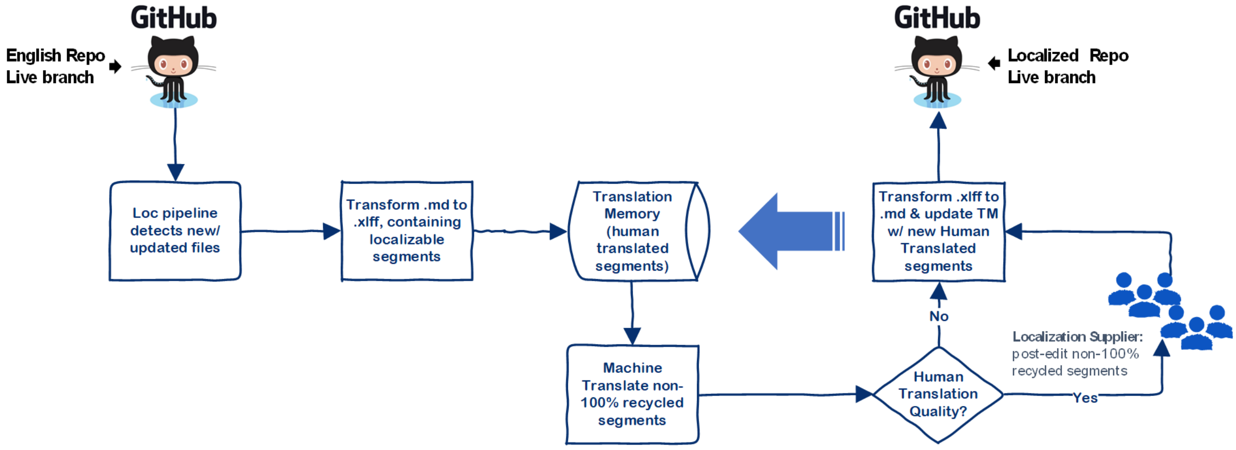
> **IMPORTANT**
> DevRel Global loc team supports the localization of the content authored by C+AI teams. When you have a new set of content haven't been published to DOCS and you are not sure whether they are onboarded to localization pipeline, always contact your content loc PM before merge files to the publishing branch! We'd like to avoid over-localization, thus cross-charge to your team afterwards.

Besides the translation, the localization process supports:

- **Terminology Change Request**: this process defines what to do when the loc team receives the customer or internal feedback about one term translation for a specific language.
- **Source Issue Escalation**: through this process, we provide the translators a channel to escalate questions about the source sentences when they pursue the accurate meaning.
- **Policheck**: the process to check politically sensitive terms in the localized version is a required process for any Microsoft product and content releases. The Content & Learning loc team runs Policheck every month and reviews the false positives term list.

Localization Automation workflow

The DevRel localization tooling automatically detects new or updated files within our localization scope. Loc files are created and pre-translated by using our Translation Memory, which contains all of our previously human translated content. Any segments that cannot be recycled are then machine translated by our own Azure Translation service. All files set to be human translated are sent to our localization supplier, who reviews and post-edits the pre-translated segments to ensure premium quality human translations.



Localization contacts

> **IMPORTANT**
>
> Always create a Localization Feedback ticket for the status of or questions on existing localized articles.

Contact the DevRel Content Localization Production PM team for general localization questions, or escalate to the team manager Hyun-Kyung Song. The following list shows the PM owner for each content area:

- AI Sunny Deng
- Azure Sunny Deng
- Azure Devloper Experiences Sunny Deng
- BizTalk Sunny Deng
- Commerce Bhargavi Desai
- Developer Tools, Frameworks, Languages Khoi Pham
- DOCS Blog Bhargavi Desai
- DOCS Leagal Bhargavi Desai
- DOCS roots/teamplate/tech profile Bhargavi Desai
- DTP (PowerBI and PowerQuery) Khoi Pham
- Enterprise Mobility Bhargavi Desai
- Identity Sunny Deng
- Learn and LearnShared Bhargavi Desai
- Mooncake Sunny Deng
- PartnerCenter, CDA and more Khoi Pham
- Security Sunny Deng
- SQL Server Sunny Deng
- System Center Sunny Deng
- Windows Bhargavi Desai

Frequently asked questions (FAQ) about localization

6/22/2021 • 11 minutes to read

The Developer Relations Global Experiences team owns the localization processes, tooling, and work model for strategic C+AI content. We provide a predictable release schedule and systematize the GitHub workflow for localization. This article addresses common questions about localization.

General localization for C+AI

Do I have a specific localization contact?

Existing repos have 1-2 loc PMs assigned: see Localization contacts. If you cannot identify your loc contact there, email to DevRel Content Localization Production PM Team.

How can I request localization for new content?

- If the content is not/will not be published to docs.microsoft.com, contact the loc PM assigned to your product area (see Localization contacts). If you can't find your product area, email the general team: DevRel Content Localization Production PM Team.
- If the content is/will be published to docs.microsoft.com and you submitted an onboarding request, @mention your localization contact in the onboarding ticket (you will receive a link to the ticket after you complete the form).

> **NOTE**
>
> Not every new repo or docset is localized by default—make sure to notify your loc PM. The PM will go through the loc selection criteria and make the final decision. See How do we select content to localize for details.

What languages do you support?

Technically, we support almost all languages for localization. But different projects may support different sets of languages, including: the strategic requirements; the necessity to align with the SW support language; Microsoft sub and marketing team feedback etc. The final decision is made by the Global Experiences team—refer to the C+AI Language Set for the language set concept.

See below for Azure-specific language availability.

Will *all* my articles be localized for *all* locales?

If a content set is scoped for localization, all articles in that set will be localized, either via machine translation or human translation. Each Global Experiences PM works with product and content team stakeholders to decide which method to use for which articles, depending on the priority of the content and the available budget. We try to align our locales with software and UI localization.

For detailed information, read about the general selection process and also which Azure topic types are prioritized.

I have a new repository. Will my content be localized?

Not every new repo is localized by default. To request localization, see How can I request localization?. The Global Experience PM will go through the loc selection criteria and make the final decision.

If you work with the Docs Contributor Program or a Content Team member, they can help guide you through this step.

My content was added to an existing repository. Do I need to tell the loc team?

Localization scope is defined by file metadata or the repo's folder structure. If new content uses the predefined metadata or folder names, then the Global Experiences team tooling will detect the file-level changes automatically. Here are some repo-specific scope examples:

- Azure content localization is defined by ms.topic.
- Windows Dev Content localization is defined by ms.localizationpriority.
- SQL Server and DevDiv localization is defined by the folder structure.

To be safe, contact your loc PM when you create a new folder in an existing repo because this can affect their budget and the localization persistency.

I have an existing translation for my article in Word format. Can I pass this along to you for publishing?

We prefer you submit feedback directly in the localized repo (for example, as a GitHub issue). You can find this repo if you select **Edit** on a Docs page from the corresponding locale. After we receive the translation, we run a linguistic quality review to make sure the translation complies with the Microsoft Style Guide and terminology definitions.

If the translated language is not supported for that project, contact your project loc owner who will add the scenario in the backlog for the future EOL evaluation.

When do you start translation of my article?

For repos and folders where localization is configured, the loc pipeline will detect changes when your article is merged into the live branch of the source repo. This kicks off the handoff depending on the loc cadence for that repo/project. Then the translator will receive the new or updated articles through the localization pipeline.

We define the loc cadence when we onboard a project. For example, Azure docs handoff kicks off at 5:00 pm every Wednesday. Contact your project loc owner for more about your project.

Why don't I see the localized version one month after I published my article?

The reasons vary by project. Contact your project loc owner to discuss timelines or request a faster release.

Do I need to tag my content somehow to ensure it is localized?

It depends on the repo. See My content was added to an existing repository...

Do you localize art and image files?

Art localization is slow and expensive. The only art localized by default is for Microsoft Learn modules, and the overview and quickstart topics for Azure and Power BI. All other art is by default shown in English.

Occasionally, we make an exception. The acceptance criteria, however, is strict. For example, we will localize art to avoid potential legal/compliance issues or to comply with Microsoft branding in each international market.

To request an exception, contact your project loc owner.

How can I make sure the right words are being translated?

Make sure to capitalize only what should be capitalized, for example, product or service names. This will help the loc team identify a possible proper noun and not translate words unnecessarily. If a word is translated that shouldn’t be, it will be confusing to the local reader. Likewise, if a word is not translated but should be, it is equally confusing.

Will you translate my code?

The loc team will never localize anything formatted as code block or code fence. You can use `inline code style` to prevent translation of code elements within a sentence.

How can I exclude a word from localization/translation?

Some product details and labels need to stay in English, regardless of the reader's locale. In these scenarios, you can identify content to exclude with the wrapper `:::no-loc text="Words to exclude":::`. See the New Hope guide for detailed information. This feature only works for Markdown topics.

Another option is to use inline code style, as explained above, but note that this will change the font formatting.

How do I respond to GitHub issues and public PRs in other languages/locales?

Going forward, DevRel Loc team will partner with other teams at Microsoft (e.g. subsidiaries, events, marketing, ...) and focus on driving usage of the localized technical documentation. We will no longer make GitHub repos public for localized content due to this change. Any existing PRs and issues will be closed, with the exception of azure-docs, which will remain open for community contribution for human translated topics for 6 languages (Simplified Chinese, French, German, Japanese, Korean, Spanish). In addition, starting this summer, our localization pipeline will be updated and will no longer be based on GitHub repos, which will make the current community contribution process obsolete.

We still respond on-demands feedback of the localized content. Click https://aka.ms/devrellocfeedback to create ADO feedback ticket or refer details about Provide Feedback to DevRel Localization Team.

Will you localize legal content, such as our policies or license information?

No. Any content with legal, policy, or compliance implications are outside the scope of the Global Experiences team. This content needs to go through the CELA team first and is localized by special contractors. Individual product teams incur this cost. For detailed information, see our localization guidelines for legal articles.

CELA hosts pre-localized legal text in a separate repo. When possible, you should point to existing legal content that has already been localized. This includes:

- End User License Agreement (EULA)
- Terms of Service (TOS)
- Terms of Use (TOU)
- Licenses
- Privacy statement
- THIRD-PARTY SOFTWARE NOTICES

> **IMPORTANT**
>
> Don't put legal documents in a technical documentation repo. They will be translated, which can have legal implications for the company. If you plan to publish legal documentation, regardless of localization needs, please see Legal content.

My service will onboard to Azure China (code name Mooncake). Can you help release the content to docs.azure.cn?

Yes. DevRel Global Experiences team handles content adaptation, localization, and release to Azure China docs.

The adaptation refers editing Azure docs to be updated according to Azure China features, and Azure branding will be adapted for Chinese market. All reference site links and reference article links will be validated for the Chinese end users.

The adptation team monitors Cayman to identify the newly onboarded service to Azure China, synchronizes the global content from the Azure repo live branch three weeks before the launch date, they contact the service Engineering PM or you to understand the specific Azure China feature. The adaptation takes one or two weeks depending on the article number or the adaptation complexity. Then the localization takes one week.

For any previously onboarded service to Azure China, the adaptation team also maintains the content and checks for updates every two weeks.

Our Azure China content structure aligns with the global Azure version, however we don't support extra articles or TOC entries in Azure China docs. Contact Sunny Deng with any questions.

Learn localization

Which languages are supported for Learn localization?

Learn Azure

Almost all Learn Azure and derivatized Azure modules are localized in 17 languages, which cover learn-pr repo folders of \azure, \advocates, \ai-school, \aspnetcore, \cmu-cloud*, \commercial-marketplace, \data-ai-cert, \databricks, \github, \iot, \language, \quantum, \student-evangelism, and \reactors etc.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, Simplified Chinese, Czech, Dutch, Hungarian, Swedish, Turkish, Polish, and Portuguese (Portugal)

> **NOTE**
>
> Due to the popularity of some specific modules, we also support additional languages: all the modules in **Azure Fundamental** learning path are localized in Danish, Finnish, and Norwegian; **Introduction for Python** are localized in French (Canadian), Hindi and Arabic.

Learn Windows

All Learn Windows modules in \windows folder of learn-pr repo are localized in 10 languages.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, and Simplified Chinese

Learn SQL Server

All Learn SQL Server modules in \sqlserver folder of learn-pr repo are localized in 10 languages.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, and Simplified Chinese

Learn Xamarin

All Learn Xamarin modules in \xamarin folder of learn-pr repo are localized in 10 languages.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, and Simplified Chinese

Learn Power BI

All Learn Power BI modules in \powerbi folder of learn-bizapps-pr repo are localized in 23 languages.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, Simplified Chinese, Hungarian, Swedish, Czech, Dutch, Iberian Portuguese, Polish, Turkish, Danish, Greek, Slovak, Finnish, Thai, and Norwegian

Azure localization

Which languages are supported for Azure docs localization?

We localize 17 languages for Azure docs, the language set is the same as Ibiza portal language set.

- **APEX10**: German, French, Italian, Spanish, Russian, Portuguese (Brazil), Japanese, Korean, Traditional Chinese, and Simplified Chinese;
- **APEX13(+3)**: Czech, Polish, and Turkish;
- **APEX17(+4)**: Dutch, Hungarian, Portuguese (Portugal), and Swedish.

Do you localize articles and includes for *all* Azure services/folders in the same language sets?

We localize for all Azure services in the same language sets except the following content sets:

- The content in Azure Government Document stays in English for all languages.
- The content in Azure Germany Document stays in English for all languages except German.
- The content in Azure China 21Vianet Document stays in English for all languages except Simplified Chinese.

For all other services that we do localize, we prioritize articles by topic type (see next FAQ).

How do you prioritize the articles to localize for Azure docs?

We use the article's metadata **ms.topic** value to prioritize the articles for the localization language set and the release cadence.

- **Azure High Impact** content also known as **Azure HI**: refers to the article *ms.topic* value in either 'overview', 'quickstart', 'tutorial' or 'sample'.
  - For several non_MVC services, that is, nonstandard content, **Azure HI** also refers to the *ms.topic' value is either 'hero-article' or 'get-started-article'
- **Azure NON High Impact** content also known as **Azure nonHI**: refers to the other value of *ms.topic* of the articles.

When do you start to localize my article?

We currently pick up the English updates from the azure-docs-pr repo **LIVE branch** only. We schedule the hi-pri handoff at **every Tuesday 6:00pm PST** when we kick off the translation process.

- If your article stays in your own branch, the localization system can't detect the updates, thus there will be no localization.
- If you publish the article before Wednesday 3:00pm PST, then we pick up your article in that Wed handoff, and deliver to the translators.
- If you publish the article after Wednesday 3:00pm PST, then we pick up your article in the *next* Wed handoff, and deliver to the translators one week after.

How long will it take for the localized version to be published, for example, in Japanese?

As long as the localized files are delivered back in loc pipeline, it takes no more than 30 minutes for the content to be published live.

Azure docs is a CPUB project, we pre-defined the delivery cadence.

- For the **Azure HI** articles, the delivery is in five working days after the handoff, thus the next Wednesday after the handoff;
- For the **Azure nonHI** articles, the delivery is in 10 working days after the handoff, thus the second Wednesday after the handoff.
- For all the TOC and Index pages, the delivery is three workings days after the handoff.

Where can I find Simplified Chinese articles?

There are two Simplified Chinese versions for Azure docs:

- The global Simplified Chinese version document is translated from Azure docs English content, you may go to the global zh-CN version: `https://docs.microsoft.com/zh-cn/azure/index`
- The Azure China technical document is adapted from the Azure docs English content according to Azure China functions, you may go to Microsoft Azure 21st Vnet.

DevRel Authoring Checklists for Localization

7/7/2021 • 7 minutes to read

This article describes a series of checklists and guidelines for writers to keep in mind for localization in most common scenarios. Use these checklists and guidelines to work with your localization contact as you plan for new content or changes to existing documentation.

Who is your Localization Contact?

Find your Localization PM (loc PM) contact here.

Ask Loc Team Lead Sonja Saltzman if you aren't sure who your loc PM is from link above.

New Products or Services

Contact your loc PM in advance before publishing content for new products or services.

- Is your content using any new product names, service names, branding names, or terminology?
- Is there a new repository created? Be sure to @ mention your loc PM in the repo onboarding ticket for visibility.
- Is there a new docset created?
- Is there a new sub-folder created in the existing repo?
- When is the English release date?

Major release

Contact your loc PM before a major content releases such as:

- **BUILD**, **Ignite** and **Connect**
- Major feature release with large update at one time
- Rebranding of product or Azure service

- Is there any confidential content that should **not** be localized? For example, **GA** or **pre-release**.
- Is there a release branch to be created for the content manager to sign off before merging to the main branch?
- Are there any files being moved among sub-folders?
- Keep term changes to a minimum including renaming of product, feature, or service.
- Is your content using any new product names, service names, branding names, or terminology?
- Are there any branding changes?
- When is the English release date?

SIM-SHIP

SIM-SHIP means to release or publish English and localized versions simultaneously, or within 24 hours on Microsoft Docs. We usually don't support SIM-SHIP unless there's a mandatory requirement from the country regulation, the security risk, or any legal obligation. We don't support it because SIM-SHIP requires more manual work and coordination.

To support SIM-SHIP, the loc team must be notified **six weeks** before the SIM-SHIP day and the English **Content Complete** day must meet loc team's request.

- Is there a strong business justification to request SIM-SHIP?
- File SIM-SHIP loc request ticket.

Content Migration

Contact your loc PM before you migrate your content to a new web site, repo, or publishing environment:

- Is your content migrating from one repo to another? If so, is the new repo in same or different GitHub org?
- Are there any files being moved among sub-folders?
- Are there any new or renamed docsets?
- Is there any content being archived?

Doing something different, new, or experimental

Contact your loc PM before using new or updated features.

- Are there any file format changes for index, includes, Leanrning Path, or Module Unit files? For example, .md changes to .yml.
- Is there any content using a new file format to the existing repo?
- Is there any new or updated UHF, Zone pivots, or tabbed conceptual content?
- Is there any new or updated content for an AB test?

Content not on docs.microsoft.com

Contact your loc PM if the content you want localized isn't publishing to docs.microsoft.com. The Loc team will investigate and determine the loc approach, which is different from the Docs localization. For example, loc approaches like in-box release, GB18030 documents, Intellisense, blogs, and so on.

- What is the file format?
- Where is your content published or hosted?
- When is the Content Complete date?
- When is the release date?
- How frequently will this content be updated?
- Who is the content owner to update the content?
- Who is the publishing owner that the loc PM may contact to setup the localization pipeline?

Legal documentation

Legal documents are made up of content from the following content types:

- EULA
- Terms of Service
- Terms of Use
- Licenses
- Privacy statement
- THIRD-PARTY SOFTWARE NOTICES

Legal documents aren't localized by the same team and processed as technical documentation.

- Is there any legal documentation living in the technical document repository? If so, **REMOVE them**!
- Is there any legal documentation to be localized? See Legal article localization guideline.

Common mistakes

The following sections describe mistakes that seem fine for the English content but will break or affect localization.

English language content repo shouldn't contain any content not published to LIVE site

The localization pipeline automatically detects file changes in the live branch of your content repo. Localizable files are handed off for translation unless you configure the files to be excluded. To avoid unnecessary translation costs, use of pipeline bandwidth, and translation resources, remove any content that isn't being published from your repo.

- Are there any art files, media files, module units, articles, or includes files in your content repo that aren't referenced or published to the LIVE site?

Unprotected code

- Use correct syntax to prevent over-translation of code, command line, API names, and so on.
- See How to include code in docs.

Broken includes

- Use correct Includes syntax!
- Not doing so will leave the content in the Includes file unlocalized. Your article will display a mix of localized content with content from Includes files displayed in English.
- *INCLUDE* must be ALL CAPS, with a space after it. Like this: `[!INCLUDE [<title>](<filepath>)]`

Don't use "en-us" in URLs

- Don't hardcode locale codes such as `en-us` in your links.
- This rule applies to links to docs.microsoft.com or Azure.com where the language selection UI is available on the target site.
- See Locale codes in external links.

Don't use "aka.ms" links

- Don't use aka.ms links.
- See Don't use aka.ms link.

Make sure metadata "ms.topic" match the Standard ToC

- Is your content using the Standard TOC?
- If so, keep the metadata `ms.topic` (overview/quickstarts/tutorials/sample) match the ToC nodes (Overview/QuickStart/Tutorials/Sample).
- Besides being used for content metrics reporting, loc uses article type to determine whether to use machine or human translation services.

Incorrect metadata

- Keep your `author`, `ms.author`, and `manager` metadata current and correct.
- Everyone uses these values to quickly identify the article owner. The Loc team depends on this information for when they have a translation bug or question.

Remarks on art and media files

When creating new badges or trophies for Learn content or icons or other conceptual images for technical documentation, ensure the art pieces aren't offensive or their intent isn't misunderstood in certain markets. Some images aren't localized (badges, trophies, icons), but some are potentially localized (screenshots, conceptual images):

- Are trophies, badges, and icons globalized?

- Review the checklist created and maintained by the CELA team: Image Review Checklist to ensure your images are acceptable worldwide. If you have further questions or aren't sure about your specific image, contact your global readiness contact.

- Are art files linked correctly? See Feature Usage.

- If art syntax is incorrect, then the localization platform won't recognize that art file as a dependency to the article. Then, that art file won't get localized when it should be.

- Is the update to conceptual art/image files necessary?

- Art files take longer and is more expensive than text to localize.
- Any update to an art file is same as adding a new art file from localization perspective.

- Are there any duplicate art files saved in different file name in different folder or in different file format?

- Avoid duplicate art files to reduce localization cost.

- Is renaming art file necessary?

- Don't change between **upper case** and **lower case** in file name. It causes the localization pipeline pending where a manual data fix is needed.
- Our pipeline can't differentiate such changes and will consider the renamed file as a new art file.

- Are the UI strings in content consistent with the UI strings in screenshot?

- If there's inconsistency between the text and screenshot in the content, it won't only confuse the translator, but also the customers. And fixing those inconsistencies later will also increase localization cost.

- Avoid using dynamic gifs.
- The video should be published on RedTiger where localization tool supports.

Project-specific requests

Here are some project-specific requests for each product family.

Microsoft Learn

Contact your loc PM if your Learn content meets following scenarios:

- New Module Group is created.
- Module is shared across multiple repos.
- Content in one repo is shared between different localization groups.

Azure Docs

- Are all **service names** using title case and capitalized consistently?

- This request is critical for localization. This request is critical especially when the product or service name shouldn't be translated, but a common noun of the same term should be translated. Inconsistency in capitalization won't only confuse translators, but also confuse customers.

- Is the new service metadata added to allowlist through new ms.service value request form?

- This request is critical when loc PM reviews BI data (for example, Page Views) to break down the BI data based on correct service metadata.

SQL Server / Dev Tools & Services

Contact your loc PM when any major refactoring is done to existing source content.

- Adding new folder, renaming folder name, or moving content from one folder to another can cause the files in those folders to be unlocalized.
- The folder level change could also cause human translated content to be machine translated, or vice versa.

Windows dev content

- Is `ms.localizationpriority` metadata up-to-date?

- The `ms.localizationpriority` metadata is used to define the translation quality only when the Loc team and content owners agree beforehand. The standard practice is if it's `high`, we'll localize in human translation quality. If it's `medium` or `empty`, we'll localize the content as machine translation quality.

Localization guidance for Docs writers

6/22/2021 • 2 minutes to read

Do not force-push files in the en-US source repo

When the en-US source repo is force-pushed (`git push -f <remote> <branch>`), there can be conflicts. These conflicts, while unresolved, can block the syncing of the localized repo structures.

The OpenLoc localization tool (OpenLoc) is constantly monitoring the specified branch (typically /live branch for cpub projects, /master for milestone projects) and any file moves or deletions performed by writers on the en-US repo are then automatically done on the loc repos for all locales. If there are repo merge conflicts on the english repo, the loc tool is blocked from working properly.

If you ever really need to force push in the en-US repo, please contact your DevRel Content Localization Production PM Team first, so we can involve the dev team to support this on the loc repo side without breaking our tooling.

What are impacts of source repo Force-Push?

- Git conflicts when syncing with remote source repo. OpenLoc Handoff and structure-syncing are blocked until conflicts were manually resolved.
- Last processed commit in OpenLoc cannot be found in current force-pushed live branch. OpenLoc cannot calculate revision number based on last processed commit.

How to fix Manually resolve merge conflicts in all OpenLoc local repos Fix all OpenLoc file revision data according to current force-pushed live-branch.

NOTE: Avoid Force-Push unless necessary Force-push in the en-US source repo requires heavy data-fixing on the OpenLoc system data. It should be avoided unless absolutely necessary.

Localization of legal content

7/7/2021 • 4 minutes to read

The Global Experiences Content Localization team doesn't manage localization of any product- or service-specific legal content. The current loc supplier (MOR) is approved for technical documentation translation only under the Content & Learning SOW.

The exceptions are site legal documents for the Microsoft Azure site and the Microsoft Docs site. There's a standard template for the en-US EULA legal document, which already exists in most or all of the localized locales. If the standard template is used, then you don't need to arrange for localization of another document.

Only the LCA approved translation vendors can meet the legal translation quality. **Product teams who require localization for new legal content must coordinate and fund translations using certified vendors.** Before you start this process, work with your CELA representative to confirm the need for your new legal documentation and corresponding translations.

What is considered legal content

The following document types are legal content:

- End User License Agreement (EULA)
- Terms of Service (TOS)
- Terms of Use (TOU)
- Licenses
- Privacy statement
- THIRD-PARTY SOFTWARE NOTICES
- Any other content that requires CELA review or that poses a legal risk to the company.

For more information about identifying, scoping, or publishing legal content, see Legal content.

Legal localization process

Content owners are responsible for coordinating and funding the translation of legal articles as needed. Once translated, you submit a work item to Content Production Services (CPS) to merge and publish the translated files in each localized counterpart of the DocsLegal-pr repo. Use the following steps to request translations and publish your localized legal articles.

> **IMPORTANT**
> Before you start this process, make sure your updates have been published to the main (en-us) legal repo MicrosoftDocs/DocsLegal-pr. The localization repos are configured to automatically remove any files that are not mirrored in the en-us repo.

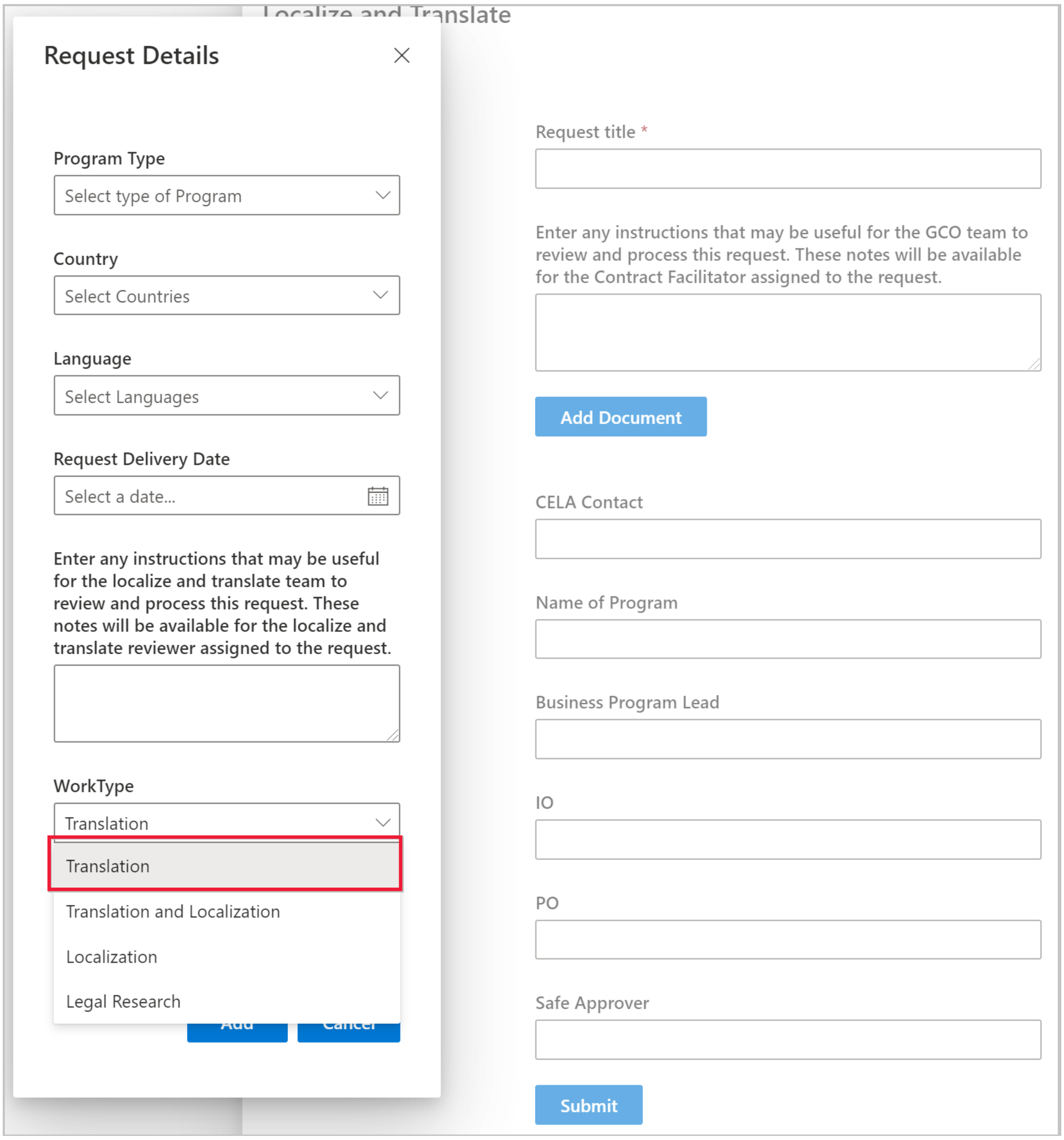
1. Confirm necessary locales for translation with your CELA representative. In some cases, there's no requirement to localize a legal article.

2. Submit a localization request via the Legal Contracting Experience portal. This process includes paying for the translation work.

   a. Under CONTRACTING HELP, select **Localize & Translate**.

b. Complete the form fields.

c. When you choose **Add Document**, the Request Details form appears. Complete the fields, specify the countries and locations, and for **WorkType** choose Translation.



3. When the translations are ready, use this template to create a work item for CPS. The template includes instructions to attach the translated files and confirm the locales and folder locations.

> **NOTE**
>
> The publication SLA is 48 hours. CPS will assume that this content is ready to go live when you create the work item.

4. The CPS vendor will commit the articles to each repo and then create a PR from the main branch to the live branch to validate the build before publishing. The vendor or the Onboarding team might contact you in the ticket or a PR if they need help with resolving an issue.

5. When the vendor completes the work item, review the published pages to make sure they render as expected.

If you have any questions, contact the CPS leads.

FAQ

- **Can I save legal documents in my product repository?**

  No.

  If you save legal documents in a product repo, such as sql-docs-pr, the current loc system automatically detects those files and queues them for localization by the technical document translation vendor. As explained above, the translation isn't legally valid.

  Instead, you must host the legal document in the DocsLegal-pr repo. See Legal content for detailed requirements and publication steps.

- **Can I host all languages/localized versions of a legal document in the one DocsLegal-pr repo?**

  No.

  The publishing system (OPS) defines the language publish per repo. If the article is to be published in multiple languages, you must save each translated article in its respective localized repo. The good news is we have already set up 68 localized Docs-Legal-pr repositories.

- **Which languages should I target for localization?**

  Work with your CELA representative to confirm your legal localization needs. You may need to localize legal documents for the same languages that your product or service is localized.

List of localized DocsLegal-pr repos

There are 68 localized versions of the DocsLegal-pr repo:

- DocsLegal-pr.ar-sa - Arabic
- DocsLegal-pr.eu-es - Basque (Basque)
- DocsLegal-pr.bs-Cyrl-BA - Bosnian-Cyrillic (Bosnia and Herzegovina)
- DocsLegal-pr.bs-Latn-BA - Bosnian-Latin (Bosnia and Herzegovina)
- DocsLegal-pr.bg-bg - Bulgarian
- DocsLegal-pr.ca-es - Catalan (Catalan)
- DocsLegal-pr.zh-hk - Chinese-Hong Kong
- DocsLegal-pr.hr-hr - Croatian
- DocsLegal-pr.cs-cz - Czech
- DocsLegal-pr.da-dk - Danish
- DocsLegal-pr.nl-nl - Dutch
- DocsLegal-pr.nl-BE - Dutch (Belgium)
- DocsLegal-pr.en-AU - English (Australia)
- DocsLegal-pr.en-CA - English (Canada)
- DocsLegal-pr.en-IE - English (Ireland)
- DocsLegal-pr.en-my - English (Malaysia)
- DocsLegal-pr.en-nz - English (New Zealand)
- DocsLegal-pr.en-sg - English (Singapore)
- DocsLegal-pr.en-za - English (South Africa)
- DocsLegal-pr.en-in - English India
- DocsLegal-pr.en-gb - English-United Kingdom of Great Britain and Northern Ireland
- DocsLegal-pr.et-ee - Estonian (Estonia)
- DocsLegal-pr.fil-PH - Filipino (Philippines)

- DocsLegal-pr.fi-fi - Finnish
- DocsLegal-pr.fr-fr - French
- DocsLegal-pr.fr-BE - French (Belgium)
- DocsLegal-pr.fr-CH - French (Switzerland)
- DocsLegal-pr.fr-CA - French-Canada
- DocsLegal-pr.gl-es - Galician (Galician)
- DocsLegal-pr.de-de - German
- DocsLegal-pr.de-at - German (Austria)
- DocsLegal-pr.de-CH - German (Switzerland)
- DocsLegal-pr.el-gr - Greek (Greece)
- DocsLegal-pr.he-il - Hebrew (Israel)
- DocsLegal-pr.hi-in - Hindi
- DocsLegal-pr.hu-hu - Hungarian
- DocsLegal-pr.is-is - Icelandic
- DocsLegal-pr.id-id - Indonesian
- DocsLegal-pr.ga-IE - Irish-Ireland
- DocsLegal-pr.it-it - Italian
- DocsLegal-pr.it-ch - Italian (Switzerland)
- DocsLegal-pr.ja-jp - Japanese
- DocsLegal-pr.kk-kz - Kazakh
- DocsLegal-pr.ko-kr - Korean
- DocsLegal-pr.lv-lv - Latvian
- DocsLegal-pr.lt-lt - Lithuanian
- DocsLegal-pr.lb-L - Luxembourgish-Luxembourg
- DocsLegal-pr.ms-my - Malay
- DocsLegal-pr.mt-MT - Maltese-Malta
- DocsLegal-pr.nb-no - Norwegian (Bokmål)
- DocsLegal-pr.pl-pl - Polish
- DocsLegal-pr.pt-br - Portuguese (Brazil)
- DocsLegal-pr.pt-pt - Portuguese (Portugal)
- DocsLegal-pr.ro-ro - Romanian
- DocsLegal-pr.ru-ru - Russian
- DocsLegal-pr.sr-cyrl-rs - Serbian-Cyrillic-Serbia
- DocsLegal-pr.sr-latn-rs - Serbian-Latin-Serbia
- DocsLegal-pr.zh-cn - Simplified Chinese
- DocsLegal-pr.sk-sk - Slovak
- DocsLegal-pr.sl-si - Slovenian
- DocsLegal-pr.es-es - Spanish
- DocsLegal-pr.es-MX - Spanish (Mexico)
- DocsLegal-pr.sv-se - Swedish
- DocsLegal-pr.th-th - Thai
- DocsLegal-pr.zh-tw - Traditional Chinese
- DocsLegal-pr.tr-tr - Turkish
- DocsLegal-pr.uk-ua - Ukrainian
- DocsLegal-pr.vi-vn - Vietnamese

How do we select content to localize

1/22/2020 • 2 minutes to read

Selecting content for localization

There is no hard and fast rule to how the localization team chooses content to localize, and into which languages to localize the content. For each project, the MSFT loc PM is carefully evaluating before making a decision. The loc PM typically works closely with the content manager/writing team, to find out which areas of the content are most important (e.g. If we cannot localize the whole content set, should we localize getting started vs. Some other content buried deeper in the TOC?).

We take many factors into consideration; the most common are:

- **Legacy.** Was the product localized before? To which locales? We typically support a certain language set per pillar (e.g. all Azure-related content is localized to 17 locales)
- **BI.** Is the content used by many international customers? Should we expand/contract the language set?
- **Importance/Visibility of content.** In discussion with content owners, which areas of content are more important?
- **Software/UI.** Into which languages is the software/UI localized? Should we match the language set, or should there be more/less languages supported for content, and why?
- **Budget.** What can we afford, what tradeoffs can/should we make? There are basically 2 levers, which are the amount of files and amount of languages, plus the localization quality (human translation vs. Machine translation).
- **Tools.** We can support markdown and yaml file localization of files living in Github repos. If you have different file types and CMS, please contact the localization PM to discuss.
- **Related projects.** Are components of our content used by partner teams, e.g. Office/Windows? Do they have a different language set we need to support?

Reference content and Intellisense

- As a rule, **reference content is localized with TM/MT**. This means, the translation memory for a project is used (translation memory always contains human translated strings) to recycle matching sentences, and the non-matching sentences are then machine translated.
- The exception is Intellisense. For most projects where Intellisense is shipped as a downloadable or the Intellisense xmls are checked into the software build we will provide **human quality translations for the intellisense-strings,** depending on schedule and budget availability.
- For **new dev languages** (e.g. Python, java, ...) which we traditionally have not localized, there may be **advanced notice** and tooling changes required before we are able to provide localized content.

Images and Videos

Image and video are very costly to localize and require additional turnaround time from our suppliers, so any decision is made on a project-by-project basis. If you believe that some of the images/videos in your content are especially important to the customers, please advise the loc PM. As a general rule, we localize images and video subtitles for all Learn content, but only for subsets of technical documentation.

Source escalation for localization issues

7/7/2021 • 2 minutes to read

What is source escalation?

Sometimes people find issues in the English language source content need to be corrected. The localization team usually escalates the issue to an English writer for resolution. This process is called a source escalation.

Escalate source issues

We've agreed with the DevRel Content managers that the localization team may escalate English language content issues or localization questions to the DevRel writing vendor team directly. The team may also raise issues in the English language source content `private` repository. Escalation to the English language source content `public` repository only applies to projects where a `private` repository doesn't exist.

The following process applies to all DevRel product repositories except SelfHelpContent, TechProfile, and VS Recon.

Simple source issues

When source issues are simple and the Loc team knows how to fix them, the loc PM will create a PR in the default branch of the English repository. The loc PM verifies the fix on the staging site. If no error, then the loc PM will comment `#sign-off` and mention the article author.

- If the PRMerger is enabled on that repository, then `#sign-off` kicks off the PR merge automatically.
- If the PRMerger isn't enabled on that repository, then `#sign-off` reminds the repository admin to merge the PR.

Types of issues that can be escalated as GitHub PRs are limited to:

- English typos. Always confirm the word spelling from the Merriam-Webster dictionary.
- Syntax errors that block localization handoff or localization rendering on the Docs site.

Complex source issues or localization queries

The loc PM isn't always sure how the source issue should be fixed or if it needs a clarification to translate. When the loc PM isn't sure how to continue, then the loc PM will create a work item for the DevRel vendor editorial team through DevRel loc-fix template.

Types of issues you can escalate as a DevRel localization fix work item include, but aren't limited to, the following types:

- Ambiguous sentences in a specific language.
- Ambiguous technical concepts related to a term, a sentence, or a paragraph.
- Markdig format syntax errors that occur in multiple lines or multiple articles.
- Grammatical errors.

Once the loc PM creates the work item, the vendor editorial team reviews the work item and makes the necessary updates to the article. If the vendor editorial team can't confidently provide a fix or an answer, then they'll contact the article's listed author.

The SLA response time for a localization ticket is 10 working days excluding Microsoft holidays.

DevRel localization fix work item guidance

- Always use the DevRel loc-fix template to report complex source issues or localization questions.
- Don't change the **Assign To**, **State**, **Area**, and **Iteration** values in the template.
- For every new work item, always modify `\<summary description\>` according to the request in the title before saving.
- The ticket owner will change the **State** from **Proposed** to **Committed**, then **Completed** or **Cut**.
  - If the **State** becomes **Completed**, then the **Reason** will be either **Done** or **Duplicated**.
- Before creating a new escalation work item, always check for a duplicate work item. If there's a similar work item, link it to the old work item.

Writing for translation

7/7/2021 • 2 minutes to read

How you write English source content can significantly affect the quality of its translation. Simplified English is easier for non-native English speakers to understand, and translates better into different languages.

Here are some general guidelines on how to write for translation and help to improve the quality of both human translation and machine translation.

Simple guidelines

Fortunately, the most fundamental rules are easy to remember:

- Use short sentences.
- Use correct spelling, capitalization, and punctuation.
- Use precise language.

More guidelines

Here are more detailed guidelines for writing content that's easy to translate into other languages (that is, 'globalized' English):

- Split long sentences into several shorter sentences whenever it's logically possible.
  - Split compound sentences into two or more simple sentences.
  - Make a "which" clause its own sentence.
  - For long instructions, consider restructuring the content into a bulleted list.
  - Short sentences should be typically about 15-20 words.
- Use standard English word order:
  - Subject + Verb + Object
  - Avoid inverting the subject and the verb. For example, Never have I..., No sooner had they..., and Not only did they....
- Use simple verb tenses: simple present, simple past, and simple future.
  - Avoid the use of progressive verb tenses am/is/are, was/were, will be.
  - Avoid the use of perfect verb tenses have/has, had, will have.
- Don't use a weak "to be" verb when you can use a stronger verb.
  - Example: "Work will be finished" should be "Work will finish."
- Use active voice instead of passive voice.
  - Example: "Work was completed" should be "The team completed the work."
  - Exception: It's fine to use passive voice to avoid assigning blame to a user, or when the actor is unknown or immaterial.
- Use commas correctly.
- Use semicolons sparingly. It's better to create two sentences.
- Don't use - or / as punctuation. For example:
  - "Run the app and verify you can create/edit/display the page." should be "Run the app and verify you

can create, edit, and display the page."
  - "Run the app - verify you can create a page." should "Run the app and verify you can create a page."
- Avoid parenthetical clauses.
- Use abbreviations or acronyms sparingly.
- Use words and phrases consistently.
- Don't use nominalizations.
  - For example, "The rapid increase in incident tickets is becoming a concern" should be "Incident tickets are rapidly increasing and becoming a concern."
- Avoid noun stacks.
  - For example, "Install the incident tickets monitoring app" should be "Install the app to monitor incident tickets."
- Avoid stacking prepositional phrases on top of one another.
  - For a list of prepositional phrases, see If a list of Prepositional Phrases is what you need, look no further.
- Avoid wordiness and jargon, like English clichés and slang.
  - Technical prose translates better than marketing-style language, jokes, or wordplay.
- Include all words that will clarify a sentence.
  - Include *that* and *who*. They help to clarify the sentence structure. For example, "Select the check box of each folder *that* you want to sync with your desktop."
  - Don't omit articles and other little words from parallel structures.
- Don't use possessives for product or feature names.
  - For example, don't use "Azure's" or "Spell Checker's." "Azure's security features include" should be "The security features of Azure include."
- Don't append "s" to make types or APIs plural.

Resources

- Microsoft Writing Style Guide for localization

Provide feedback to DevRel localization team

7/9/2021 • 2 minutes to read

This article is for anyone who wants to provide feedback to the DevRel localization team. The DevRel localization team uses CEINTL DevOps to track various issues and requests with their loc vendor. Anyone with a Microsoft alias can request for access to CEINTL DevOps and open a feedback ticket. Feedback can be about translation quality or any suggestions for localization.

> **NOTE**
>
> The **external contributor** who doesn't have an MS alias can download the email template by typing **aka.ms/docsitelocfeedback** in browser URL, then send the feedback through email. The feedback email will be saved as an ADO ticket in CEINTL DevOps. Our team will monitor and process the feedback.

Access to CEINTL DevOps

If you don't have access to CEINTL DevOps, then you can request for access to the `CEAPEX Work Items Access` project from MyAcess.

How to fill in feedback tickets

When opening a feedback ticket, use DevRel localization feedback template. Or, you can type aka.ms/devrellocfeedback in the URL to create a new feedback ticket. Providing the following information will help us to understand your feedback better and speed up the process:

- **Assigned To**: By default, feedback tickets are assigned to *MS International Feedback Team* account, MS Feedback PM assigns the ticket to the right loc PM.
- **Description**: Provide the following information per suggestion:
  - Language for which the issue is found
  - Technology where the issue is found
  - What is the URL where the issue is found?
  - Brief description of the issue
  - Steps to reproduce
  - Screenshot attached (Yes/No)
  - Current translation (if it's a linguistic issue)
  - Expected translation (if it's a linguistic issue)
  - Corresponding English text (if it's a linguistic issue)
  - Comments to the localization team
- **Project Type**: Select if your feedback is for **Software**, **Content**, or **Sites** localization. If your feedback affects multiple project types, then select **Cross-Team**.
- **Issue Found By**: Select the source of your feedback. For example, if you're a customer relationship PM and you received feedback from an MVP, then select **MVP**.
- **Language Origin**: Select the language your feedback was originally found in.
- **Affected Languages**: Select all the languages affected by this feedback. This will helps us understand the scope of your feedback.

Create a feedback ticket via email

From time to time, you may receive feedback from your partners via email. In this case, we strongly recommend for you to forward those e-mails to CEINTL e-mail alias. The email will automatically create a feedback ticket in CEINTL DevOps. Then you should receive a confirmation email from Mail2Bug bot within 2 minutes including a link to the feedback ticket.

You can use mnemonics (@) and pound (#) characters to set fields in the feedback ticket to be created:

> **NOTE**
>
> - As best practice, always include an example or attach a screenshot for the feedback.

How to Avoid Over-Localization

7/20/2021 • 3 minutes to read

What is Over-Localization?

Over-localization is a localization bug when certain term or keyword that should be left in English for localized languages is translated by error. Example of terms or keywords that shouldn't be translated are:

- Product/service/branding names (for example, "Azure DevOps", "Xamarin.Forms")
- File names or folder paths.
- Code elements or programming languages keywords such as:
  - Class/function/property names (for example, "CAccelerateDecelerateTransition")
  - Predefined macros (for example, "__cplusplus_cli")
  - C# keywords/operators (for example, "static" or "default")
  - HTML/XML tags (for example, "<element>")

Over-localization may happen as human error for human-translated (HT) topics. For machine-translated (MT) topics, there's limitation of the MT engine. Root cause can also be considered as source issue in English content. As an example, if a programming language name "Python" isn't properly capitalized like "python", then that typo could affect both HT and MT. Both human translator and MT engine may translate it as a type of snake instead of the programming language. Content writers can help to avoid these issues by using certain Markdown syntax to block terms/keywords from being over-localized.

How to Avoid or Fix Over-Localization

Here are current solutions in which content writers can help to avoid over-localization issues:

- For new product/service/branding names, or keywords that shouldn't be translated, create a Localization Request ticket. Loc PMs will make sure those terms are added to Term Studio with an approved term to use for each language.
  - For HT topics, human translators will reference the approved terms in Term Studio.
  - For MT topics, terms added to Term Studio will be added to MT dictionary, which is then used to retrain the MT engine on quarterly basis.
- For new and existing product/service/branding names, be sure to use proper capitalization so they're correctly recognized.
- Use inline code style or code blocks for programing language keywords, file names, or folder paths.
- Use the no-loc Markdown extension inline or using metadata in YAML header.
- Add the loc-scope attribute to screenshots with a different localization scope than the article or module that contains them.
- For HTML/XML tags (for example, <element>), feature was deployed as of June 2020 to protect these escaped <element> tags from being over-localized.

> **NOTE**
>
> Due to this feature, placeholder text within angled brackets (for example, <placeholder text>) will be blocked from localization. Placeholder texts in angled brackets can be considered part of code examples, which we do not localize. However, in order to have placeholder text localized, writer can use Italic font instead of angled brackets as recommended in the style guide.

How to Report Over-Localization Issues

When you find any over-localization issues, first consider one of the options listed above on how to avoid or fix them. If you have any questions about those options, create a Localization Request ticket to get support from a loc PM. If you receive GitHub ISSUE from customer in English repo for over-localization issue, then you may use @ mention to notify your localization PM for assistance. Otherwise, you can also submit any feedback for localized content by opening a Feedback ticket.

> **NOTE**
>
> For HT topics, you may open GitHub PR or ISSUE in the affected localized repo. However, Community Contribution (GitHub PR) or Feedback feature (GitHub ISSUE) are currently not accepted for MT content. MT topics have a disclaimer stating that part of the topic may be machine translated, and the **Edit** button won't be displayed as we have disabled it for MT topics. We would still welcome any internal feedback on MT over-localization issue, so please don't hesitate to either create a Localization Feedback ticket.

Related topics

- Get to know the Global Experiences Content Localization team
- Frequently asked questions (FAQ) about localization
- DevRel Authoring Checklists for Localization
- Provide Feedback to DevRel Localization Team

How to specify the localization scope for an image

3/10/2021 • 2 minutes to read

The localization scope is the set of locales a product or content set is localized into. For example, most Azure services and their corresponding documentation are localized into 17 languages, but Visual Studio is localized into 13 languages.

Sometimes an image in an article or Learn module shows a product or service that has a different localization scope than the main subject of its parent. For example, you might include a screenshot of Visual Studio in a module about Azure. Most of the content of the module should be localized into Azure's 17 languages, but the screenshot should only be localized into the 13 languages supported by Visual Studio. Otherwise, some international users would see misleadingly localized screenshot.

To prevent the over-localization of images, use the `loc-scope` attribute on the `:::image:::` extension for any image with a different localization scope than the content that contains it. This is particularly important for Learn modules, which often include screenshots from multiple products.

loc-scope syntax

The `loc-scope` attribute can be applied to any `:::image:::` with `type="content"` (standard conceptual images such as screenshots) or `type="complex"` (complex images that require a detailed long description, such as graphs and architectural diagrams). The `loc-scope` attribute doesn't apply to images of `type="icon"`. Icons are purely decorative and aren't localized.

The value of `loc-scope` must be one of the following:

- A valid product per the Product taxonomy allowlist.
- `third-party` for non-Microsoft products, such as Swagger.
- `other` for images that don't apply to a particular product, such as output screens for example apps.

When should I add a non-default loc-scope?

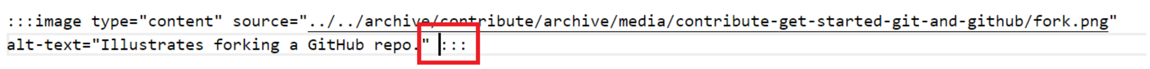
You should specify a `loc-scope` that differs from the default if your image is a screenshot of a different product. For example, if you're showing a VS Code screenshot in an Azure module, you should specify `loc-scope="vs-code"`. For conceptual images, such as a workflow diagram that shows an Azure task with steps involving VS Code, use the Azure default, `loc-scope="azure"`.

Adding loc-scope with the Docs Authoring Pack

The Docs Authoring Pack makes it easy to add images with `loc-scope`.

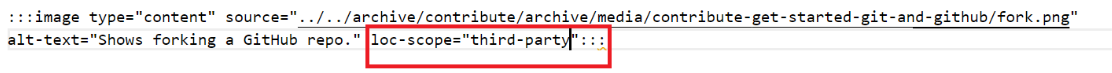
By default, the `loc-scope` attribute isn't included when you insert a new `:::image:::`, because it's optional for most Docs content. To add `loc-scope` to an image, follow these steps:

1. Put your cursor inside the `:::image:::` tag.



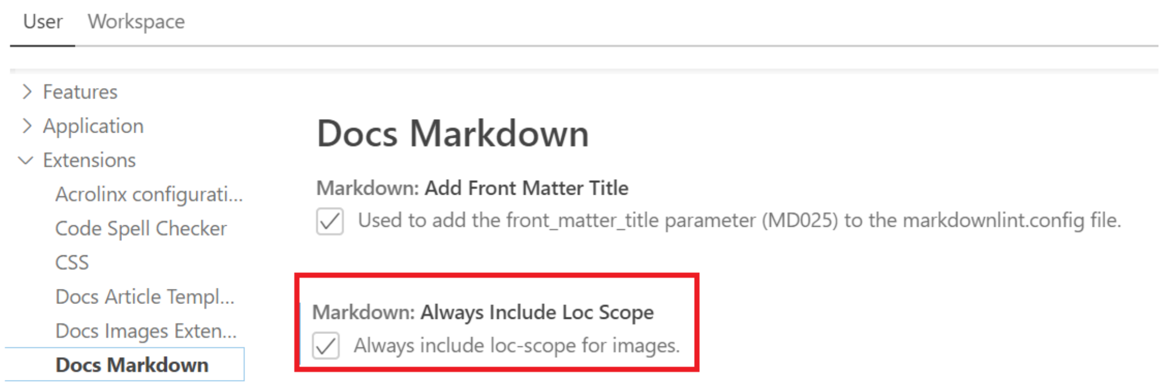
2. From the Docs Markdown menu (**Alt + M**) or the VS Code command palette (**F1**), select **Add localization scope to image**. The valid product list opens.

3. Type to filter the list if necessary, then select the product that applies to the image, or `third-party` or `other` if appropriate. The selected `loc-scope` is added:



Instead, you can update your VS Code extension settings to always include `loc-scope` in images. This is recommended if you frequently work in Learn repos.

1. From the VS Code **File** menu, select **Preferences** then **Settings**.
2. On the **Settings** page, choose **User** to update your settings for all workspaces, or **Workspace** for just the current workspace (this is useful, for example, if you just want to always include `loc-scope` when working in Learn repos).
3. Under **Extensions**, click **Docs Markdown**.
4. Check the box for **Markdown: Always Include Loc Scope**.



Now, whenever you insert an `:::image:::`, you'll be prompted to select a `loc-scope` from the product list. You can still add `loc-scope` to existing images by selecting **Add localization scope to image** from the VS Code command palette (**F1**).

Request localization of video captions

6/28/2021 • 2 minutes to read

The Developer Relations Global Experiences team localizes *video captions* only. If you want to localize audio content, contact the DevRel Content Localization Production PM.

Video management platform

The Microsoft Docs engineering team plans to provide a new video management platform, which could potentially replace the current video player and video management platform. For now, we primarily localize videos on the *RedTiger* platform. For localizing videos on any other platform, contact the DevRel Content Localization Production PM Team.

RedTiger can upload and publish video files with video captions.

- When a video is uploaded as a new asset, it gets a new VideoID as an unique identifier. Each VideoID is associated with multiple elements including the caption files and the language setting. For example, if a RedTiger subsite is configured for ten languages, then any video uploaded to this subsite can have ten language options. One caption file can be uploaded for each language. However, the video file (.mp4) stays the same across ten languages.
- When a video is uploaded to an existing asset, then the VideoID retains its existing properties. This is commonly used when the video or video caption is updated.

Currently, our loc pipeline doesn't detect the Markdown syntax for embedded videos. So we won't know when a new video is embedded in your article. To start localization for video caption, we temporarily need the video owner to create a loc request ticket in a manual way.

Request for caption localization

For any request to localize a video caption, create an Azure DevOps ticket as described here. The loc team will use the Azure DevOps ticket to communicate with the requestor, track any issues found during production, and monitor the loc progress.

Use the Video Captioning Request Template to create a loc request ticket. If you can't access the link, then join the "DevOps_CEAPEX_Partners" security group in IDWeb.

In the ticket, fill out the following information. The first four items are mandatory to start localization.

- VIDEO ASSET ID: Enter the asset ID from RedTiger.
- LANGUAGE LIST: If it's the first time you're requesting video localization for your team, list the languages you want the captions localized in.
- COMPONENT NAME: Enter a product name, such as Azure or Power BI. If you're unsure, contact your localization PM.
- ENGLISH CAPTION: If the English caption files are available, attach them to the ticket. If the files aren't available, see Guidelines for English video captions.
- VIDEO URL: Enter the public URL from RedTiger. It will help us test the video on the video player.
- DOCS URL: Enter the article URL where the video is embedded. It will help us verify that all Docs live pages contain valid localized video.

The loc team will validate the information on RedTiger and contact you with any questions. Localization will start after all questions are answered. The loc team will also estimate the completion date and add it in the **Due Date** field in the ticket.

Caution

The VideoID doesn't indicate any change to video element, so loc team will not know when video is updated. For updated videos, create a *new loc request* for loc team to localize the updated video captions.

After a localized caption is published to the video, the loc team will attach the localized caption and resolve the ticket back to you.

Guidelines for English video captions

3/5/2021 • 2 minutes to read

The localization (loc) team localizes only video captions and not the speech audio, so English video captions are required to localize videos. The English content team should provide the caption files.

Guidelines for captions

The following guidelines are for the English content team or supplier to create English video captions. The loc team will check that these guidelines are followed before localizing the captions. The first three points are required, so the loc team will reject a caption if those aren't followed. The loc team can also help create or correct a caption if requested.

- The caption file is in .ttml or .vtt format. However, Red Tiger only supports .ttml format.
- The caption includes running time.
- There are no midsentence breaks in one time label. (Each caption needs to be a complete sentence.)
- Leave 30% space in the time label for translation wherever possible. (The shorter the string is, the more likely the translated string is to expand.)
- All technical terms are correctly spelled out.
- There are no colloquial terms or phrases.

Guidelines for video creation

The following guidelines are for creating English videos, which helps the localization of video captions:

- The narrator speaks slowly and clearly, for better speech-to-text conversion and to reduce the manual effort needed for editing captions.
- On average, leave 30% additional pauses in the spoken script to accommodate expanded length from localization.
- Avoid colloquial terms and phrases, such as "to be honest" and "frankly speaking."
- Provide scripts to the loc team when you're recording audio, if available.
- Place related videos in one folder (based on product) that the loc team can access.
- Provide a glossary file (any file format works), which contains at a minimum:
  - Technology or product of the video content, such as "GitHub" or "Intune."
  - Product or technology-specific terminology, such as service names.
  - Name of the speaker or event, such as "Scott Guthrie" or "Build."
- Remember that uploading a video takes about six times longer than the video duration. For example, a 2-minute video will take about 12 minutes to upload.

  Contact the RedTiger team before you upload a large number of videos in a short time frame.

Aliases for Azure technical content

4/6/2021 • 2 minutes to read

Frequent contributors to the Azure technical documentation need to belong to certain aliases and be aware of others for information sharing.

ALIAS NAME	ALIAS	REQUIRED?	PURPOSE
Azure technical content authors	wacom-tech-authors	Yes	All contributors to the private repo should join this alias. It is used to communicate publishing status, changes to content strategy and process, and other official updates for Azure technical content. The purpose of the alias is outbound communications only. Only authorized senders can mail to this alias.
Azure docs publishing notifications	azure-docs-publish	No	Receives automated publishing notifications for the azure-docs-pr repository. You need to create a mail rule to filter for items where the subject or body contains the text 'GitHub Branch Name: live' - these are the live publishing runs.
C&L Pull Request Review Team	techdocprs	Only for pull request reviewers.	Use this alias to contact the group of people who review pull requests for Azure tech docs.

Azure documentation feedback using GitHub issues: process for CXP forum moderators

7/7/2021 • 8 minutes to read

The Content & Learning technical content team and program management content owners will collaborate with the Azure Customer Experience (CXP) team to respond to customer feedback for Azure technical documentation.

- Azure CXP will triage all new issues in the azure-docs repo.
- Azure CXP will make their best effort to respond to and close the issues.

  When the CXP team needs help responding to the feedback, they'll assign the issue to the author who's listed in the article's metadata.

The Azure implementation aligns to the GitHub issues documentation feedback policies and guidance. However, because of the partnership with Azure CXP and because of the large scope of the Azure repo, there are specific processes for Azure. This article describes the specific implementation of the new documentation feedback experience for the Azure content set. The team will incorporate the lessons learned into the general guidance as appropriate.

Repositories covered by this process

Azure CXP manages initial triage and also provides issue response and deflection in the following Azure content repositories:

- architecture-center
- azure-cli
- azure-docs
- azure-docs-cli
- azure-docs-powershell
- azure-sphere-issues
- azure-stack-docs

Process for CXP forum moderators

1. Go to the azure-docs Issues page.
2. Run this query:
   - *https://github.com/MicrosoftDocs/azure-docs/issues?q=is%3Aopen+is%3Aissue+no%3Alabel*

   Or, use this query in the issues search:

   - *is:issue is:open no:label*
3. Review each untriaged issue:

   a. Add the **triaged** label.

   b. Assign the appropriate service label to the issue.

   c. Take the appropriate next step, as defined in the table in the next section.
4. Daily, run this query to find issues that an author has indicated can be closed, review, and if appropriate,

close:

- *https://github.com/MicrosoftDocs/azure-docs/issues?utf8=%E2%9C%93&q=is%3Aissue+is%3Aopen+%23please-close*

Or, use this query in the issues search:

- *is:issue is:open #please-close*

5. Daily, run this query to find issues that an author has marked as in process, and assign the in-process-author label:

   - *https://github.com/MicrosoftDocs/azure-docs/issues?utf8=%E2%9C%93&q=is%3Aissue+is%3Aopen+%23in-process*

   Or, use this query in the issues search:

   - *is:issue is:open #in-process*

6. Daily, run this query to find issues where an author or content team has @mentioned the Azure CXP triage team:

   - *https://github.com/MicrosoftDocs/azure-docs/issues/mentioned/azure-cxp-triage*

   Or, use this query in the issues search:

   - *is:issue is:open mentions:azure-cxp-triage*

Process and labeling table

If an issue is assigned to, answered, or resolved by CXP, add the **cxp** label. *Don't* add the **cxp** label if the CXP triager is closing an issue resolved by a PM or author.

ACTION	ASSIGN LABEL(S)	NEXT STEPS	EXAMPLE/NOTES
CXP can respond to the question related to the document or product.	**cxp**, **in-progress** Add the **product-question** label to classify the issue as a product question.	Provide the answer to the user's inquiry. If there's no further questions, close the issue	
CXP has identified the user's problem is related to an issue with the product.	**cxp**, **in-progress** Add the **product-issue** label to classify the issue as a product issue.	Work with engineering to check when they'll address the product issue or if there's a work-around available. Inform the user of the issue status and if they're satisfied with the reply, close the issue.	

ACTION	ASSIGN LABEL(S)	NEXT STEPS	EXAMPLE/NOTES
CXP can respond to the doc feedback and fix immediately.	**cxp** Add the **doc-bug** or **doc-enhancement** label to classify the doc feedback issue.	If the feedback is about something that is out of date, unclear, confusing, or broken in the article, assign the **doc-bug** label. Classify feedback about missing information that is blocking the customer's success as **doc-bug**. If the feedback suggests additions or improvements to the article, but there's no evidence the info in the article blocked the customer, assign the **doc-enhancement** label.	
CXP can fix by editing the document, but must do some work first	**in-progress**, **cxp** Add the **doc-bug** or **doc-enhancement** label to classify the doc feedback issue.	Finish the work required. When you've finished and merged the work, provide confirmation in the issue and close. Add the **cxp** labels.	Example: CXP files a pull request to address the issue. Close the issue after the PR is merged.
CXP needs help from product team to fix	**in-progress**, **cxp**, **awaiting-product-team-response**	After an initial response setting expectations with the customer, follow up with service teams similar to the process for the forums. After you've provided the answer, remove the **in-progress** label and the **awaiting-product-team-response** label, and add **cxp** label, then close.	
CXP can't fix and needs help from content author	**assigned-to-author**, and **doc-bug** OR **doc-enhancement**	In GitHub, assign the issue to the listed article author. If the feedback is about something that is out of date, unclear, confusing, or broken in the article, assign the **doc-bug** label. Classify feedback about missing information that is blocking the customer's success as **doc-bug**. If the feedback suggests additions or improvements to the article, but there's no evidence the info in the article blocked the customer, assign the **doc-enhancement** label.	

ACTION	ASSIGN LABEL(S)	NEXT STEPS	EXAMPLE/NOTES
CXP determines the issue violates Microsoft's code of conduct	**code-of-conduct**	Enter a comment indicating the issue is spam, close the issue, and then lock it to remove it from the article comments.	
CXP determines the issue is product feedback	**product-feedback**	Enter a comment for the customer indicating it's product feedback, provide a link to Azure User Voice or the appropriate feedback channel, and close the PR. **Note:** Product questions that CXP can answer will be answered immediately and not redirected to Azure User Voice	**Sample response:** "Thanks for taking the time to provide feedback on Visual Studio! This feature idea is interesting and we want to make sure it gets directly into the hands of the product team. Fortunately, we've got a dedicated channel for that very purpose! You can find it over here. We'd be happy to submit your feature request for you, if you like, but we want to give you the opportunity to submit it directly, if you'd like to receive notifications, vote, or monitor progress more closely. Let us know what you prefer. If we don't hear back from you in the next few days, we'll happily submit it on your behalf and link the created request here."
CXP reviews issue and determines that a support ticket is required to continue	**support-request**	Enter a comment for the customer to open a support ticket if they have a support plan using Azure support. If not, ask them to send an email to Azure CPX Community Connect with a link to the GitHub issue and their subscription ID. Work with the customer to file the support request, and close the issue after you get the SR number. If there's no response from the customer when you ask for the subscription ID within 48 hours, close the issue. Let the customer know they can reopen the issue at any time.	

ACTION	ASSIGN LABEL(S)	NEXT STEPS	EXAMPLE/NOTES
The issue doesn't contain enough information for CXP to triage it.	**needs-more-info**	Assign the issue to the contributor and ask for more info. What you ask for will depend on the issue contents. What service is this issue about? Is the question or problem with the documentation or with an Azure product itself? Once you have sufficient info, remove the needs-more-info label and triage per the other triage instructions.	
The issue is a request to collaborate on or contribute new documentation	**doc-idea**	Assign the issue to GitHub user **msmbaldwin** for secondary triage to content teams.	
The issue has positive feedback	**kudos**	Thank the contributor, then close the issue.	
The issue has feedback about localized content or translation quality	**loc**	Assign the issue to GitHub user **Sara-Nicolini** for triage to localization teams.	
The issue has feedback about the docs.microsoft.com experience or design	**docs-experience**	Assign the issue to GitHub user **Powerhelmsman** (Jian Dong) for secondary triage.	
CXP determines the question is best answered in a docs.microsoft.com Q&A forum or StackOverflow forum	**forum**	Enter a comment redirecting the customer to the appropriate forum for their question.	
CXP can provide a documentation or blog link that directly answers the question.	**doc-provided**	Enter a comment with a link to the existing documentation that answers the customer question.	
The customer enters a follow-up comment that says they've solved the problem, and it's clear there's no follow up necessary	**resolved-by-customer**	Thank the customer and close the issue where there's no needed fix in docs or the product.	

Miscategorized issues

If an author sends an email to Azure CXP Doc Feedback to report that an issue has been incorrectly categorized as a documentation-related issue, follow these steps:

1. Assign the **category-reassigned** label to the issues.
2. Remove the incorrect category label, and assign the correct new one.
3. Triage to the appropriate new assignee, and remove the author from the list of assignees.

Metadata

The triage team can assign issues only to the people listed in the metadata, in this priority:

1. The person listed in the **author** attribute.
2. The person whose Microsoft alias is in the **ms.author** attribute.
3. The person whose Microsoft alias is in the **ms.manager** attribute, if that person is a content publishing team manager.
4. The content publishing manager accountable for the service - see the Business approver list for Azure services.

To look up a user by their GitHub or MSFT alias, use the search feature in GitHub for Open Source - People.

Prevent assigning issues to people who are no longer employed at Microsoft

Each member of the Azure CXP triage team should install the Chrome extension. It lets you see in the GitHub UI whether a GitHub user is a Microsoft employee or not:

1. To install, go to Personal access tokens - Extension to install the extension.
2. Go to Personal access tokens - Create to generate a token.
3. Right-click the extension icon on the **Chrome** toolbar, and select **Options**.
4. Enter the token in the extension.

If the listed author of an article is no longer shown in the Chrome extension as an employee, assign the issue the content manager for the service. For names, see the Business approver list for Azure services.

Handling legal issues

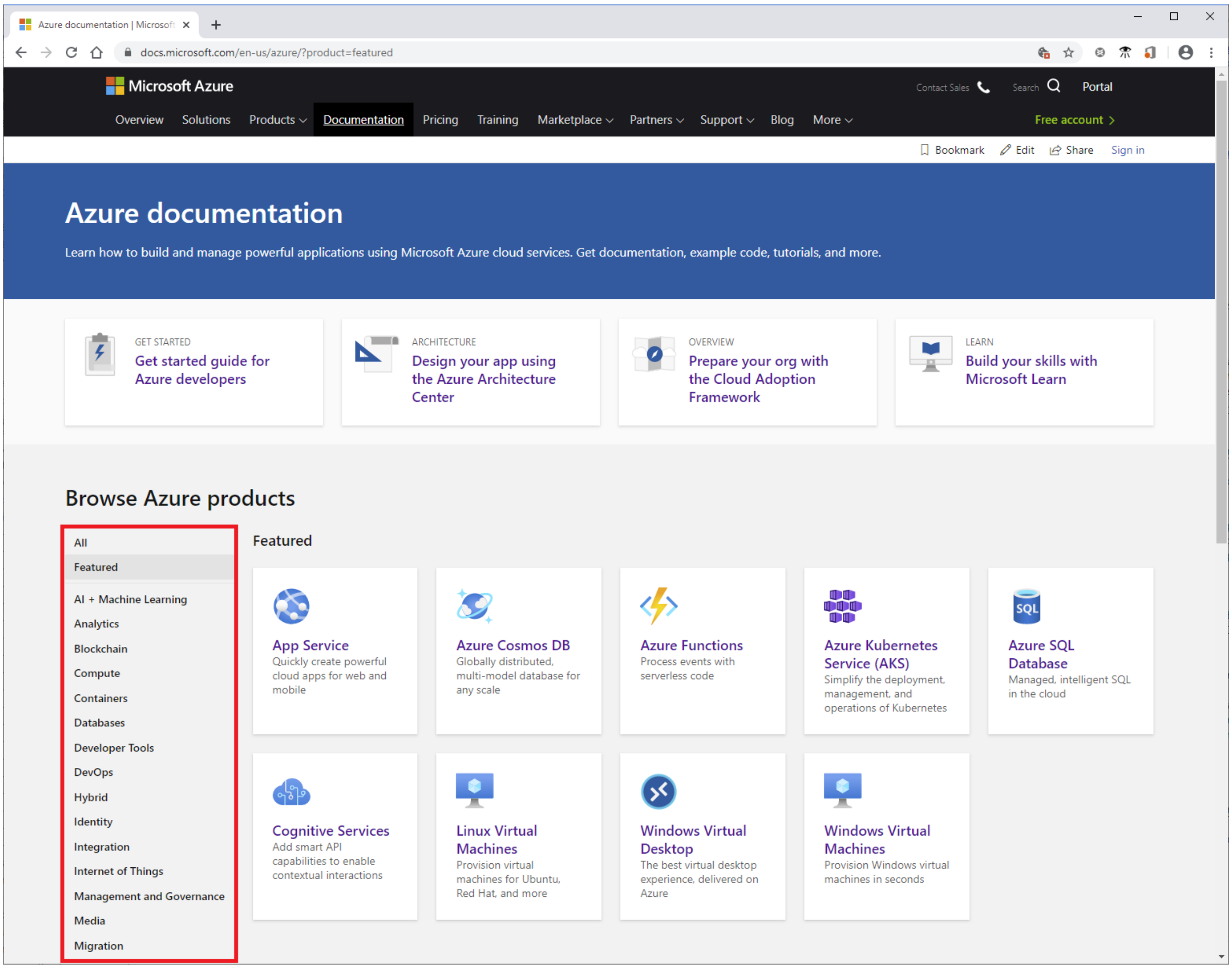
If an issue could have a potential legal implication for Microsoft, immediately escalate the issue to Azure CXP Doc Feedback. Legal issues include but aren't limited to issues like:

- Plagiarism: an issue mentions that our content is a copy of someone else's content or mentions the word "plagiarize".
- A report of any sort of misconduct, privacy issue, bias, or discrimination.

Update Azure docs hub page

6/24/2021 • 3 minutes to read

The Azure docs hub page is the entry point for all official Azure documentation. This article describes the steps to update the Azure docs hub page.



What appears on the Azure docs hub page?

The intent of the Azure docs hub page is to help customers understand the breadth and depth of documentation available for Azure. The Azure docs hub page also has a strict interlock with Azure.microsoft.com (ACOM) products to provide consistent messaging between these sites.

Here are the links that **can** appear on the Azure docs hub page:

- Product or service documentation that also appears on ACOM products
- Key programming language and tool documentation (maximum of 12)
- Sovereign cloud documentation

Here are the links that should **not** appear on the Azure docs hub page:

- Product or service documentation that is not listed on ACOM products
- Marketing content
- Links outside of the docs.microsoft.com domain (the one exception is https://code.visualstudio.com/docs)

Steps to update Azure docs hub page

1. Verify the service or product has been added to (or removed from) ACOM products.

2. If ACOM products do not reflect the needed changes, submit a request to update ACOM products.

3. Create a work item in mseng to request an update to Azure docs hub page and assign to Robert Lyon:

   https://mseng.visualstudio.com/TechnicalContent/_workitems/create/Task

   Alternatively, you can create a pull request to update index.yml in MicrosoftDocs/azure-docs-pr to match ACOM products.

   > **NOTE**
   >
   > ACOM products and Azure docs hub page must match. Before changes to the Azure docs hub page can be published, the changes must be published first on ACOM products.

The pull request reviewer will contact the designated business approver to request approval.

productDirectory cards

The Azure docs hub page is the index.yml file in MicrosoftDocs/azure-docs-pr. Each card in `productDirectory` has the following format:

- `summary` - Should match text on ACOM products.
- `imageSrc` - To keep file size small, should be a .svg file.
- `azureCategories` - One or more Azure categories. (Azure products can appear in more than one category)

```
# Card
- title: cardtitle
  summary: cardsummary
  imageSrc: svgimage
  azureCategories:
    - category1
    - category2
    - category3
  url: file OR https://docs.microsoft.com/file
```

Here is an example card:

```
# Card
- title: Azure Cosmos DB
  summary: Globally distributed, multi-model database for any scale
  imageSrc: https://static.docs.com/ui/media/product/azure/cosmos-db.svg
  azureCategories:
    - featured
    - blockchain
    - databases
    - iot
  url: cosmos-db/index.yml
```

Icons

If you need to add an icon to a card, make sure it is a .svg file. You can get icons from one of the following sources:

Docs shared Media (preferred)

- Preview page: https://static.docs.com/ui/media/index.html
- Repo: https://github.com/MicrosoftDocs/media/tree/master/src/product/azure
- Example imageSrc format: https://static.docs.com/ui/media/product/azure/file.svg

Docs.microsoft.com Shared Image Gallery

- Preview page: https://review.docs.microsoft.com/en-us/content-production-service/internal/image-gallery?branch=master#pivot=common&panel=logos
- https://github.com/MicrosoftDocs/DocsMedia/tree/master/media/logos
- Example imageSrc format: https://docs.microsoft.com/media/logos/file.svg

Local media folder

- Repo: https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/media/index
- Example imageSrc format: ./media/index/file.svg

Azure Icons Repository

- Preview page: https://cds-icons.azurewebsites.net/

Azure architecture icons

- Download page: https://docs.microsoft.com/azure/architecture/icons/

Azure category values

For `azureCategories`, you can use one of the following values. You cannot use any other values.

- ai-machine-learning
- analytics
- blockchain
- compute
- containers
- databases
- developer-tools
- devops
- featured
- hybrid
- identity
- integration
- iot
- management-and-governance
- media
- migration
- mixed-reality
- mobile
- networking
- security
- storage
- web
- windows-virtual-desktop

How is Azure docs kept in sync with ACOM products?

Approximately once a month, a script is run to detect differences between the Azure docs hub page and ACOM products. Updates are made to keep Azure docs hub page in sync with ACOM products.

Troubleshoot

When you update the Azure docs hub page, you might encounter errors. This table describes some issues and the corresponding solution.

EXAMPLE ERROR MESSAGE	SOLUTION
Unable to load file 'articles/test/index.yml' via processor 'Hub': Schema validation failed. Please validate the file and make sure it conforms to schema 'Hub' (https://static.docs.com/ui/latest/schemas/Hub.schema.json):	You are most likely trying to create a category hub page outside the predefined location (azure-docs-pr\articles\index.yml). This scenario is not supported.

Questions

If you have questions or problems related to the Azure docs hub page, send email to:

- docshub@microsoft.com

Related links

- Hub page schema
- Create or update a hub page

Azure subscriptions for Content and Learning organization (C+L)

5/10/2021 • 4 minutes to read

Employees IN C+L organization

The Azure internal registration and subscription (AIRS) setup information in this article is **only for people in Content & Learning (those who report up to Jeff Sandquist)** who don't want to or can't use the Visual Studio Subscription benefit.

To use your Visual Studio Subscription benefit for Azure account, go to https://my.visualstudio.com to activate your benefits.

Employees OUTSIDE C+L organization

If you can't use the Visual Studio Subscription benefit and you're in a group other than Content & Learning (C+L), contact your manager about setting up an Azure account through the internal registration system.

Set up Azure Subscription for a Content & Learning author

Internal Azure subscriptions are available for content developers in the Content & Learning organization. With an internal subscription, you don't need to use your Visual Studio subscription, your personal email, or your credit card in order to work with Azure features.

These subscriptions are charged back to Content & Learning. In other words they cost $$.

- DO NOT Use services which use a lot of processing power for a long period of time.
- DO NOT Leave a service running if you are **not actively** using it - Shut it down.
- DO NOT Run any production application under your test subscription. You can use your Visual Studio subscription for that.

Vendors: Have your Microsoft manager use the steps below to create a registration and subscription using their FTE alias. They should complete step 3 to add you to the subscription in the Azure portal. Due to the temporary nature of vendor assignments, this process is recommended by AIRS support to simplify tracking and management of accounts.

Step 1: Register for an Azure subscription (FTE)

Before you register for an Azure subscription, check with your manager to determine what codes you should use for cross-charging.

The following steps call out the fields that you need to update. You don't need to address all of the fields on the form.

1. Go to the Azure Internal Registration System (AIRS). You should have access, but if you don't, request access through the Azure Business Desk.
2. Click **New Registration**.
3. **Account Type**: Internal Billable Account.
4. **Account Owner ID**: Use your Microsoft account for your ID (FTE).
5. **Property/Service Name**: Azure Content Development
6. **PC Code for cross charging**: "P" + [your cost center number]

7. **PFAM for cross charging**: 000
8. **Program name**: C+AI Content & Learning Transform the Datacenter
9. **Paid support**: Developer
10. **Finance contact**: Mischelle Schimelpfenig <mischime> (Another backup contact in event of OOF: Alex Lejeune <allejeun>)
11. **Budget approver**: Alias of budget approver (your immediate manager can provide approval)
12. **Will a production application be hosted under this account**: No

Some Azure features require that you request access - you can usually do so on the preview page, or through the appropriate product team.

Step 2: Create your first subscription

1. After your initial request is approved, you receive an email message - click the activation link in the email to visit the AIRS site again.
2. Under **Purpose**, select the option for a personal subscription used for learning, support, projects, and sandbox work.
3. Under **Billing**, select **Provide a PC code that isn't showing in the list below**.
4. In the **PC Code** box, paste the profit center information you obtained from your business manager.
5. Click **Next**.
6. You'll be redirected to another internal consumption site, where you fill out another form with your personal details.

Step 3: Rename your subscription

AIRS names all new subscriptions with the same name. To make life easier for your business manager and budget team, rename your subscription so the team that owns it and the purpose of the subscription are obvious.

1. Sign in to the Account portal with your Microsoft account.
2. Click **Cost Management + Billing** in the left navigation pane.
3. Under **My subscriptions**, click the new subscription.
4. On the next page, click **Rename**.
5. In the **Subscription name** box, type a more descriptive name that makes it clear what team owns the subscription and what the subscription is used for.
6. Click **Save**.

Step 4: Onboard to 'Standard' Azure Security Center.

Ensure your subscription security setting is marked as 'Standard'; changing from default setting of 'Basic' in the Account portal. All our subscriptions must have this security setting in place to protect the C+AI fleet from unauthorized/anomalous activity. More information: (https://docs.microsoft.com/azure/security-center/security-center-get-started/)

Optional step 5: Add vendor to subscription

As an FTE, use this procedure when you've created a registration and subscription under your alias but it's specifically for vendor use.

1. Sign in to the Account portal with your Microsoft account.
2. Click **Cost Management + Billing** in the left navigation pane.
3. Select **Subscription**.
4. Under **My subscriptions**, choose the subscription you've created specifically for the vendor to use.
5. Select **Access control (IAM)** > **Add**.
6. For **Role**, select **Contributor**.
7. In **Assign access to**, select **Azure AD user, group, or application**.

8. In **Select**, type the vendor's name.
9. Click **Save**.

For more information about AIRS, see AIRS and Sponsorship FAQ.

Sovereign clouds

7/7/2021 • 2 minutes to read

This article summarizes sovereign clouds and the documentation process for them.

What sovereign clouds are

Sovereign clouds are physically isolated instances of Azure dedicated to the needs of a country, region, or government. They allow these customers to move sensitive services and apps to the cloud by addressing specific regulatory requirements. They're an important compete effort with major cloud providers with an estimated revenue of $29 billion by 2022.

Current sovereign clouds

Currently there are three sovereign clouds:

- Azure Government
- Azure Germany
- Azure China 21Vianet (***21Vianet** is the company that independently operates Azure in China*.)

Technical documentation

- **What**: *Only technical differences between the sovereigns and global Azure (Microsoft Azure) are documented in the sovereign docs*. Sovereign docs link to global Azure for info common to both. We don't add sovereign info to global Azure docs. Examples: Azure Government security and Azure Government identity.
- **Where**: There's a hub page on Docs for sovereign clouds. There are individual landing pages for Azure Government, Azure Germany, and Azure China 21Vianet.

Who does what

- **Global Ecosystems**: Owns sovereign clouds. They're part of Azure engineering. They onboard engineering service teams that have sovereign cloud offerings to relevant processes. The Global Ecosystems team creates some documentation, and they onboard Azure service PMs to do the rest.
- **Azure engineering**: After onboarding with Global Ecosystems, service PMs write or work with their content teams to write the documentation for their sovereign cloud offerings.
- **Marketing**: Owns the marketing pages for sovereigns on azure.microsoft.com. The pages include info on compliance, service availability, and pricing.
- **Content & Learning**: Content & Learning doesn't write sovereign cloud technical documentation but contributes in this way:
  - **Hosting**: Hosts sovereign technical content on Docs.
  - **Updates**: Updates engineering and Global Ecosystems on significant changes to the Docs platform and document conventions (ex. new content model).
  - **Support**: Provides ongoing support through the Docs contributor guide and the Azure Technical Content Chat alias (azcontentchat@microsoft.com), a forum for Q+A about the authoring process and tools.
  - **Contacts**: Monica Rush (monicar) and Femila Anilkumar (femila)

See also

The Azure Government marketing site on azure.microsoft.com.

> **NOTE**
>
> Azure Germany and Azure China 21Vianet don't currently have marketing pages on azure.microsoft.com.

New Azure service checklist

5/14/2021 • 3 minutes to read

If you're a technical writer and you need to create content for a new Azure service, use the following checklist to make sure you do all the required tasks:

☑	TASKS
☐	Identify the lead PM and sponsor for the new service.
☐	Keep your manager informed about the new service content progress.
☐	Identify the expected timeline for the release.
☐	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 and the azure.microsoft.com URL that identifies your service, and the slug is used for the folder name, and metadata value `ms.service`. For example, `virtual-machines` is the service slug for Azure Virtual Machines and is consistently used between the docs site at https://docs.microsoft.com/azure/virtual-machines/ and the ACOM site https://azure.microsoft.com/services/virtual-machines/.
☐	Create a new release branch for your new content.
☐	Create a new folder for your service's docs using the new service slug value. Create the folder locally, and add your first files. Then the folders are "created" in GitHub.com when the first file is added and pushed into the online repository. For example, this folder matching the service slug is used for the Azure Virtual Machines docs: `https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/virtual-machines` .
☐	Create the new service metadata value. Use the new ms.service value request form to add your service metadata to the list. Make sure to select **Yes** to include your service docs metrics in the content engagement reports. The `ms.service` metadata property typically uses the same value as the service slug.
☐	Work with your product team to determine content plan and create new articles.
☐	Get the new service .svg icon from the marketing representative for your service.
☐	If available, copy a related service toc.yml file into your service folder and edit for your new service.

☑	TASKS
☐	If available, copy a related service index.yml (landing page) into your service folder and edit for new service.
☐	Determine if the service is a public preview or general availability. For a preview service, refer to the Identify public preview content article, and make the necessary changes in the TOC and landing page.
☐	Based on your content needs, list the articles that you need to create. Review the recommended article types for Docs.
☐	Based on your article list, define the TOC structure 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 TOC checklist spreadsheet to draft your TOC.
☐	Ensure your articles align with the docs.microsoft.com writing and technical principles checklist.
☐	Understand and incorporate SEO techniques in your content.
☐	See the guidance to create or update a landing page for creating a new index.yml landing page file.For example, see the [Azure Virtual Network landing page at https://review.docs.microsoft.com/en-us/azure/virtual-network/?branch=main.
☐	Add your service to the master breadcrumb file: azure-docs-pr\bread\toc.yml.
☐	If you have a product forum URL, add it to the docset docfx.json file: `MicrosoftDocs/azure-docs-pr/blob/master/docfx.json`. Under `"feedback_product_url"` add your folder path and feedback URL (see other line items as an example).
☐	Submit a request to add the new service on your area hub page (if it applies) and include the new service icon to use.
☐	Divy Sharma (DIVYARATNA) is responsible to include your new service on the "all services" page. Contact Divy to verify everything is ready to go.
☐	When your content is ready, understand the pull-request (PR) submission recommendations and create a PR to receive feedback from SMEs.
☐	Check the build validation, and address any issues.
☐	Check the Acrolinx results, if available. See Acrolinx coverage.
☐	Invite reviewers to provide feedback on the new content.

☑	TASKS
☐	After you incorporate the reviews and feedback, sign off (#sign-off) 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.
☐	Schedule your [release branch merge](contribute-content-release-planning. md#release-planning-process-for-self-managed-releases-with-a-release-branch-azure-docs-pr).
☐	Generate an fwlink to your landing page so your product team can link to the content from the portal.
☐	Consider whether you'll need reference content for any APIs, libraries, packages, CLIs. Onboarding times can be long, so take timelines into consideration.
☐	Whether your service is going into public preview or GA stage, remember to request a Product Launch Readiness(PLR) sign-off for docs. PLR is a new Azure-wide process where the product and as the docs need to be signed off. Specifically, you could look at the Azure Doc PLR process for content-specific guidelines. Private previews don't require a PLR sign-off.

What is the Content Standards report?

11/2/2020 • 5 minutes to read

The goal of the Content Standards report is to advise Azure content owners and content managers about how well their content meets content standards for their overviews, quickstarts, and tutorials, as well as their tables of content (TOCs). The report also provides specific information for which criteria are not being met in an article and links to guidance for those criteria.

Individual writers can assess content they own. Managers can aggregate based on service or team members or both.

The criteria chosen for this dashboard are those that can be evaluated by logic and in general speak to metadata, scannability, and consistent article structure. The audit does not assess the quality of the content.

Why these criteria are important:

- **Metadata**: content needs to have the right metadata for reporting purposes. This is especially important for the content performance dashboard since the calculations for article health are tied to the ms.topic value.
- **Scannability**: one of the five writing principles, it supports the quickness with which a customer can determine if the article addresses their question.
- **Consistent article structure**: ensuring that our TOC is standard and that our content shows up in the sections it belongs in helps customers find our content when browsing the TOCs.

Report scope

- Repo: azure-docs-pr
- Audit criteria

How it works

All criteria are from existing standards that are documented in the Contributor Guide.

Backend programming evaluates each article against the criteria. Each article receives a pass/fail grade for each criterion. If the article meets 80% of the criteria or more, it is considered compliant. If the article meets less than 80% of the criteria, it is considered not compliant. There is no weighting of criteria.

The report is refreshed daily.

How to use it

The dashboard has two tabs: **Charts** and **Articles**.

The **Charts** tab provides data visualizations for groups of content—a portfolio view. You can filter by article type, ms.service, author, manager. You can exclude exempt content from the charts. This report is useful for seeing and sharing aggregated views on a set of topics.

Learn more about the Charts tab.

The **Articles** tab provides article-specific audit results. It is useful for identifying articles that need to be updated to meet the 80% pass rate. For each article, it tells you which rules the article passes, and which rules the article fails. You can request exemptions for some of the criteria from this page.

Learn more about the Articles tab.

Audit criteria

RULE ID	RULE NAME	CRITERIA
Overview rules		
Ov01	ms.topic value	ms.topic must be "overview"
Ov02	H1 format	H1 format must be "What is *service*?" or "What are *services*?"
Ov03	Last H2	Last H2 must be "Next steps"
Ov04	Next steps link	Overview "Next steps" section must have 1-3 links
Ov05	Next steps link format	Overview "Next steps" link format must not use blue box link
Quickstart rules		
QS01	ms.topic value	ms.topic must be "quickstart"
QS02	meta title keyword	Meta title must include the word "quickstart"
QS03	meta description	Meta description must include the word "quickstart"
QS04	H1 keyword	Quickstart H1 must start with "Quickstart:"
QS05	First H2	First H2 must be "Prerequisites"
QS06	H2 numbering	H2s must not be numbered as steps in a procedure
QS07	CUR H2	"Clean up resources" section must come before "Next steps" section
QS08	Last H2	Last H2 must be "Next steps"
QS09	H3 numbering	H3s must not be numbered as steps in a procedure
QS10	Green checkmarks	Quickstart must not contain green checkmarks
QS11	Next steps link	Quickstart "Next steps" section may only have 1 link
QS12	Next steps link format	Quickstart "Next step" link format must be blue box link

RULE ID	RULE NAME	CRITERIA
Tutorial rules		
T01	ms.topic value	ms.topic must be "tutorial"
T02	meta title keyword	Meta title must include the word "tutorial"
T03	meta title numbering	Series number not allowed in meta title
T04	meta description	Meta description must include the word "tutorial"
T05	H1 keyword	Tutorial H1 must start with "Tutorial:"
T06	H1 numbering	Series number not allowed in Tutorial H1
T07	Green checkmarks	Intro: Tutorial must use green check marks before first H2
T08	First H2	Tutorial first H2 must be "Prerequisites"
T09	H2 numbering	H2s must not be numbered as steps in a procedure
T10	CUR H2	"Clean up resources" section must come before "Next steps" section
T11	Last H2	Last H2 must be "Next steps"
T12	H3 numbering	H3s must not be numbered as steps in a procedure
T13	Next steps link	Tutorial "Next steps" section may have only 1 link
T14	Next steps link format	Tutorial "Next steps" link format must be blue box link
TOC rules		
TOC01	Top link label	Top link in TOC must be in format "*Service* documentation"
TOC02	Top link destination	"*Service* documentation" must link to *service* index.yml
TOC03	Overview node number of topics	TOC has no more than 2 topics in Overview node, and only 1 can be ms.topic: overview
TOC04	Overview node topic type	TOC must have an Overview node and it must contain an Overview topic

RULE ID	RULE NAME	CRITERIA
TOC05	Tutorials node expanded	TOC Tutorials node must be expanded by default if no Quickstarts node
TOC06	Tutorials node required	TOC must have required Tutorials node with at least one Tutorial]
TOC07	Quickstarts node	TOC Quickstarts node must be expanded by default if there is one
TOC08	Node order + labels	TOC nodes must be in correct order. Labels must be spelled correctly.
TOC09	Node type	TOC may only contain allowed nodes
TOC11	Overview topic location	Overview topic must be in Overview node of the TOC
TOC12	Tutorial topic location	Tutorial topic must be in the Tutorials node of the TOC
TOC13	Quickstart topic location	Quickstart topic must be in the Quickstarts node of the TOC
TOC14	Quickstart in H1	Do not add “Quickstart” to the H1 of any article not in the Quickstarts node of the TOC
TOC15	Green checkmarks	Article uses green checkmarks but is not a tutorial
TOC16	Tutorial in H1	Do not add "Tutorial" to the H1 of any article not in the Tutorials node of the TOC

Next steps

- Learn more about the Charts tab.
- Learn more about the Articles tab.

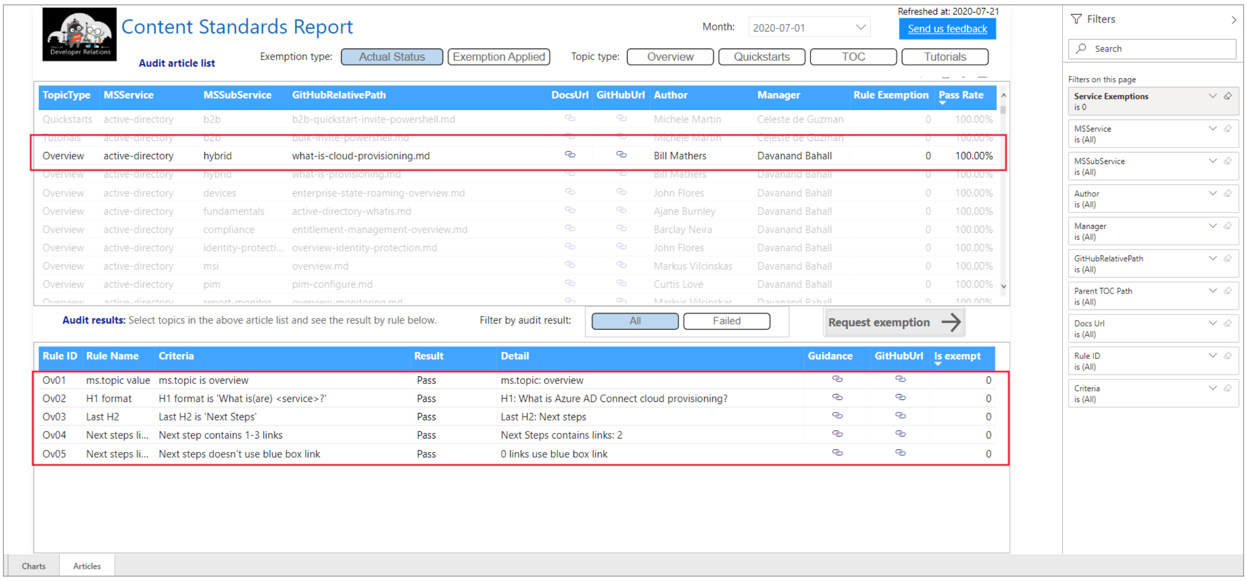
How to use the Articles tab of the Content Standards report

11/2/2020 • 7 minutes to read

The **Articles** tab of the Content Standards report has two output tables and multiple filters across each table and in the **Filter** pane on the right.

The top table is the **Audit article list** and lists all the articles that are the result of the applied filters. The bottom table, **Audit results**, lists the audit results of the articles in the top table.

If you select a single article in the **Audit article list**, it will filter the results in the **Audit results** to the rules that apply only to the selected article.



Filters set on the **Charts** page do not apply to this page.

What's in the Audit article list?

The Audit article list contains the following columns:

COLUMN	INFORMATION DISPLAYED
Topic type	Overview, Quickstarts, Tutorials, TOC
MSService	The value in the ms.service field
MSSubService	Derived from the folder hierarchy of azure-docs-pr repo
File name (GitHubRelativePath)	File name of the article
Links to the article in GitHub and on docs.microsoft.com	
Author	The value in the ms.author field
Manager	Derived from HeadTrax

COLUMN	INFORMATION DISPLAYED
Rule exemption	0 if no rule-level exemption applied to article
Service exemption	0 if no service-level-exemption applied
Pass rate	80% or higher is needed to be considered compliant

Click a column heading to sort by that column. You can only sort on one column at a time.

If the list is long, you can apply additional filters from the Filters pane or toggle the Exemption type and Topic type filter buttons at the top of the list.

To reset filters

There are several ways to reset the filters:

- The filter buttons toggle between on (blue) and off (white).
- To reset a filter in the **Filters** pane, select the **erase** icon on the filter.
- Select **Reset to default** at the top of the report to remove all applied filters.

What's in the Audit results list?

The Audit results list gives you the specific rules applied to the content in the Audit article list, including whether the content passed or failed that rule.

This list displays:

COLUMN	INFORMATION DISPLAYED
Rule ID	Unique ID for each rule
Rule name	Short name for the rule. **Rule ID** and **Rule name** are mainly used by the report and the exemption workflow.
Criteria	The criteria the content needs to meet for the rule.
Result	**Pass**: the article meets the criteria for the rule. **Fail**: the article doesn't meet the criteria for the rule.
Detail	What the audit report found in the article when testing for that rule.
Guidance	A link to the section in the Contributor Guide article that explains the rule.
GitHubUrl	A link to the article in GitHub.
Is exempt	0 if no exemption to this rule is applied in the article.

To focus on the criteria you need to address, you can filter out the passing rules by clicking the **Failed** button at the top of the list.

Note that sometimes, the only thing you need to fix is a spelling or capitalization error. For example, there's a lot of variety in how the node for how-to content is labeled. The correct label is "How-to guides".

Find articles by author

To find articles belonging to a specific writer or multiple authors:

1. In the **Filters** pane, open the **Author** filter, and click to select the author's name for each author whose articles you want to see.
2. The results of the filter will be displayed in the **Audit article list**. The results of the audit will appear in the **Audit results** list.
3. To see only audit results for a single article, select that article in the **Audit article list**. Only the rules that apply to that topic type and the pass/fail result will be displayed in the **Audit results** list.

Find articles by manager

To find articles belonging to writers reporting to a specific manager:

1. In the **Filters** pane, open the **Manager** filter, and click to select the manager's name.
2. The results of the filter will be displayed in the **Audit article list**. The results of the audit will appear in the **Audit results** list.
3. To see only audit results for a single article, select that article in the **Audit article list**. Only the rules that apply to that topic type and the pass/fail result will be displayed in the **Audit results** list.

Find TOCs

Because there is no ms.author for a TOC.yml file, TOC ownership is assigned to the manager of the writing team that writes the content for the service. To find the TOCs a manager owns, filter on the manager's name in the **Author** filter. Service and subservice values for TOCs are derived from the folder path for the TOC file.

Find articles by service or sub-service

You can filter on both service and subservice. When you select a service in the **MSService** filter, the **MSSubService** filter list is scoped to only the subservices of the selected filter.

To find articles belonging to a single service or to see articles for multiple services:

1. In the **Filters** pane, open the **MSService** filter, and click to select the service or services whose articles you want to see.
2. The results of the filter will be displayed in the **Audit article list**. The results of the audit will appear in the **Audit results** list.
3. To see only audit results for a single article, select that article in the **Audit article list**. Only the rules that apply to that topic type and the pass/fail result will be displayed in the **Audit results** list.

Request an exemption

You can request that an article or TOC be exempt from some of the rules. The allowed exemptions are:

EXEMPTION TYPES	CRITERIA IDS	FOLDER-LEVEL EXEMPTION	TOPIC-LEVEL EXEMPTION	TOC-LEVEL EXEMPTION	NOTES

EXEMPTION TYPES	CRITERIA IDS	FOLDER-LEVEL EXEMPTION	TOPIC-LEVEL EXEMPTION	TOC-LEVEL EXEMPTION	NOTES
Retiring service	n/a	Yes	No	No	An example is a service that is being replaced by or merged with a newer service, and writing and developer resources are no longer available for the content.
Legacy content	n/a	Yes	No	No	An example is content derived from a purchased service and whose content is published as-is until it can be replaced.
Content in Azure repo that doesn't follow MVC by design	n/a	Yes	No	No	An example is a mature service whose content comes to our org because of a larger re-org or migration.
TOC contains non-standard nodes	TOC09	Yes	No	Yes	Most frequent: stacked TOC, but also can be used for adding a single node, such as Troubleshoot.
TOC contains required Tutorials section	TOC06	No	No	Yes	
TOC nodes are in a non-standard order	TOC08	No	No	Yes	
Overview topic in a node other than Overview	TOC11	No	Yes	No	
Tutorial topic in a node other than Tutorials	TOC12	No	Yes	No	
“Tutorial” is in H1 of an article not in the Tutorials node	TOC16	No	Yes	No	

EXEMPTION TYPES	CRITERIA IDS	FOLDER-LEVEL EXEMPTION	TOPIC-LEVEL EXEMPTION	TOC-LEVEL EXEMPTION	NOTES
Quickstart topic in a node other than Quickstarts	TOC13	No	Yes	No	
“Quickstart” is in H1 of an article not in the Quickstarts node	TOC14	No	Yes	No	

You can only file exemptions against TOC files although they may actually affect another topic type. For example, TOC14 allows you to use the “Quickstart” H1 keyword in an article that is not linked to from the Quickstarts node.

To file an exemption in the report

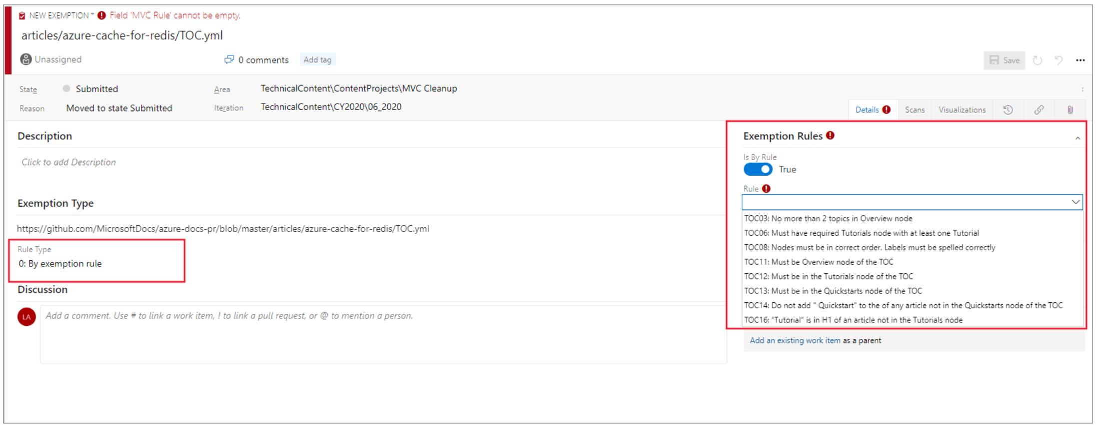
1. In the **Audit article list**, select the TOC you want to file an exemption for. The **Request exemption** button becomes active and turns blue. Note, the **Request exemption** button doesn’t turn blue, you cannot file an exemption against that article.

2. Click the **Request exemption** button. An Azure DevOps exemption request work item opens. The **Title** field is populated from the report and the article’s GitHub file path is pulled into the **Exemption Type** area.

3. In **Exemption Type** area, under **Rule Type**, select the exemption type you want. The options are:

   - **0. By exemption rule**

     - When you want a rule-level exemption, select this option, and then select the specific rule in the **Exemption Rules** area. Click under **Rule** to see the list of rules available. You can only choose one.

   - **1. Retiring service**

   - **2. Legacy content**

   - **3. Service is exempt from standards**

   - **4. Request non-standard nodes**

4. In the **Business Justification** area, provide the reason for your exemption request.

5. Save the new exemption request.

The request goes into the review backlog. The reviewers will review the request and make a decision. If the request is approved, the **State** will be set to **Approved** and the exemption will be integrated into the audit report. If the request is denied, the **State** will be set to **Denied**, and you will need to revise your content to meet the criteria.

Note: You can also file an exemption request outside of the tool. To do so, open Azure DevOps in TechnicalContent, select **Work items** > **New Work item** and, then select the **Exemption** work item type. If you log a request this way, you will need to provide the full github path to the TOC file in The **Exemption Type** area:

```
https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/<folder>/TOC.yml
```

Exemption request queries

- Request backlog
- Requests in review
- Approved requests
- Denied requests

Report feedback

To report a problem or provide feedback on the report, use the **Send us feedback** link at the top of the report.

Next steps

- Content Standards report overview
- Learn about the Charts tab

How to use the Charts tab of the Content Standards report

11/2/2020 • 2 minutes to read

The four charts on the **Charts** page provide an aggregate view of your portfolio.

For all charts on this page, if no filters are applied, the data applies to the entire Azure portfolio being measured. The available filters on this page are MSService, Author, and Manager, as well as the topic type filter buttons. You can also take approved exemptions into consideration by selecting the **Exemption Applied** filter button.

Filters set on the **Articles** tab do not apply to this page.

Compliance rate trend

Use this chart if you want to see a month-over-month trend chart for compliance. Data begins with FY2021.

Compliance rate by topic type

The bar chart indicates the percent compliance by topic type. This chart is also a filter for the other charts. Click a column to filter for that topic type. Ctrl+click to select more than one column.

Compliance count by typic type

The stacked bar chart indicates the number of compliant and not compliant topics by type. This chart is also a filter for the other charts. Click a column to filter for that topic type. Ctrl+click to select more than one column.

Number of passing and failing articles by rule type

The stacked bar chart indicates the number of compliant and non-compliant topics by rule type. To learn more about the specific rule, hover over the column to see the details fly-out.

Next steps

- Content Standards report overview
- Learn about the Articles tab

Write security documentation for an Azure service

6/24/2021 • 2 minutes to read

This article is for writers who are writing security documentation for their Azure service.

An Azure service that has articles about security must have the following table of contents (TOC) node:

- Concepts > Security

The service documentation can have the following security articles under the Security node:

- A security features article that provides an overview or introduction to security for the service.
- Conceptual articles to provide in-depth explanation of functionality related to the service.
- A security baseline article that contains recommendations to improve the security posture of a service.

Service documentation may also have security how-to articles. How-to articles show the customer how to complete a security task and go under the TOC node **How-to guides**.

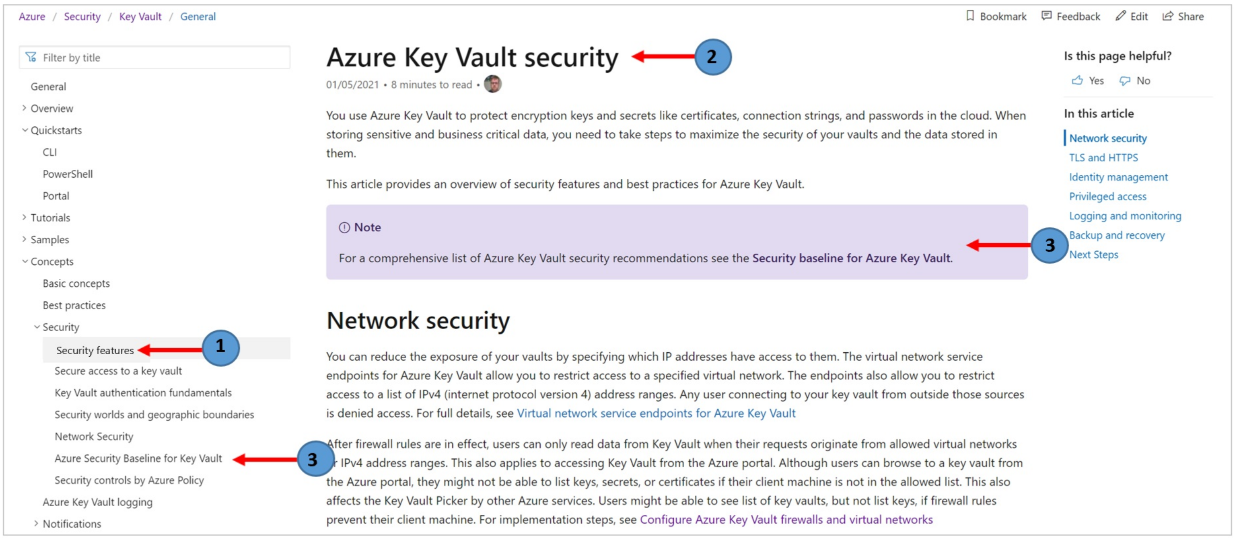
Security features

The security features article is an overview-type article that introduces the service's security capabilities. The service writer determines the capabilities to address in this article.

Security features checklist

Here's a checklist to make sure your security features article meets the key formatting guidelines.

NUMBER	REQUIREMENT
1	Use the TOC entry **Security features** as the first node under **Concepts > Security**.
2	H1 is "*Service name* security".
3	If the service has a security baseline article, include the following note in the section under the H1: "For a comprehensive list of *Azure service* security recommendations see the *service's security baseline article*." Link to the baseline article.
4	The .md file name for the security features article is security-features.md.
5	Metadata attribute ms.topic is set to conceptual.



Use this template to develop the security features article for an Azure service.

Conceptual article

Service documentation may have security conceptual articles. A conceptual article should focus on service-specific information.

Azure security fundamentals documentation provides high-level conceptual information on most security capabilities. All services that support a capability should link to security fundamentals docs for this high-level information. Linking to security fundamentals documentation for conceptual information enables the writer for a service to focus on service-specific information in their documentation.

If security fundamentals documentation doesn't document a capability that you want to link to, contact terrylan about adding the capability to fundamentals.

Conceptual article checklist

Here's a checklist to make sure your conceptual article meets the key formatting guidelines.

- Under the TOC node **Concepts > Security**.
- Address service-specific information.
- If security fundamentals documentation covers a security capability that you need to cover in your documentation, link to security fundamentals instead of creating your own.

How-to articles

Service documentation may have security how-to articles. Security how-to articles go under the TOC node **How-to guides**. Where relevant provide guidance for portal, PowerShell, CLI, and Azure Resource Manager (ARM) templates.

Next steps

- Template for writing a security features article
- Guidance for writing a security baseline article
- The Azure security fundamentals site on docs

Base template for introducing security features supported by an Azure service

3/5/2021 • 2 minutes to read

This article provides a template in markdown code that you should use when writing the security features article. The security features article introduces key security capabilities supported by your Azure service.

All articles need the metadata header along with the required values for reporting.

You can copy and paste the following markdown for your security features article:

```
---
title: Template for writing the security features article #Required; page title displayed in search results.
Include the words "security" and the Azure service name. For example, Azure Backup security. Include the
brand.
description: Template for writing the security features article #Required; article description that is
displayed in search results. Include the words "security" and the Azure service name.
author: TerryLanfear #Required; your GitHub user alias, with correct capitalization.
ms.author: terrylan #Required; Microsoft alias of author; optional team alias.
ms.service: security #Required; service per approved list. service slug assigned to your service by ACOM.
ms.topic: conceptual #Required
ms.date: 02/16/2021 #Required; mm/dd/yyyy format.
---

<!---Recommended: Remove all the comments in this template before you sign-off or merge to master.--->

<!---The security features article contains introductory information about key security capabilities
supported by your service. This article is located under the "Concepts" node and "Security" sub-node of your
service's ToC.

The security features article does not have to address every security capability supported by the service,
only key capabilities.

A goal of the security features article is to help the customer navigate to how-to information. This article
can link directly to how-to articles. If a service, though, has a conceptual article that provides more
detail about a security capability, the security features article can link to the conceptual article

--->

# <Name of Azure service> security
<!---Required:
Starts Azure service name followed by "security."
Example H1: Azure Backup security
--->

Introductory paragraph
<!---Required:
The introductory paragraph helps customers quickly determine whether an article is relevant.
Describe in customer-friendly terms what the service is and does, and why the customer should care. Keep it
short for the intro. This article is about security capabilities so focus on how the service helps to secure
data and resources.
--->

An example of an introductory paragraph in the security features article:

One of the most important steps you can take to protect your data is to have a reliable backup
infrastructure. Azure Backup provides security to your backup environment - both when your data is in
transit and at rest. This article lists security capabilities in Azure Backup that help you protect your
data and meet the security needs of your business.

<!---Required:
```

```
<!--- Required:
If your Azure service has a security baseline article then link to the baseline article via a note.
Use the following text "For a comprehensive list of <Azure service name> security recommendations see the
<title of service's baseline article>".
--->

An example of a note in the security feature article:

> [!Note]
> For a comprehensive list of Azure Key Vault security recommendations see the [Azure Security Baseline for
Key Vault](https://docs.microsoft.com/azure/key-vault/general/security-baseline).

## <H2 title>
The service writer determines the format of and the information in the article's H2 sections.

## Next steps

<!--- Required:
Always have a Next steps H2.
Use regular links; do not use a blue box link. Insert links to other articles that are logical next steps or
help users use the service.
Do not use a "More info section" or a "Resources section" or a "See also section".
--->
The service writer determines the conceptual or how-to articles linked to under Next steps.
```

What is the Azure Architecture Center?

7/6/2021 • 5 minutes to read

The Azure Architecture Center (AAC) helps customers design, build, and operate solutions on Azure.

Our mission statement: **Publish technical content and cloud best practices to provide the right information to the right customers at the right time. Architectural guidance is proven, authoritative, actionable, and relevant.**

The Architecture Center provides real-world guidance to solve real-world problems. It helps customers onboard to Azure and ensure long-term success. To achieve these goals, guidance on the Architecture Center follows these principles:

- **Proven:** The guidance published on the Architecture Center is based on proven practices.
- **Practical:** The Architecture Center helps customer build real solutions.
- **Comprehensive:** The Architecture Center provides guidance on all aspects of building for the cloud:
  - Operational Excellence
  - Security
  - Cost Optimization
  - Reliability
  - Performance Efficiency

The Architecture Center is governed by the patterns & practices (PnP) team within Developer Relations. This platform leverages the Microsoft Docs contribution guide and training for onboarding new users to collaborate with our platform.

This article provides the structure of the Architecture Center platform. The article AAC field contributions describes how new contributors can onboard and start publishing on this platform.

Audience

Our customers are architects, developers, and administrators who are responsible for designing, creating, and operating solutions on Azure to deliver business value.

GitHub repositories

The Azure Architecture Center content is accessible via:

- a public-facing GitHub repo, for general public wanting to make minor changes using the online editing option, and
- a private-facing GitHub repo, which allows proper staging and larger content changes. This is expected to be used for contributions from Microsoft employees who have a GitHub account.

Content types

The Architecture Center publishes some well-defined content types, distinguished by **ms.topic** and **ms.subservice** metadata values.

> **IMPORTANT**
>
> Article metadata are extremely important for content traffic and performance analysis, discoverability, and specific site experiences. Most articles in the Architecture Center are *conceptual*, and hence will have corresponding value in the *ms.topic* metadata. These conceptual articles are distinguished from each other by specific *ms.subservice* values. Make sure to use the right metadata as mentioned below.

Architectural content

This includes article types that provide architectural solutions for real-world customer scenarios, at varying degrees of technical depth. The **ms.topic** value for these articles must be **conceptual**.

- Technical guides for Azure categories
  - These are design guides that discuss a solution using Azure services in depth. These reside under the ***Azure categories*** TOC node, and then under the ***Guides*** sub-node. They are sometimes organized into a series of articles. They were historically developed to cover any learnings that were not covered by a corresponding reference architecture, and so were based on tried and tested source code. These do not need to follow a set pattern. However, they should cover considerable depth of the topic. For example, see the Azure Data Guide.
  - The **ms.subservice** value for these types must be **azure-guide**.
  - Include a **## Next steps** section at the end, featuring a bulleted link list to other AAC articles and any other related articles on Docs (or other sources, as appropriate).
- Reference Architecture
  - These are the gold-standard architectures, backed by tried and tested engineering assets. These articles should discuss how the architecture affects the five pillars of a Well-Architected solution.
  - The **ms.subservice** value for these types must be **reference-architecture**.
- Example Scenario
  - These are typically architectures or solutions implemented by one or more customers, and hence typically coming from *field contributions*. These may or may not have engineering assets. They should attempt to discuss all five pillars of a Well-Architected solution.
  - The **ms.subservice** value for these types must be **example-scenario**.
- Solution Idea
  - These are an idea of a solution, which may or may not have been already implemented.
  - The **ms.subservice** value for these types must be **solution-idea**.

Cloud fundamentals

These include industry standard guides, focusing on fundamental patterns and practices of cloud architectures. The **ms.topic** value for these articles must be **conceptual**.

- Technology-agnostic guides
  - This is guidance that is fundamental to cloud development, and not limited to any Azure-specific architecture. This category includes articles such as *design principles*, *technology choices*, and so on.
  - The **ms.subservice** value for these types must be **guide**.
- Design Pattern
  - These are common engineering patterns used to solve a specific problem.
  - The **ms.subservice** value for these types must be **design-pattern**.
- Anti-patterns
  - These are common engineering patterns that may not be able to scale under pressure.
  - The **ms.subservice** value for these types must be **anti-pattern**.
- Best practices
  - These are some carefully-constructed best practices recommended for cloud development.

- The **ms.subservice** value for these types must be **best-practice**.

Other topics

- Well-Architected Framework is a framework that guides architectural excellence, in the form of five pillars: Cost Optimization, Operational Excellence, Performance Efficiency, Reliability, and Security. The **ms.topic** value for these articles must be **conceptual**, and the **ms.subservice** value for these types must be **well-architected**.
- Topic Landing Page - these are a landing page that provide an opening to a series of articles, combining a navigational index with interesting visual elements. The **ms.topic** value for these articles must be **landing-page**.

Many of the content types have associated code assets, such as code samples, reference implementations, and deployment scripts. When published, they can reside in our repos or yours, as long as they are not personal repos (org repos only).

Retirement strategy

The nature of the Azure environment means that content will frequently become inaccurate, no longer apply, or need updating. Documents that are no longer valid and cannot be made valid via editing will be archived and removed regularly. The AAC follows the Docs Retirement process with minor adjustments:

- Every article is reviewed by an SME every six months. Whenever a review occurs, the ms.date metadata is updated. This metadata is the flag used for automated stale content processing.
- If an SME cannot be located or unable to perform the edits, or the content is no longer accurate/stale, it will be flagged for removal. A notice will be placed on the page and it will be removed from the TOC (if applicable)
- After an article has been flagged for 30 days, it will be archived via the process outlined by the Docs Retirement process.

Next steps

Interested in contributing to Azure Architecture Center? Read AAC Field Contributions to learn the process.

Azure Architecture Center contributions

6/28/2021 • 9 minutes to read

The Azure Architecture Center encourages contributions from both the public and Microsoft FTEs, either via GitHub issues and pulls requests (PRs), or within the context of programmatic alignment. Some general guidelines for contributing programmatically to the Architecture Center are detailed here.

Base requirements

All external contributions *must have* the following attributes:

- Describes a specific, actionable, and repeatable pattern being described where appropriate.
- Provides templates, scripts, and commands to achieve the scenario being described (not just a story).
- Covers a gap in existing articles and describes a "golden path" (not just "a path").
- Crosses individual SKU boundaries. If your proposal is specific to an individual service, perhaps contributing to the documentation for that service would be more suitable.
- Is a supported scenario (no hacks).
- Discusses the key pillars of Architecture Center material: Cost, Security, Reliability, Scalability, Resiliency, and DevOps.

All external contributions *must not have* the following attributes:

- Has limited pattern applicability (fewer than dozens of customers could benefit).
- Incurs unreasonable costs for the consumer unless we specify what type of consumer will benefit from it. For example, if you give us a pattern that only an S500 could afford, that's not much of a pattern.

All external contributions *should have* the following attributes:

- Is well-researched and validated by peers. Should also have been successfully implemented in multiple customers.
- Won't be invalidated by changes to the platform within the next engineering semester.
- Is accessible for reading and adoption by architects or developers at level 200 or higher.

All contributions that contain code samples *should have* the following attributes:

- Use the latest .NET and C# language version for all .NET code.
- Follow the recommended C# naming conventions.
- Use StyleCop to keep your code style consistent across samples.
- Keep simplicity as the main goal for code samples. Anything done in the code that doesn't respect best practices should be explained in code comments. This approach helps readers understand that while things might be done differently in production, another approach was used for simplicity. Avoid overengineering at all costs.
- If there are code assets, identify a person or team to maintain them. If the original author leaves the company, another contact will be made within their organization for assistance. Without a contact, the material might be removed if it's no longer valid or flagged based on our retirement strategy.

Azure Architecture Center contribution program (AAC Contributions)

New content aligned with the preceding content types typically should be developed in concert with the Architecture Center. There are several benefits to this approach:

- Aligns with our content publishing timeline supporting topics and requests already in our pipeline.
- Aligns with relevant product group teams about upcoming feature changes.
- Allows the Architecture Center to provide light project management and facilitation of our process for your document to ensure the accuracy and longevity of your solution.
- Our writers will be able to proactively provide you guidance throughout the publication process to ensure your content meets our content quality and accessibility bar.
- Our team can work to make your images crisp and supportable on the platform.

The Architecture Center alignment process consists of three steps:

1. **Proposal**: This simple document in draft form describes what you want to publish and why. It includes the initial content of the text but doesn't need to include supporting graphics, code samples, or other materials. After we review your draft, we'll provide you with written guidance on whether the content can meet our content needs. We'll tell you which content type it best aligns with. We'll let you know how we can support you in your development. You'll also be assigned a PM who will be your point of contact throughout the process. During this step, we'll seek to answer these main questions:

    a. Is the material technically accurate?
        a. Engineering team from Patterns and Practices (PnP) will help evaluate technical accuracy.
        b. Optionally, SMEs from the targeted Azure technological areas might be consulted. The SME might be a product team contact PM if it's a field contribution, and vice versa.
    b. Does the solution use services or features that might be deprecated soon? Will the solution be invalidated within the next engineering semester?
        a. No part of the solution should be listed on Azure Updates as getting deprecated.
    c. Does the solution use services or features in preview?
        a. Private preview features shouldn't be considered until they're either in public preview or generally available.
        b. If the solution uses public preview features, the article should have a cautionary note at the top.
    d. Is the material repeated elsewhere?
        a. The goal is to provide a consistent story between Architecture Center, Cloud Adoption Framework (CAF), and Learn platforms.
    e. Is the material a better fit elsewhere?
        a. Would this material be a better fit as a blog post or a white paper?
        b. Is this material a cloud migration scenario? If so, it will be a better fit for the CAF platform.
        c. Is this lab-focused training material? If so, it will be a better fit for Learn.
    f. What article format will it follow?
    g. What deployable assets are planned? Who will own them?
        a. Deployable source code or scripts will be required for reference architectures and optional for other article types.
    h. For example scenarios only, has more than one customer validated the material?
    i. What industry trend or content gap within the Architecture Center will this material cover?
    j. Is there a draft write-up present during the intake?
    k. Would the solution proposed by the material be supported by a product group?

2. **Drafts**: If your draft can be included in the Architecture Center, you'll receive notification from our team. Then you can begin refining your document to meet publishing standards based on the content type. During this phase, you're welcome to share drafts with the PM team for basic guidance, grammar, and voice assistance. You can also discuss ideas or ask other questions. You'll be responsible for writing the content and providing all supporting material requested by the PM teams. These items include graphics, code samples, and other artifacts. At the end of this phase, your document should be materially complete.

3. **Publication and post publication**: If your document is approved for publication, final checks will be

performed. The authors will then participate in the Docs Contributor Training and you'll convert your material to Markdown for publication. We'll merge your material, and then it will be your responsibility to maintain the content's accuracy. We might also contact you for an update if a platform change breaks or invalidates your scenario. If the material gets outdated, it might be removed or modified by us to bring it back into alignment.

Step details

Step 1: Proposal

- An external contributor has an idea they want published in the Architecture Center.
- A draft overview of the idea is created and shared with the PnP team.
- The article scope is decided on and aligned to existing content types.
- An `fcp-` prefixed branch in `architecture-center-pr` is created for the content draft.
- The contributor forks the branch into their own repo and adds their article's content to their repo.
  - If you haven't set up your GitHub account for Docs contribution yet, see GitHub account setup.
- The PM managing the content provides feedback and is available for guidance during the writing process to ensure the article meets our guidelines and quality bar.

Step 2: Drafts

- The PM tracks and follows up on the content to ensure it gets completed.
- The contributor opens a PR from their local fork to the draft `fcp-` prefixed branch on `architecture-center-pr`. This action signals that the contributor feels their contribution is ready to enter the review and publishing pipeline.
- The PM and the contributor both agree when the content is ready to move to the publication step.
- The contributor or writers make sure the publication follows the established writing voice and SEO guidance.
- The content is reviewed for technical accuracy by PnP, peers, product or FTE field SMEs (subject matter experts), or the Microsoft technical community.
  - Modifications of various natures might be requested of the contributor at this step, and the PR will be iterated on.
- The PM approves the PR and merges it to the `fcp-` prefixed branch on `architecture-center-pr`.
- A copy-edit pass is carried out by PnP or a vendor.

Step 3: Publication

- The content PM or lead writer gives final approval.
- The TOC and landing page links are created.
- A PR is created by the PM or lead writer, taking the content from the draft `fcp-` prefixed branch to `master`.
- The PR and content are reviewed by the content validation team.
- The author, PnP, or the PM implements fixes to unblock the PR.
- The PM or lead writer approves the PR and signs off.
- The content validation team merges it to the `master` branch.
- The `master` branch is merged with the `live` branch on the Docs release schedule.

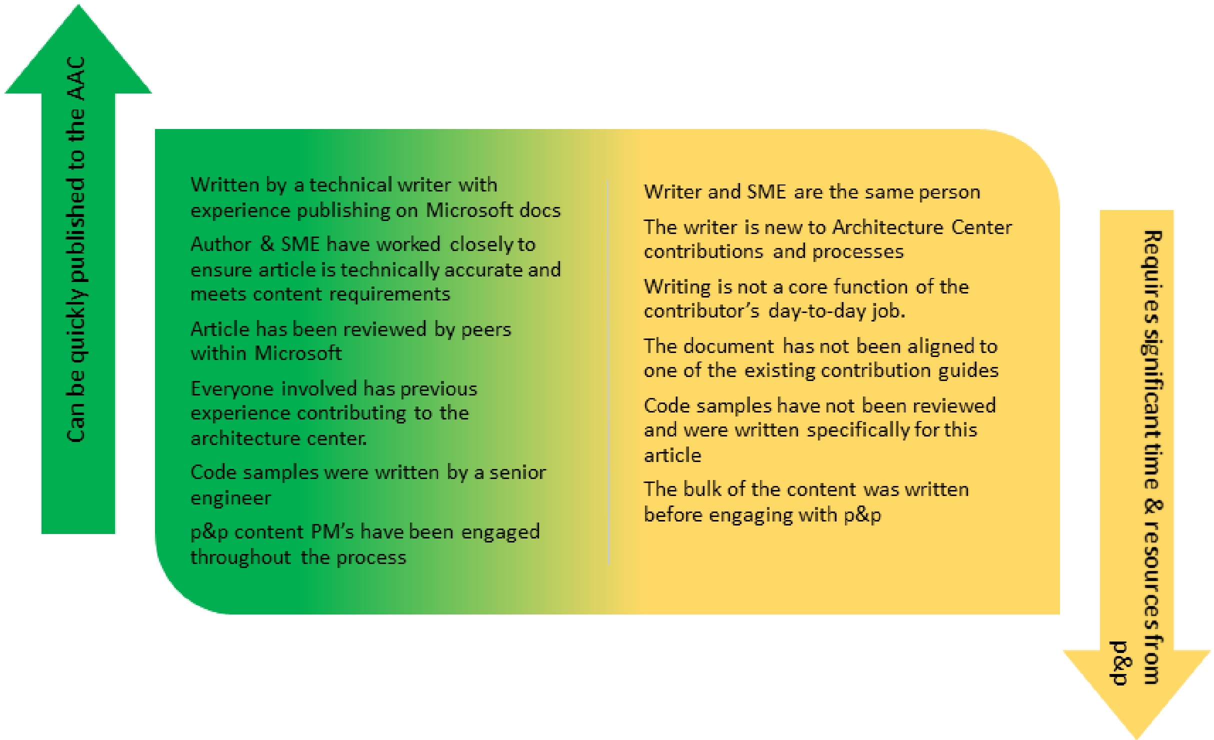
Step 4: Post publication

- **Editorial review**: The lead writer or PM might request an optional editorial review to maintain quality of the article.
- **Performance monitoring**: The PnP team will regularly review performance of AAC contributions and try to improve metrics and SEO. If, in spite of all mitigations, the article continues to perform below standards or doesn't attract traffic, the team will contact the contributors for triage.
- **Content maintenance**: Content freshness is essential for customers' confidence in our platform. For architectural articles and Azure-specific technical guidance, the PnP team will monitor if the article is *abandoned*. An abandoned article is one that hasn't been updated for more than 120 days. If so, PnP will contact the contributor to collaborate on updating the article.

Publishing timelines

New article contributions to the Architecture Center must be reviewed for quality, technical accuracy, and alignment with one of our existing content types before publication. Because of these review requirements, all contributions must be added to our backlog.

Articles that can be quickly moved through our backlog have many of the same qualities, as shown in the following diagram.



Next steps

Now that you are aware of the AAC contribution lifecycle, learn about some important guiding principles for your content:

- If you are new to GitHub, it's very easy to set up your account. Remember to link it to the Microsoft Docs organization for proper access rights.
- When you are ready to start writing, go over this quick reference for partner contributors.
- AAC contributions exclusively use release branches in GitHub to allow team collaboration. Familiarize yourself with release branches.
- Word your content keeping the writing principles in mind. Here's an easy checklist to make sure your contributions have the same voice as the rest of AAC.
- Be mindful of SEO recommendations when writing your content. Search engines provide the biggest entrypoint to our content, and they are quite easy to optimize. At the very minimum, try to:
  - write SEO-friendly titles,
  - write SEO-friendly meta descriptions, and
  - write SEO-friendly H1s. If you need more support, content your PnP PM or writer.
- Once you are ready with your markdown, invite your PnP PM or writer to review your content.
- After the article is published, you may be curious to know how well it is doing with our readers. Check out our content performance dashboard to analyze the signals from your readers.

Authoring content for the Azure Architecture Center Browser

6/24/2021 • 6 minutes to read

Patterns and Practices have adopted a new Azure Architecture Center (AAC) browser similar to the content browser experience as MS Learn and the docs code sample browser. In adopting the new browser experience, all content that is displayed in the browser must use the latest composable documentation model. This article explains how this content is composed for contribution to the AAC browser.

Browser content structure

Content presented in the AAC Browser comprises three or more files:

- A `.yml` file that contains metadata and tagging and renders into the published doc.
- One or more `.md` files that contain the document content.
- A `.png` image file that is used as the thumbnail on the browser card.

All of these are detailed in the following sections.

Article YAML file

The YAML file holds the metadata, filtering criteria, and pointers to all content files. The full schema is found here. Here's an example:

```
### YamlMime:Architecture
metadata:
  title: Manage hybrid Azure workloads using Windows Admin Center
  titleSuffix: Azure Reference Architectures
  description: Deploy Windows Admin Center to manage environments on-premises and in Azure.
  author: neilpeterson
  ms.date: 07/29/2020
  ms.topic: conceptual
  ms.service: architecture-center
  ms.subservice: reference-architecture
azureCategories:
  - hybrid
  - management-and-governance
products:
  - azure
  - azure-portal
  - windows-server
name: Manage hybrid Azure workloads using Windows Admin Center
summary: Deploy Windows Admin Center to manage environments on-premises and in Azure.
thumbnailUrl: /azure/architecture/browse/thumbs/hybrid-server-os-mgmt-wac-azure.png
content: |
  [!include[](hybrid-server-os-mgmt-content.md)]
```

The `.yml` file can be located anywhere in the repo and is picked up for indexing by the presence of the `### YamlMime:Architecture` comment on line 1. Other attributes include:

- `title` is rendered as the article's title (H1).
- `name` is rendered as the title on the browser card.
- `summary` is rendered as a description on the browser card, below the thumbnail.
- `azureCategories` contains categories used as the Categories dimension of filtering selection on the browser.

See Azure Categories for details.

- `products` contains product tags used for Products filtering on the browser. See Products for details.
- `thumbnailUrl` specifies the absolute path to the image used as the thumbnail.
- `content` contains a list of markdown files that make up the body of the article, rendered in the order listed.

Content files (Markdown)

For the AAC browser, article content is authored in Markdown much the same as other Docs content, but with the following differences:

- Do not add metadata to the content file. Metadata for the article is specified in the corresponding YAML file.
- Do not add an H1 or title to the content file. The article title is specified by the `title` metadata in the corresponding YAML file.
- The content files must be named `<yaml-filename>-content.md`. The files are intentionally excluded from the repo build process using this file naming convention.

To add one or more Markdown files into the article, create a list of content files like this:

```
content: |
  [!include[](source-one-content.md)]
  [!include[](source-two-content.md)]
```

Browser thumbnails

Each article included in the browser needs to define an image thumbnail that the AAC Browser displays in the browser card. You'll base the thumbnail image on the architecture diagram image in the article. If your article doesn't have an architecture diagram, use an image from the article that best represents it.

Existing articles already have browser cards with thumbnails. For a new article, you're responsible for creating the thumbnail image and adding that image to the repository. You create a thumbnail manually by using Paint 3D to adjust the aspect ratio so that the AAC Browser renders it correctly.

> **NOTE**
>
> There's no need to resize the image when you create the thumbnail. The docs generator dynamically scales each image to fit into the browser card.

Follow these steps to create a thumbnail image with the correct aspect ratio:

1. Open the **Paint 3D** app.

   If you don't have **Paint 3D**, use any graphics editor that allows you to set the aspect ratio of an image.

2. Select **Menu** from the upper-left corner.

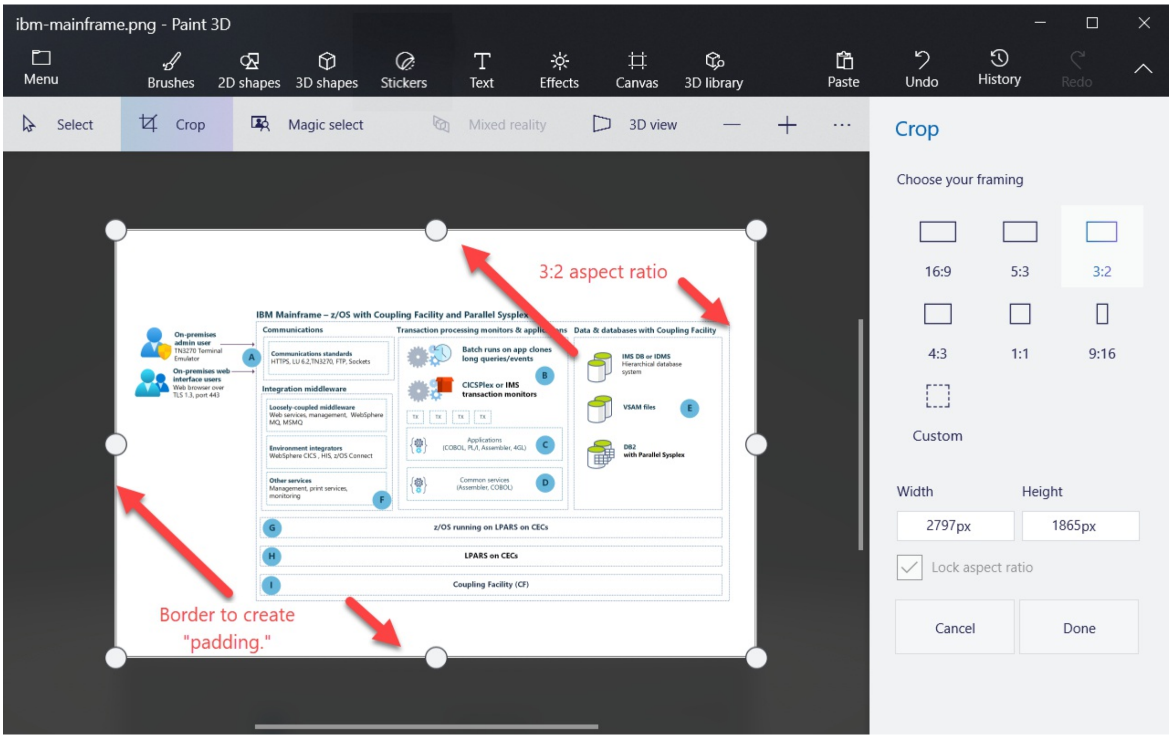
3. Select **Open** > **Browse files** and open the image you need to adjust.

4. With your image open, select **Crop**.

5. From the **Crop** sidebar, select **3:2**.

6. Drag the 3:2 ratio frame over the image until the image is centered and has a border for "padding."

   If the image doesn't fit into the frame, select **Cancel** to exit **Crop**. Then, select **Canvas** from the menu bar, make the canvas bigger, and go back to cropping the image to fit the 3:2 aspect ratio.

7. When you're ready, save the thumbnail image in the `/docs/browser/thumbs/` directory of the AAC repository.

8. Go to the **doc.yml** file that supports your article, and add the file path of your new thumbnail image to `thumbnailUrl:`.

> **IMPORTANT**
>
> As seen in the following example, the thumbnails **must be** referenced in the docs YAML file using an absolute path.
>
> ```
> thumbnailUrl: /azure/architecture/browse/thumbs/<your-thumbnail>.png
> ```

Staging browser cards

The cards found in the browser are not staged for preview when creating a pull request against the default branch. The cards are only available to view once content has been merged into the default branch and published live.

To work around this limitation, we have a staging branch that is indexed by the browser back end: https://review.docs.microsoft.com/azure/architecture/browse/?branch=wi-163353-browse After a PR has been merged into this branch, the new cards can be seen at this location.

> **NOTE**
>
> After new content is merged to the staging branch, the new card might not be visible in the staged browser for 5-10 minutes.

After the content has been verified, re-create the PR against the the default branch.

Azure Categories

The following are valid values for `azureCategories` specified in the YAML file.

```
"ai-machine-learning",
"analytics",
"blockchain",
"compute",
"containers",
"databases",
"developer-tools",
"devops",
"featured",
"hybrid",
"identity",
"integration",
"iot",
"management-and-governance",
"media",
"migration",
"mixed-reality",
"mobile",
"networking",
"security",
"storage",
"web",
"windows-virtual-desktop"
```

Valid values for `azureCategories` are maintained in the https://static.docs.com/ui/latest/schemas/Architecture.schema.json file.

Products

The `products` section of the YAML file lists tags that the browser uses for filtering by product. When specifying tags:

- Limit the list to the architecture's most important components.
- Don't include *azure* in the list unless no other tag is suitable.
- Put the list in alphabetical order.

The following are valid values for `products` specified in the YAML file.

```
"ai-builder",
"aspnet",
"aspnet-core",
```

```
"azure",
"azure-active-directory",
"azure-active-directory-b2c",
"azure-active-directory-domain",
"azure-advisor",
"azure-analysis-services",
"azure-anomaly-detector",
"azure-api-apps",
"azure-api-fhir",
"azure-api-management",
"azure-app-configuration",
"azure-app-service",
"azure-app-service-mobile",
"azure-app-service-static",
"azure-app-service-web",
"azure-application-gateway",
"azure-application-insights",
"azure-arc",
"azure-archive-storage",
"azure-artifacts",
"azure-attestation",
"azure-automation",
"azure-avere-vFXT",
"azure-backup",
"azure-bastion",
"azure-batch",
"azure-bing-autosuggest",
"azure-bing-custom",
"azure-bing-entity",
"azure-bing-image",
"azure-bing-news",
"azure-bing-spellcheck",
"azure-bing-video",
"azure-bing-visual",
"azure-bing-web",
"azure-blob-storage",
"azure-blockchain-service",
"azure-blockchain-tokens",
"azure-blockchain-workbench",
"azure-blueprints",
"azure-boards",
"azure-bot-service",
"azure-cache-redis",
"azure-cdn",
"azure-clis",
"azure-cloud-services",
"azure-cloud-shell",
"azure-cognitive-search",
"azure-cognitive-services",
"azure-communication-services",
"azure-computer-vision",
"azure-container-instances",
"azure-container-registry",
"azure-content-moderator",
"azure-content-protection",
"azure-cosmos-db",
"azure-cost-management",
"azure-custom-vision",
"azure-cyclecloud",
"azure-data-box-family",
"azure-data-catalog",
"azure-data-explorer",
"azure-data-factory",
"azure-data-lake",
"azure-data-lake-analytics",
"azure-data-lake-gen1",
"azure-data-lake-gen2",
"azure-data-lake-storage",
"azure-data-science-vm",
```

```
"azure-data-science-vm",
"azure-data-share",
"azure-database-mariadb",
"azure-database-migration",
"azure-database-mysql",
"azure-database-postgresql",
"azure-databricks",
"azure-ddos-protection",
"azure-dedicated-host",
"azure-dedicated-hsm",
"azure-dev-spaces",
"azure-dev-tool-integrations",
"azure-devops",
"azure-devops-tool-integrations",
"azure-devtest-labs",
"azure-digital-twins",
"azure-disk-encryption",
"azure-disk-storage",
"azure-dns",
"azure-encoding",
"azure-event-grid",
"azure-event-hubs",
"azure-expressroute",
"azure-face",
"azure-farmbeats",
"azure-files",
"azure-firewall",
"azure-firewall-manager",
"azure-form-recognizer",
"azure-front-door",
"azure-functions",
"azure-fxt-edge-filer",
"azure-genomics",
"azure-hdinsight",
"azure-hdinsight-rserver",
"azure-hpc-cache",
"azure-immersive-reader",
"azure-information-protection",
"azure-ink-recognizer",
"azure-internet-analyzer",
"azure-iot",
"azure-iot-central",
"azure-iot-dps",
"azure-iot-edge",
"azure-iot-hub",
"azure-iot-pnp",
"azure-iot-sdk",
"azure-iot-security-center",
"azure-iot-solution-accelerators",
"azure-key-vault",
"azure-kinect-dk",
"azure-kubernetes-service",
"azure-lab-services",
"azure-language-understanding",
"azure-lighthouse",
"azure-linux-vm",
"azure-live-ondemand-streaming",
"azure-live-video-analytics",
"azure-load-balancer",
"azure-log-analytics",
"azure-logic-apps",
"azure-machine-learning",
"azure-machine-learning-designer",
"azure-machine-learning-studio",
"azure-managed-applications",
"azure-managed-disks",
"azure-maps",
"azure-media-analytics",
"azure-media-player",
"azure-media-services",
```

```
"azure-media-services",
"azure-metrics-advisor",
"azure-migrate",
"azure-monitor",
"azure-netapp-files",
"azure-network-watcher",
"azure-notebooks",
"azure-notification-hubs",
"azure-open-datasets",
"azure-personalizer",
"azure-pipelines",
"azure-playfab",
"azure-policy",
"azure-portal",
"azure-powerbi-embedded",
"azure-private-link",
"azure-qna-maker",
"azure-quantum",
"azure-queue-storage",
"azure-rbac",
"azure-redhat-openshift",
"azure-remote-rendering",
"azure-repos",
"azure-resource-graph",
"azure-resource-manager",
"azure-rtos",
"azure-sap",
"azure-scheduler",
"azure-sdks",
"azure-search",
"azure-security-center",
"azure-sentinel",
"azure-service-bus",
"azure-service-fabric",
"azure-service-health",
"azure-signalr-service",
"azure-site-recovery",
"azure-sovereign-china",
"azure-sovereign-germany",
"azure-sovereign-us",
"azure-spatial-anchors",
"azure-speaker-recognition",
"azure-speech",
"azure-speech-text",
"azure-speech-translation",
"azure-sphere",
"azure-spring-cloud",
"azure-sql-database",
"azure-sql-edge",
"azure-sql-managed-instance",
"azure-sql-virtual-machines",
"azure-sqlserver-stretchdb",
"azure-sqlserver-vm",
"azure-stack",
"azure-stack-edge",
"azure-stack-hci",
"azure-stack-hub",
"azure-storage",
"azure-storage-accounts",
"azure-storage-explorer",
"azure-storsimple",
"azure-stream-analytics",
"azure-synapse-analytics",
"azure-table-storage",
"azure-test-plans",
"azure-text-analytics",
"azure-text-speech",
"azure-time-series-insights",
"azure-traffic-manager",
```

```
    "azure-translator",
    "azure-translator-speech",
    "azure-translator-text",
    "azure-video-indexer",
    "azure-virtual-machines",
    "azure-virtual-machines-windows",
    "azure-virtual-network",
    "azure-virtual-wan",
    "azure-vm-scalesets",
    "azure-vmware-solution",
    "azure-vpn-gateway",
    "azure-web-application-firewall",
    "azure-web-apps",
    "azure-webapp-containers",
    "blazor",
    "blazor-server",
    "blazor-webassembly",
    "common-data-service",
    "connected-services-framework",
    "consumer",
    "customer-care-framework",
    "customer-voice",
    "dotnet",
    "dotnet-core",
    "dotnet-standard",
    "dynamics",
    "dynamics-365",
    "dynamics-business-central",
    "dynamics-commerce",
    "dynamics-cust-insights",
    "dynamics-cust-svc-insights",
    "dynamics-customer-engagement",
    "dynamics-customer-service",
    "dynamics-field-service",
    "dynamics-finance",
    "dynamics-finance-operations",
    "dynamics-fraud-protection",
    "dynamics-guides",
    "dynamics-human-resources",
    "dynamics-layout",
    "dynamics-market-insights",
    "dynamics-marketing",
    "dynamics-prod-visualize",
    "dynamics-product-insights",
    "dynamics-project-operations",
    "dynamics-project-service",
    "dynamics-remote-assist",
    "dynamics-retail",
    "dynamics-sales",
    "dynamics-sales-insights",
    "dynamics-scm",
    "dynamics-talent",
    "dynamics-talent-attract",
    "dynamics-talent-core",
    "dynamics-talent-onboard",
    "ef-core",
    "expression",
    "expression-studio",
    "github",
    "hololens",
    "internet-explorer",
    "m365",
    "m365-ems",
    "m365-ems-cloud-app-security",
    "m365-ems-configuration-manager",
    "m365-information-protection",
    "m365-myanalytics",
    "m365-security-center",
    "m365-security-score",
```

```
    "m365-threat-protection",
    "m365-workplace-analytics",
    "mdatp",
    "mem",
    "mem-configuration-manager",
    "mem-intune",
    "microsoft-edge",
    "microsoft-identity-web",
    "microsoft-servers",
    "mlnet",
    "mrtk",
    "ms-graph",
    "msal-android",
    "msal-angular",
    "msal-ios",
    "msal-java",
    "msal-js",
    "msal-node",
    "msal-python",
    "msc",
    "msc-operations-manager",
    "msc-service-manager",
    "office",
    "office-365",
    "office-365-atp",
    "office-access",
    "office-adaptive-cards",
    "office-add-ins",
    "office-bookings",
    "office-excel",
    "office-exchange-server",
    "office-forefront",
    "office-kaizala",
    "office-lync-server",
    "office-onedrive",
    "office-onenote",
    "office-outlook",
    "office-planner",
    "office-powerpoint",
    "office-project",
    "office-project-server",
    "office-publisher",
    "office-skype-business",
    "office-sp",
    "office-sp-designer",
    "office-sp-framework",
    "office-sp-server",
    "office-teams",
    "office-ui-fabric",
    "office-visio",
    "office-word",
    "office-yammer",
    "passport-azure-ad",
    "power-apps",
    "power-automate",
    "power-bi",
    "power-platform",
    "power-query",
    "power-virtual-agents",
    "project-acoustics",
    "qdk",
    "silverlight",
    "skype",
    "sql-server",
    "sql-server-2008",
    "surface",
    "surface-duo",
    "vs",
    "vs-app-center",
```

```
  "vs-code",
  "vs-mac",
  "vs-online",
  "windows",
  "windows-api-win32",
  "windows-azure-pack",
  "windows-forms",
  "windows-iot",
  "windows-iot-10core",
  "windows-mdop",
  "windows-mixed-reality",
  "windows-server",
  "windows-smb-server",
  "windows-system-center",
  "windows-uwp",
  "windows-virtual-desktop",
  "windows-wdk",
  "windows-wpf",
  "xamarin",
  "xbox"
```

Valid values for `products` are maintained in the https://static.docs.com/ui/latest/schemas/Architecture.schema.json file.

Azure Architecture Center refresh

6/28/2021 • 3 minutes to read

This project aims to ensure that Azure Architecture Center content (deployments, diagrams, and copy) are up to date, well-architected, and aligned. Additional goals are to ensure deployments are continuously validated and remove AZBB from the Azure Architecture Center.

This document provides an overview of the refresh process with links to examples, and other assets that can be used as additional content is refreshed.

Published components

First, let's define the components that are included in the 2021 content refresh. For each article, these items should be present.

COMPONENT (CLICK FOR MORE DETAILS)	DESCRIPTION	EXAMPLE
Code sample *	Code + details in a mspnp GitHub repository.	secure hybrid network
Validation pipeline *	A simple validation pipeline to test each ARM deployment on pull request and on a schedule.	secure hybrid network
Diagram	Visual representation of the architecture / solution.	secure hybrid network
Reference architecture or copy	Content on docs.microsoft.com.	secure hybrid network

Code sample

For Azure infrastructure and **where applicable**, include a set of ARM templates and deployment instructions. For each deployment, ensure:

- Best practices found here Azure ARM Templates QuickStart Best Practices Doc.
- Deployment is well architected (highly available, diagnostics enabled, include Azure Monitor queries, dashboarding, etc).
- Store deployment here https://github.com/mspnp/samples.
- Include a validation pipeline for each deployment (detailed later in this doc).

Code samples browser

The readme for each deployment should include YAML front matter, which allows the sample to be indexed and published on the docs.microsoft.com code samples browser. More information can be found here.

The YAML front matter looks like this:

```
---
page_type: sample
languages:
- azurepowershell
- azurecli
products:
  - azure-app-service
  - azure-log-analytics
  - azure-key-vault
  - azure-sql-database
description: description goes here
---
```

Link to reference architecture

The readme for each deployment should include a link back to the related reference architecture. Use the docs.microsoft.com markdown button for the link. The markdown looks like this:

```
For detailed information, see the Implement a secure hybrid network:

> [!div class="nextstepaction"]
> [Implement a secure hybrid network](https://docs.microsoft.com/azure/architecture/reference-architectures/dmz/secure-vnet-dmz)
```

Build badge

The readme for each deployment should include the Azure Pipeline build badge associated with the automated validation (reference next section of this doc). The markdown used for the build badge can be found on the settings of the Azure Pipeline and will look like this:

```
[![Build Status.](https://nepeters-devops.visualstudio.com/arm-template-validation-pipelines/_apis/build/status/secure-hybrid-network?branchName=master)](https://nepeters-devops.visualstudio.com/arm-template-validation-pipelines/_build/latest?definitionId=135&branchName=master)
```

Validation pipeline

Where applicable, create a simple set of automated validation tasks using Azure Pipelines. At a minimum, complete these things:

- Run ARM TTK tests and publish test results.
- Validate template against validation API.
- Deploy template twice (ensure frictionless idempotency).
- Delete deployment assets.

A sample pipeline with annotations can be found here - secure hybrid network.

Pipeline YAML

Store the validation pipeline YAML in the same directory as the main ARM template.

Azure DevOps project

This Azure DevOps project can be used to run the validation pipeline - arm-template-validation-pipelines. If you need access, reach out to Neil Peterson.

Validation Dashboard

Add a section to this Azure DevOps dashboard with a validation badge. The markdown used for the build badge can be found on the settings of the Azure Pipeline and will look like this:

```
[![Build Status.](https://nepeters-devops.visualstudio.com/arm-template-validation-pipelines/_apis/build/status/secure-hybrid-network?branchName=master)](https://nepeters-devops.visualstudio.com/arm-template-validation-pipelines/_build/latest?definitionId=135&branchName=master)
```

Diagram

Update each diagram so that it is technically aligned with the deployment. Include things beyond resources such as network flow, diagnostic flow, and availability configurations. Some other things to consider:

- Make the main diagram short and wide. This size allows us to present it on top of the article without taking up the first page with art.
- Use updated icons where applicable.
- Consider using a shadow box for inserting a larger diagram or more detailed diagram. For more information on using shadow box, see, Create and expandable screenshot.

Copy update

Finally, update the copy of the article to match any changes to the deployment. This would be a good time to give the completed article a refresh pass. While doing so, check that:

- Everything in the copy up to date.
- Any additional screenshots to update.
- Can any extra copy be removed?

End to end examples

These articles have gone through this refresh process and can be used as examples.

- Hub and Spoke
- Basic Web Application
- Secure hybrid network

AAC refresh feedback

Raise any feedback and / or suggest options that you feel would improve the Azure Architecture Center better to the pnp team.

Create a reference architecture

3/24/2021 • 5 minutes to read

A **reference architecture** shows a proven architecture that a customer can deploy today on Azure.

It gives detailed recommendations on how to design and deploy the architecture to Azure. Also it includes guidance on the quality pillars (availability, scalability, security, operations) as they relate to the architecture. Finally, a reference architecture should (in most cases) include a deployment artifact such as an Azure Resource Manager template that incorporates the recommendations and best practices.

Target Audience: TDM, Architect, Infra/DevOps

Audience Level: 300-400

Examples: Azure reference architectures

Purpose

A reference architecture gives customers a “golden path” for a particular scenario or workload. If you follow this path, you'll avoid common pitfalls.

It also gives customers a jump-start to design their own production-ready deployments, using the provided scripts and templates as a starting point.

Reference architectures aren't quickstarts. They're intended to help customers plan real-world deployments, so reading time can be 10-30 minutes as needed to cover the details.

Reference architectures are also not tutorials. First, a reference architecture is oriented around a scenario or solution, rather than a task. Second, a reference architecture will generally point to existing how-to topics for individual tasks.

Contributing

In order to contribute a Reference Architecture, please follow the below outline and keep in mind these base requirements:

- Is it generalized and canonical? (A reference architecture represents the most recommended settings per each architecture style.)
- Does it include a reference implementation?
- Was the reference implementation perf tested
- Best practices for security, scalability, etc.

TOC and URL

Reference architectures are located in the Reference Architectures node of the Architecture Center TOC.

Related architectures are grouped into subnodes under Reference Architectures node.

The URL structure follows the same pattern as the TOC nodes:

```
/azure/architecture/reference-architectures/<group>/name-of-architecture
```

Document structure

We use a consistent structure for reference architecture documents.

H1 title - State the scenario

> Example: "Connect an on-premises network to Azure using ExpressRoute"

After the title, there's an introductory section, consisting of:

Intro paragraph (no heading)

A short paragraph that describes the scenario and the solution that is deployed. This paragraph should be concise, so that the diagram (next item) appears close to the top of the page.

Example:

> This reference architecture shows a set of proven practices for running multiple Linux virtual machines (VMs) behind a load balancer, to improve availability and scalability. This architecture can be used for any stateless workload, such as a web server, and is a foundation for deploying n-tier applications.

Architecture diagram

The diagram shows the Azure resources that get deployed. The level of detail will depend on the scenario.

> Example: For an IaaS deployment, the diagram *should* include the network topology (VNet, subnets, gateways, public IP addresses), VMs or VM scale sets, plus load balancers, Traffic Manager, and App Gateway, as appropriate. It *might* include other elements such as NICs, storage accounts, or VHDs, but only if the document contains specific guidance related to those elements. For example, if some VMs need multiple NICs, include those in the diagram. Otherwise, it can be assumed that each VM has a NIC.

H2 - Architecture

This section is a bullet list that describes the elements in the diagram. In most cases, there should be a 1:1 correspondence. Everything in the diagram should be described here.

For each element, describe its purpose within the context of the overall solution.

- **Weak:** "A subnet is a way to segment a VNet into multiple address spaces."**
- **Better:** "Put each application tier into a separate subnet."

It's OK to do both, describing *what* something is and *why* it's there. But avoid repeating information that's already in the product documentation for that service/feature. Instead, link to the product documentation.

H2 - Recommendations

This section gives more detailed recommendations for the elements in the previous section. Put all of the messy details and here.

This section can have subheadings (H3) to organize the recommendations. For example, "VM recommendations", "SQL Server recommendations", "Networking recommendations"

Examples of recommendations:

> - For each tier, put two or more VMs in an availability set.
> - Create a separate storage account for diagnostic logs.
> - For SQL Server high availability, we recommend using Always On Availability Groups.

As much as possible, provide the justification or rationale for each recommendation. Examples:

> - Put two or more VMs in an availability set. This makes the VMs eligible for a higher SLA.

> - Create a separate storage account for diagnostic logs to avoid hitting IOPS limits.

Recommendations should be definite when possible. In practice, sometimes you'll need a qualifier like "depending on your workload..."

H2 - Considerations (pillars)

After the recommendations section, include sections (H2) that describe non-functional characteristics of the architecture.

- Performance considerations
- Scalability considerations
- Availability considerations
- Manageability considerations
- Security considerations
- Cost considerations

Only include the sections that are relevant for the reference architecture. In some cases, there won't be anything to say for a particular pillar, especially if a reference architecture builds on a previous reference architecture, as part of a series.

These sections may include whatever discussion points are relevant for that pillar, with respect to the architecture. Including:

- What the architecture gives you.

  > "Scale out by adding more VMs to the load balancer pool." (Scalability) ← **A capability that this architecture gives you.**

- What the architecture doesn't give you. When possible, point to a possible alternative.

  > "If you need higher availability, replicate the application across two regions and use Traffic Manager for failover." (Availability) ← **What this architecture does *not* give you (availability during a regional outage), along with a pointer to something actionable.**

- Things to keep in mind to achieve the pillar, when using this architecture, including design and operational considerations.

  > "If your VMs run an HTTP server, create an HTTP health probe. Otherwise, create a TCP health probe." (Availability) ← **A best practice that depends on the specific application workload**

  > "The web front end should be stateless, to avoid the need to maintain client affinity." (Scalability) ← **An app dev consideration.**

- Additional options to consider, which may not be part of the golden path.

  > "If you need additional throughput, consider creating additional Event Hubs and sharding the messages." (Scalability) ← **Advanced option, not shown in the reference architecture.**

H2 - Deploy the solution

If there's a deployable asset (Azure Resource Manager template or script), include instructions for how to deploy.

Markdown templates

The markdown templates for a reference architecture are at Reference architecture markdown templates.

Create an example workload

6/24/2021 • 4 minutes to read

Example workloads guide customers through the design process of solving specific problems Azure. They provide actionable architecture guidance based on real customer examples. Their goal is to shorten a customer's learning curve by telling them the story of another customer who has been there before.

Each example workload has an architecture diagram, discusses decisions that went into the architecture, explores some alternatives, and includes cost estimates based on the pricing calculator. It may include a Resource Manager template to deploy.

Target Audience: TDM, Architect, Lead Developer

Adoption Stage: Planning

Audience Level: 100-200

Example of this content type: Baseball ML workload

Tenets

Example Workloads adhere to the following tenets:

- **Accelerate Customer Adoption**: Example Workloads accelerate customer adoption by providing guidance and lessons learned from others who have been there before.
- **Easy to Consume**: Workloads should be easy to consume for audiences of all technical levels.
- **Solve business problems**: Workloads use technology to solve problems facing the business, they aren't focused on the technology itself.
- **Based on real deployments**: They are either built from a customer deployment or set of common patterns derived from customer deployments.
- **Teaching to fish**: Example Workloads talk about the decisions that were made, what their constraints were, and the thought process that lead to the solution. They don't give guidance without explaining why.

What a workload isn't

- An Example Workload is not a step-by-step implementation guide or a tutorial
- Workloads aren't built to solve technology problems.
  - Things like setting up a load balancer, building a VPN, or running micro-services belong in other documents.
- They are not focused on a single specific technology or Azure service.

Purpose

Example workloads are similar in many ways to reference architectures, but have a different customer intent:

- Example workloads help a customer to understand the different types of architectures that are possible on Azure. After reading an example workload, the next step is to go deeper into the architectural guidance or go to the product documentation to learn more about the services used.
- Workloads are specific to an industry, vertical, or application domain. Reference architectures are more general ("N-tier application")
- Workloads are "bite sized"; they can be consumed more quickly than a reference architecture.

- Workloads aim for level 100-200, where reference architectures are built for a more technical audience.
- A reference architecture is Microsoft's recommendation for that architecture. It's a "golden path" that customers can follow. An example workload shows one way that a customer has solved their specific domain problem.

TOC and URL

All articles are located in the "Example Workloads" node of the Architecture Center TOC.

Related workloads are grouped into subnodes. The URL structure follows the same pattern as the TOC nodes:

```
/azure/architecture/example-workload/<group>/name-of-workload
```

Document structure

Example workloads have a consistent structure.

Markdown templates

The markdown templates for an example workload, example scenario, or sample solution are at Sample solution markdown templates.

Title

The title is a noun phrase that describes the workload.

> Example: "Insurance claim image classification on Azure"

Avoid naming the workload after the Azure technologies that are used.

Introductory section (no heading)

The introduction contains:

- A paragraph that describes what the solution does (the domain)
- A paragraph that contains a brief description of the main Azure services that make up the solution. This paragraph should convey the Azure value proposition, not a complete description of the architecture (which comes later).

Potential use cases

A bullet list of example use cases for this solution. The purpose is to show that an example architecture may apply to more than one application domain.

Architecture

This section starts with an architecture diagram. The diagram shows the Azure resources in the solution plus the data flow. It may include other elements such as a user, a mobile device, or an external data source.

After the architecture diagram, include a numbered list that describes the data flow through the solution. Start from the user or external data source and then through rest of the solution.

Components

A bullet list of components in the architecture (including all relevant Azure services) with links to the product documentation.

Alternatives

Use this section to list alternative Azure services or architectures that you might consider for this solution. Include the reasons why you might choose these alternatives.

Considerations (pillars)

Within this section, describe the non-functional characteristics of the architecture, under the following H3 headers:

- Scalability considerations
- Availability considerations
- Security considerations
- Resiliency considerations

Deploy the solution or See it in Action

If there's a deployment (Azure Resource Manager template or script), include instructions for how to deploy. This is greatly encouraged (even a small sample would be valuable), but it is not required.

Pricing

For this section, you should create three cost estimates for small, medium, and large deployments, using the Azure pricing calculator. Save and share the estimates, and include links to the three estimates in this section.

The exact definition of small, medium, and large will depend on the workload. Describe what the sizes mean for this workload. Size is defined by throughput or scale.

Next steps

This section walks customers through their next steps when they want to start building this out. It can link to additional training that may be required, if there are any reference architectures already built that cover this solution.

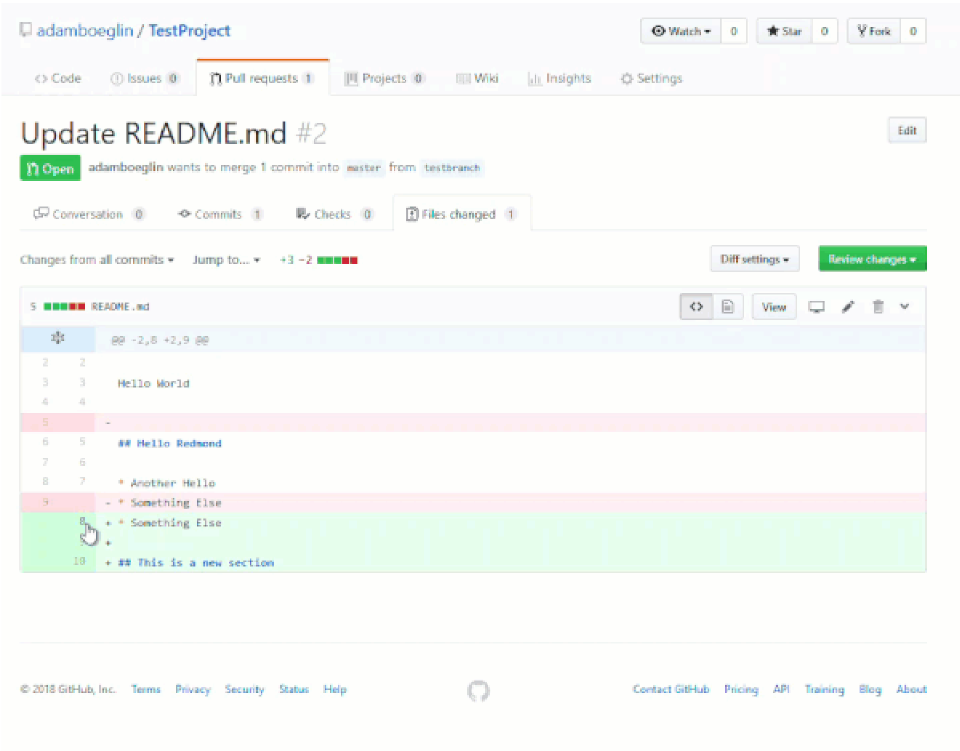
Related resources

Links to other resources that are relevant such as product documentation, case studies, or other workloads.

Article reviews

Before an article is published, it goes through both an editorial review and a technical review to validate the article and look for any technical inaccuracies or areas that should have been covered.

You'll need a GitHub account if you don't have one already. Using your personal account is fine. Once your account is setup, reviewing is easy. Navigate to the "Files" section of the pull request you're reviewing. Then click the “+” by the line number you'd like to comment on, add your notes, and click “Start Review”. You can then add multiple comment lines and click “Finish Review” when complete.



adamboeglin / TestProject
Watch 0
Star 0
Fork 0
Code
Issues 0
Pull requests 1
Projects 0
Wiki
Insights
Settings
Update README.md #2
Edit
Open
adamboeglin wants to merge 1 commit into master from testbranch
Conversation 0
Commits 1
Checks 0
Files changed 1
Changes from all commits
Jump to...
+3 −2
Diff settings
Review changes
5 README.md
View
@@ -2,8 +2,9 @@
Hello World
Hello Redmond
* Another Hello
- * Something Else
+ * Something Else
+
+ ## This is a new section
© 2018 GitHub, Inc.
Terms
Privacy
Security
Status
Help
Contact GitHub
Pricing
API
Training
Blog
About

Create a solution idea

3/24/2021 • 2 minutes to read

Solution ideas provide brief overviews of solutions that customers can implement using Azure services.

Each solution idea has an architecture diagram, data flow, overview of the services used, and next step links.

Target Audience: CXO, TDM, Architect, Developer

Adoption Stage: Planning

Audience Level: 100-200

Examples of this content type: Solution Ideas

Document structure

Below you'll find the structure of a typical Solution idea

Title

The title is a noun phrase that describes the scenario.

> Example: "Back up on-premises applications and data to Azure"

Azure technologies can be used in the name if the goal is to highlight solving a specific problem with a specific Azure technology.

Introductory section (no heading)

The introduction contains:

- A paragraph that describes what the solution does (the domain)
- A paragraph that contains a brief description of the main Azure services that make up the solution. This paragraph should convey the Azure value proposition, not a complete description of the architecture (which comes later).

Potential use cases

A bullet list of example use cases for this solution. The purpose is to show that an example architecture may apply to more than one application domain.

Architecture

This section starts with an architecture diagram. The diagram shows the Azure resources in the solution plus the data flow. It may include other elements such as a user, a mobile device, or an external data source.

After the architecture diagram, include a numbered list that describes the data flow through the solution. Start from the user or external data source and then through rest of the solution.

Components

A bullet list of components in the architecture (including all relevant Azure services) with links to the product documentation.

Next steps

This section walks customers through their next steps when they want to start building this out. It can link to additional training that may be required, if there are any reference architectures already built that cover this solution.

Related resources

Links to other resources that are relevant such as product documentation, case studies, or other scenarios.

Markdown templates

The markdown templates for a solution idea are at Solution idea markdown templates.

Create a design pattern

3/24/2021 • 3 minutes to read

A software design pattern is a "reusable solution to a commonly occurring problem ... in software design" (Wikipedia). Cloud design patterns address specific challenges in distributed systems. They often address a quality pillar such as availability, or resiliency.

The qualities of a design pattern include:

- Proven. A design pattern emerges from actual implementations that have been shown to work. Often, a pattern is identified after the fact, by observing a common approach that appears repeatedly.
- Abstract. A design pattern is not the same thing as an implementation of that pattern. Design patterns are described in terms of objects or functional blocks that communicate or exchange data. An implementation may use a particular language or framework
- Generalized. A design pattern solves a problem that appears across multiple application domains. Design patterns are the building blocks of application architectures.
- Technology agnostic. A design pattern does not depend on a specific technology. For example, *publish-subcribe* is a design pattern, and Azure Service Bus is one way to implement the pattern.

Target Audience: Lead developer

Audience Level: 200-400

Find the current catalog of design patterns here.

Purpose

Provide customers with a catalog of patterns that they can use like a toolkit.

- A design pattern distills a set of real-world proven practices. They're technology agnostic. For example, “publish/subscribe” is a pattern, and using Azure Service Bus is one way to implement this pattern.
- The catalog of design patterns supports other architectural guidance, such as design guides and best practices articles. They allow us to refer to the patterns without explaining them each time. (The reason they're called “patterns” is because they show up in many places!)
- Design patterns help customers reason about their architectures from a functional/logical viewpoint, before worrying about the specifics of individual technologies.
- Design patterns aren't specific to a particular scenario, technology, or programming language. Usually there's more than one way to implement a pattern.

TOC and URL

Cloud Design Patterns are located in the Cloud Design Patterns node of the Architecture Center TOC.

The URL for a Cloud Design Pattern article is:

```
/azure/architecture/patterns/name-of-pattern
```

Categories

The design patterns are grouped into the following categories:

- Availability
- Data management
- Design and implementation
- Messaging
- Management and monitoring
- Performance and scalability
- Resiliency
- Security

A pattern will generally fall under at least one of these categories, and sometimes more than one. Under the Design Patterns landing page, there's a Categories node that has subpages for each of the categories. These allow customers to browse by category.

Document structure

A Design Pattern article follows a fixed structure that's inspired by the original book on software design patterns, Design Patterns: Elements of Reusable Object-Oriented Software.

Title

The title is the name of the pattern. If the pattern is already established within the developer community, use the standard name for the pattern.

After the title, include a short paragraph that summarizes the pattern. This paragraph should briefly describe the pattern and the reasons to use it. Don't include a lot of detail. The remainder of the article will go into the details. The purpose of the summary is for the reader to quickly decide whether the pattern is relevant to their problem.

Context and problem

This section describes the problem that the design pattern solves. Provide enough context to motivate the discussion of the problem. Often, the context involves an application's non-functional requirements, such as throughput, maximum latency, or concurrency.

Solution

This section describes the pattern. Describe how the pattern works, including the objects or functional blocks and their interactions. Also describe how the pattern solves the problem. When possible, include a diagram of the pattern. Don't use this section to describe any specific implementation of the pattern.

Issues and considerations

Use this section to describe considerations such as:

- Considerations for implementing the pattern.
- Challenges that you might face when implementing or using the pattern.
- Tradeoffs or possible negative implications of using a pattern.

When to use this pattern

Describe, in general terms, the situations when you would use this pattern. If possible, also list examples of when the pattern is not useful.

Example

Use this section to provide a concrete example of how to implement the pattern. In some cases, the example is a code sample. In other cases, the example might just describe how to use an Azure service to implement the pattern.

Markdown template

The markdown template for a design pattern is at Design pattern markdown template.

Create a topic landing page

5/5/2021 • 3 minutes to read

A topic landing page should be thought of as a hub that links to articles that already exist across documentation, blogs, GitHub, MSLearn, and the broader internet (if MS content is not available). There should be little to no content created specifically for this page.

The format was inspired by a recent user study that we've completed around how customers consume documentation. These pages were built to address the following high-level learnings from that study:

- Provide pathways that allow developers to learn through example projects and videos.
- Ensure that documentation has clear pathways for next steps.
- Participants desire case studies, tutorials, and logical links to other resources in the documentation so that they can learn how to effectively deploy to the cloud.

Topic focused pages should be a sort of virtual whitepaper helping customers to see and understand a path through our documentation and be a "home base" for learning about a specific technology on Azure. Think of everything required to be successful when implementing complex topics such as machine learning, IOT, microservices, and HPC. These require knowledge that spans products and services, relies on a foundation of knowledge around specific technologies, and are complex enough that there's no one single way of doing it. The topic landing page should help current and prospective customers feel confident that they're heading the right direction and not missing anything.

Example Page: https://docs.microsoft.com/azure/architecture/topics/high-performance-computing/index

Building the page

The topic pages should follow these guidelines:

- Assume the audience has no knowledge of the topic and provide a quick introduction to it
- Be visually interesting
  - Have a video outlining the topic & solutions at the very top
  - Include images and links to architecture diagrams and solutions
- Walk the customer through everything they would need to know about a given topic
  - Start at the basics and continue to add depth
  - Provide lots of examples
- Provide opportunities to go hands on with code & walk-throughs
- Outline clear next steps such as contacting sales, launching a demo, and viewing pricing information
- Include customer case studies and examples of others successfully implementing this solution

Start with an outline of the page before linking to any content. First outline your h2 headers covering the important high-level concepts and topics that are necessary for customers to fully understand this technology area. Then, locate the documentation required to fill in those areas and add explaining text to describe why each of these areas is important and what they're used for. When everything is complete, a Contextual TOC will be built to provide a cohesive experience when navigating between pages and breadcrumbs will be set up to help customers understand where they are across pages. This will be created by the Architecture Center team.

Markdown template

Use this Markdown template when creating a topic page:

```
---
title: TOPIC NAME on Azure
description: A guide to building running TOPIC NAME on Azure
author:
ms.date: 2/4/2019
---
<!-- markdownlint-disable MD033 -->
<!-- markdownlint-disable MD026 -->

# TOPIC NAME

## Introduction to TOPIC

<!-- markdownlint-disable MD034 -->

> [!VIDEO https://www.youtube.com/embed/]

<!-- markdownlint-enable MD034 -->

<description of Topic>

## Implementation checklist

As you're looking to implement TOPIC, ensure you're reviewed the following topics:

<!-- markdownlint-disable MD032 -->

> [!div class="checklist"]
> - Important [SECTION1](#section1)
> - Important [SECTION2](#section2)

<!-- markdownlint-enable MD032 -->

## SECTION1


### SUB-SECTION

This is a template that allows for the card layout on topic pages.  Use it for links to scenarios, reference
architectures, and solutions.  Only the highlighted sections (link, image, title, & description) will need
to be changed.

<ul class="columns is-multiline has-margin-left-none has-margin-bottom-none has-padding-top-medium">
    <li class="column is-one-third has-padding-top-small-mobile has-padding-bottom-small">
        <a class="is-undecorated is-full-height is-block"
            href="/azure/architecture/example-scenario/apps/hpc-saas?
context=/azure/architecture/topics/high-performance-computing/context/hpc-context">
            <article class="card has-outline-hover is-relative is-fullheight">
                    <figure class="image has-margin-right-none has-margin-left-none has-margin-top-none has-
margin-bottom-none">
                        <img role="presentation" alt="" src="../../example-scenario/apps/media/architecture-
hpc-saas.png">
                    </figure>
                <div class="card-content has-text-overflow-ellipsis">
                    <div class="has-padding-bottom-none">
                        <h3 class="is-size-4 has-margin-top-none has-margin-bottom-none has-text-
primary">Computer-aided engineering services on Azure</h3>
                    </div>
                    <div class="is-size-7 has-margin-top-small has-line-height-reset">
                        <p>Provide a software-as-a-service (SaaS) platform for computer-aided engineering
(CAE) on Azure.</p>
                    </div>
                </div>
            </article>
        </a>
    </li>
    <li class="column is-one-third has-padding-top-small-mobile has-padding-bottom-small">
        <a class="is-undecorated is-full-height is-block"
```

```
            href="/azure/architecture/example-scenario/infrastructure/hpc-cfd?
context=/azure/architecture/topics/high-performance-computing/context/hpc-context">
            <article class="card has-outline-hover is-relative is-fullheight">
                    <figure class="image has-margin-right-none has-margin-left-none has-margin-top-none has-
margin-bottom-none">
                        <img role="presentation" alt="" src="../../example-
scenario/infrastructure/media/architecture-hpc-cfd.png">
                    </figure>
                <div class="card-content has-text-overflow-ellipsis">
                    <div class="has-padding-bottom-none">
                        <h3 class="is-size-4 has-margin-top-none has-margin-bottom-none has-text-
primary">Computational fluid dynamics (CFD) simulations on Azure</h3>
                    </div>
                    <div class="is-size-7 has-margin-top-small has-line-height-reset">
                        <p>Execute computational fluid dynamics (CFD) simulations on Azure.</p>
                    </div>
                </div>
            </article>
        </a>
    </li>
 </ul>


## Cost or pricing

Details of how much this will cost and what effects it

## Security

For an overview of security best practices on Azure, review the [Azure Security Documentation]
(/azure/security/azure-security?context=/azure/architecture/topics/high-performance-computing/context/hpc-
context).


Anything else that would relate to security


## Customer stories

Links to case studies or customer stories running this workload

## Other important information


## Next steps
```

Markdown template for design patterns

6/11/2021 • 4 minutes to read

Use this Markdown template to create a design pattern.

For guidance, see Create a design pattern.

```
---
title: Replace this text based on the guidance in Completing the metadata under the Article Title section.
description: Replace this text based on the guidance in Completing the metadata under the Article Title
section.
keywords: Replace this text based on the guidance in Completing the metadata under the Article Title
section.
ms.date: mm/dd/yyyy
ms.service = architecture-center
ms.topic = conceptual
ms.subservice = design-pattern
pnp.series.title: Cloud Design Patterns
pnp.pattern.categories: Replace this text based on the guidance in Completing the metadata under the Article
Title section.
---

# Article title

_Your article can have only one **H1 heading (#)**, which is the article title. The H1 heading is always
followed by a succinct descriptive paragraph that informs the reader what the article is about and how it
can help them. Do not start the article with a note or tip. Descriptive text always appears in italics.
Descriptive text should be deleted or replaced with your text prior to publication._

_This template is specific to **Cloud Design Patterns**. Your article should describe the problem that the
pattern addresses, identify considerations for applying the pattern, and provide an example based on
Microsoft Azure. Most of the patterns include code samples or snippets that show how to implement the
pattern on Azure. Your article should be broken down into six subheadings (H2, ## in markdown)--Context and
problem, Solution, Issues and considerations, When to use this pattern, Example, and Next steps. The H2
headings and descriptions are included in this template. If you need to create a new heading under one of
the H2 headings, use an H3 heading (###)._

_**Completing the metadata:**_
_This section provides guidance on completing the metadata section at the top of this template. Update the
placeholder text based on the following guidance:_

- _**title:** Use the H1 (#) title of your article from the top of this section. Both titles should be
identical. Maximum recommended length is 60 characters._
- _**description:** Provide a brief summary of your article. This is the description that appears in search
engine results, so ensure the summary is clear and concise and attracts your intended audience. Maximum
recommended length is 150-160 characters._
- _**keywords:** Add a comma-separated list of key concepts and terms from your article. These are the words
your intended audience will submit in a search engine._
- _**ms.date:** Enter the date in mm/dd/yyyy format, as shown in the metadata field. Initially this should
be the date your article is published. After publication, this field should be refreshed whenever the
article is updated so readers can see that the content is fresh._
- _**pnp.series.title:** Cloud Design Patterns is always the correct patterns and practices series title for
this template. Do not change._
- _**pnp.pattern.categories:** Review the bracketed list that follows and delete all categories that don't
apply to your article: [availability, data-management, design-implementation, messaging, management-
monitoring, performance-scalability, resiliency, security]. Copy and paste the final bracketed list into the
metadata section._

## Context and problem

_Provide a brief background on the specific pattern and the problem that the pattern addresses._
```

```
## Solution

_Describe the solution for the problem identified in the **Context and problem** section. If there are
multiple solutions, put them in order of complexity and provide instructions on how the reader can choose
the best solution for their given problem. Best practices are to provide a step-by-step approach to
implement the solution, and to include screenshots to help guide the reader._

## Issues and considerations

_Highlight pertinent concerns relevant to implementing the solution provided in the previous section. Best
practices are to start this section with the following boilerplate sentence, followed by a bulleted list of
topics for the reader to contemplate prior to implementing the given solution._

Consider the following points when deciding how to implement this pattern:

- Item 1
- Item 2
- Item 3

## When to use this pattern

_Deliver specific guidance on when this pattern should and should not be used. Use the following boilerplate
text, followed by bullet points, to help the reader determine if the solution is applicable to their
specific scenario._

Use this pattern when:

- Item 1
- Item 2
- Item 3

This pattern might not be suitable:

- Item 1
- Item 2
- Item 3

## Example

_Include a working sample that shows the reader how the pattern solution is used in a real-world situation.
The sample should be specific and provide code snippets when appropriate._

## Next steps

_Provide links to other topics that provide additional information about the pattern covered in the article.
Topics can include links to pages that provide additional context for the pattern discussed in the article
or links to pages that may be useful in a next-steps context. Use the following boilerplate sentence
followed by a bulleted list._

The following information may be relevant when implementing this pattern:

- Item 1
- Item 2
- Item 3

## Related guidance

_Bulleted list that links to one or more related patterns._
```

Markdown and YML templates for reference architectures

6/25/2021 • 2 minutes to read

Use these Markdown and YML templates to create a reference architecture.

For guidance, see Create a reference architecture.

Each article must have two files. Follow the instructions for browse format authoring at Authoring content for the Azure Architecture Center Browser.

Name this file *<article-title>.yml*:

```
### YamlMime:Architecture
metadata:
  title: <H1 title of your article. Maximum recommended length is 60 characters>
  titleSuffix: Azure Reference Architectures
  description: <A brief summary of your article. This is the description that appears in search engine
results. Maximum recommended length is 150-160 characters.>
  author: <github username>
  ms.author: <Microsoft email alias>
  ms.date: <publish or major update date - mm/dd/yyyy>
  ms.topic: conceptual
  ms.service: architecture-center
  ms.subservice: reference-architecture
azureCategories:
  - <choose at least one category from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#azure-categories>
  - <there can be more than one category>
products:
  - <choose 1-5 products from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#products>
  - <1-5 products>
  - <1-5 products>
name: <H1 title>
summary: <Write a summary. Can be same as description.>
thumbnailUrl: /azure/architecture/browse/thumbs/<browse-png-filename>.png
content: |
  [!include[](<article-filename>-content.md)]
```

Name this file *<article-title-content>.md*:

```
> The H1 title is the same as the title metadata. Don't enter it here, but as the **name** value in the
corresponding YAML file.

_Brief introduction goes here._ [**Deploy this solution**.](#deploy-the-solution)

![alt text.](./media/folder_name/architecture-diagram.png)

_Download a [Visio file](https://arch-center.azureedge.net/architecture.vsdx) that contains this
architecture diagram. This file must be uploaded to `https://arch-center.azureedge.net/`_

## Architecture

The architecture consists of the following components:

- **Thing 1**. Description

- **Thing 2**. Description
```

```
- **Thing 2**: Description

## Recommendations

The following recommendations apply for most scenarios. Follow these recommendations unless you have a
specific requirement that overrides them.

_Include considerations for deploying or configuring the elements of this architecture._

## Scalability considerations

_Identify and address scalability concerns relevant to the architecture in this scenario._

## Availability considerations

_Identify and address availability concerns relevant to the architecture in this scenario._

## Manageability considerations

_Identify and address manageability concerns relevant to the architecture in this scenario._

## Security considerations

_Identify and address security concerns relevant to the architecture in this scenario._

## Deploy the solution

_Describe a step-by-step process for implementing the reference architecture solution. Best practices are to
add the solution to GitHub, provide a link (use boilerplate text below), and explain how to roll out the
solution._

A deployment for a reference architecture that implements these recommendations and considerations is
available on [GitHub](https://www.github.com/path-to-repo).

1. First step
2. Second step
3. Third step ...

## Next steps

Links to articles on the AAC, and other Docs articles. Could also be to appropriate sources outside of Docs,
such as GitHub repos or an official technical blog post.

Examples:
* [Artificial intelligence (AI) - Architectural overview](/azure/architecture/data-guide/big-data/ai-
overview)
* [Choosing a Microsoft cognitive services technology](/azure/architecture/data-guide/technology-
choices/cognitive-services)
* [What are Azure Cognitive Services?](/azure/cognitive-services/what-are-cognitive-services)

## Related resources

Another optional link list, in bulleted form. If you have several links to other articles in Docs, include
those in "Next steps" and use "Related resources" for related architecture guides and architectures.

Here is an example section:

Fully deployable architectures:

* [Chatbot for hotel reservations](/azure/architecture/example-scenario/ai/commerce-chatbot)
* [Build an enterprise-grade conversational bot](/azure/architecture/reference-
architectures/ai/conversational-bot)
* [Speech-to-text conversion](/azure/architecture/reference-architectures/ai/speech-ai-ingestion)
```

Markdown and YML templates for example workloads

5/21/2021 • 5 minutes to read

Use these Markdown and YML templates to create an example workload, previously called an example scenario or sample solution.

For guidance, see Create an example workload.

Each article must have two files. Follow the instructions for browse format authoring at Authoring content for the Azure Architecture Center Browser.

Name this file *<sample-solution>.yml*:

```
### YamlMime:Architecture
metadata:
  title: <Article title, which becomes the title metadata>
  titleSuffix: Azure Example Scenarios
  description: <Write a 100-160 character description that ends with a period and ideally starts with a call
to action. This becomes the browse card description.>
  author: <contributor's GitHub username. If no GitHub account, use EdPrice-MSFT>
  ms.author: <contributor's Microsoft alias. If no alias, leave out this metadata as it will be
automatically set to pnp by the docfx.json.>
  ms.date: <publish or major update date - mm/dd/yyyy>
  ms.topic: conceptual
  ms.service: architecture-center
  ms.subservice: example-scenario
azureCategories:
  - <choose at least one category from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#azure-categories>
  - <there can be more than one category>
products:
  - <choose 1-5 products from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#products>
  - <1-5 products>
  - <1-5 products>
name: <The H1 title is a noun phrase that describes the scenario. Avoid naming the scenario after the Azure
technologies that are used. Example: "Insurance claim image classification on Azure">
summary: <Write a summary. Can be same as description.>
thumbnailUrl: /azure/architecture/browse/thumbs/<browse-png-filename>.png
content: |
  [!include[](<sample-solution>-content.md)]
```

Name this file *<sample-solution-content>.md*:

```
> The H1 title is a noun phrase that describes the scenario. Don't enter it here, but as the **name** value
in the corresponding YAML file.

Introductory section - no heading

> This should be an introduction of the business problem and why this scenario was built to solve it.
>> What industry is the customer in?
>> What prompted them to solve the problem?
>> What services were used in building out this solution?
>> What does this example scenario show? What are the customer's goals?

> What were the benefits of implementing the solution described below?
```

```
## Potential use cases

> Are there any other use cases or industries where this would be a fit?
> How similar or different are they to what's in this article?

These other uses cases have similar design patterns:

- List of example use cases

## Architecture

_Architecture diagram goes here_

> What does the solution look like at a high level?
> Why did we build the solution this way?
> What will the customer need to bring to this?  (Software, skills, etc?)

Under the diagram, include a numbered list that describes the data flow or workflow.

### Components

A bulleted list of components in the architecture (including all relevant Azure services) with links to the
service pages.

> Why is each component there?
> What does it do and why was it necessary?
> Link the name of the service (via embedded link) to the service's product service page. Be sure to exclude
the localization part of the URL (such as "en-US/").

- Examples:
  - [Azure App Service](https://azure.microsoft.com/services/app-service)
  - [Azure Bot Service](https://azure.microsoft.com/services/bot-service)
  - [Azure Cognitive Services Language Understanding](https://azure.microsoft.com/services/cognitive-
services/language-understanding-intelligent-service)
  - [Azure Cognitive Services Speech Services](https://azure.microsoft.com/services/cognitive-
services/speech-services)
  - [Azure SQL Database](https://azure.microsoft.com/services/sql-database)
  - [Azure Monitor](https://azure.microsoft.com/services/monitor): Application Insights is a feature of
Azure Monitor.
  - [Resource Groups][resource-groups] is a logical container for Azure resources.  We use resource groups
to organize everything related to this project in the Azure console.

### Alternatives

Use this section to talk about alternative Azure services or architectures that you might consider for this
solution. Include the reasons why you might choose these alternatives.

> What alternative technologies were considered and why didn't we use them?

## Considerations

> Are there any lessons learned from running this that would be helpful for new customers?  What went wrong
when building it out?  What went right?
> How do I need to think about managing, maintaining, and monitoring this long term?
> Note that you should have at least two of the H3 sub-sections.

### Availability

> How do I need to think about managing, maintaining, and monitoring this long term?

### Operations

> How do I need to think about operating this solution?

### Performance

> Are there any key performance considerations (past the typical)?

### Scalability
```

```
> Are there any size considerations around this specific solution?
> What scale does this work at?
> At what point do things break or not make sense for this architecture?

### Security

> Are there any security considerations (past the typical) that I should know about this?

### Resiliency

> Are there any key resiliency considerations (past the typical)?

### DevOps

> Are there any key DevOps considerations (past the typical)?

## Deploy this scenario

> (Optional, but greatly encouraged)
>
> Is there an example deployment that can show me this in action?  What would I need to change to run this
in production?

## Pricing

> How much will this cost to run?
> Are there ways I could save cost?
> If it scales linearly, than we should break it down by cost/unit. If it does not, why?
> What are the components that make up the cost?
> How does scale affect the cost?
>
> Link to the pricing calculator with all of the components in the architecture included, even if they're a
$0 or $1 usage.
> If it makes sense, include small/medium/large configurations. Describe what needs to be changed as you
move to larger sizes.

## Next steps

> Where should I go next if I want to start building this?
> Are there any reference architectures that help me build this?
> Be sure to link to the Architecture Center, to related architecture guides and architectures.

- Examples:
  - [Artificial intelligence (AI) - Architectural overview](/azure/architecture/data-guide/big-data/ai-
overview)
  - [Choosing a Microsoft cognitive services technology](/azure/architecture/data-guide/technology-
choices/cognitive-services)
  - [What are Azure Cognitive Services?](/azure/cognitive-services/what-are-cognitive-services)
  - [What is Language Understanding (LUIS)?](/azure/cognitive-services/luis/what-is-luis)
  - [What is the Speech service?](/azure/cognitive-services/speech-service/overview)
  - [What is Azure Active Directory B2C?](/azure/active-directory-b2c/overview)
  - [Introduction to Bot Framework Composer](/composer/introduction)
  - [What is Application Insights](/azure/azure-monitor/app/app-insights-overview)
  - [Chatbot for hotel reservations](/azure/architecture/example-scenario/ai/commerce-chatbot)
  - [Build an enterprise-grade conversational bot](/azure/architecture/reference-
architectures/ai/conversational-bot)
  - [Speech-to-text conversion](/azure/architecture/reference-architectures/ai/speech-ai-ingestion)

## Related resources

> Are there any relevant case studies or customers doing something similar?
> Is there any other documentation that might be useful?
> Are there product documents that go into more detail on specific technologies that are not already linked?

<!-- links -->

[calculator]: https://azure.com/e/
```

Markdown and YML templates for solution ideas

6/24/2021 • 3 minutes to read

Use these Markdown and YML templates to create a solution idea.

For guidance, see Create a solution idea.

Each article must have two files. Follow the instructions for browse format authoring at Authoring content for the Azure Architecture Center Browser.

Call this file *<solution-idea>.yml*:

```
### YamlMime:Architecture
metadata:
  title: <Article title, which becomes the title metadata>
  titleSuffix: Azure Solution Ideas
  description: <Write a 100-160 character description that ends with a period and ideally starts with a call
to action. This becomes the browse card description.>
  author: <contributor's GitHub username. If no GitHub account, use EdPrice-MSFT>
  ms.author: <contributor's Microsoft alias. If no alias, leave out this metadata as it will be
automatically set to pnp by the docfx.json.>
  ms.date: <publish or major update date - mm/dd/yyyy>
  ms.topic: conceptual
  ms.service: architecture-center
  ms.subservice: solution-idea
azureCategories:
  - <choose at least one category from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#azure-categories>
  - <there can be more than one category>
products:
  - <choose 1-5 products from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#products>
  - <don't use "azure" unless there is nothing else>
  - <1-5 products>
name: <The H1 title is a noun phrase that describes the scenario. Avoid naming the scenario after the Azure
technologies that are used. Example: "Insurance claim image classification on Azure">
summary: <Write a summary. Can be same as description.>
thumbnailUrl: /azure/architecture/browse/thumbs/<browse-png-filename>.png
content: |
  [!include[](<solution-idea>-content.md)]
```

Call this file *<solution-idea-content>.md*:

```
> The H1 title is a noun phrase that describes the scenario. Don't enter it here, but as the **name** value
in the corresponding YML file.>
> Include the solution idea header note at the top of the solution idea. This adds clarification why this is
a scaled-back architecture (and provides consistency with our other SIs)...

[!INCLUDE [header_file](../../../includes/sol-idea-header.md)]

Introductory section (no heading)
The introduction contains:

- A paragraph that describes what the solution does (the domain)
- A paragraph that contains a brief description of the main Azure services that make up the solution. This
paragraph should convey the Azure value proposition, not a complete description of the architecture.

## Potential use cases

> Are there any other use cases or industries where this would be a fit?
```

```markdown
> How similar or different are they to what's in this article?

These other uses cases have similar design patterns:

- List of example use cases

## Architecture

_Architecture diagram goes here_

After the architecture diagram, include a numbered list that describes the data flow or workflow through the
solution. Explain what each step does. Start from the user or external data source, and then follow the flow
through the rest of the solution.

### Components

A bullet list of components in the architecture (including all relevant Azure services) with links to the
product service pages. This is for lead generation (what business, marketing, and PG want). It helps drive
revenue.

> Why is each component there?
> What does it do and why was it necessary?

- Example: [Resource Groups][resource-groups] is a logical container for Azure resources.  We use resource
groups to organize everything related to this project in the Azure console.

## Next steps

Links to articles on the AAC, and other Docs articles. Could also be to appropriate sources outside of Docs,
such as GitHub repos or an official technical blog post.

Examples:
* [Artificial intelligence (AI) - Architectural overview](/azure/architecture/data-guide/big-data/ai-
overview)
* [Choosing a Microsoft cognitive services technology](/azure/architecture/data-guide/technology-
choices/cognitive-services)
* [What are Azure Cognitive Services?](/azure/cognitive-services/what-are-cognitive-services)

## Related resources

Another optional link list, in bulleted form. If you have several links to other articles in Docs, include
those in "Next steps" and use "Related resources" for related architecture guides and architectures.

Here is an example section:

Fully deployable architectures:

* [Chatbot for hotel reservations](/azure/architecture/example-scenario/ai/commerce-chatbot)
* [Build an enterprise-grade conversational bot](/azure/architecture/reference-
architectures/ai/conversational-bot)
* [Speech-to-text conversion](/azure/architecture/reference-architectures/ai/speech-ai-ingestion)![image.]
(https://user-images.githubusercontent.com/13895622/116135227-b73cac00-a685-11eb-92d3-003350ba6604.png)
```

Old markdown templates

6/24/2021 • 4 minutes to read

These are old templates moved from the AAC repo. Do not use them. For more information, see What is the Azure Architecture Center?

```
---
title: <Use the H1 title of your article. Maximum recommended length is 60 characters>
titleSuffix: Azure Reference Architectures
description: <A brief summary of your article. This is the description that appears in search engine
results. Maximum recommended length is 150-160 characters.>
author: <github username>
ms.date: <publish or update date - mm/dd/yyyy>
ms.service: architecture-center
ms.topic: conceptual
ms.subservice: reference-architecture
---

[NOTE: You can find detailed guidelines about creating reference architectures [here](../reference-
architectures.md).]

# Article title

_Brief introduction goes here._ [**Deploy this solution**.](#deploy-the-solution)

![alt text.](./media/folder_name/architecture-diagram.png)

_Download a [Visio file](https://arch-center.azureedge.net/architecture.vsdx) that contains this
architecture diagram. This file must be uploaded to `https://arch-center.azureedge.net/`_

## Architecture

The architecture consists of the following components:

- **Thing 1**. Description

- **Thing 2**. Description

## Recommendations

The following recommendations apply for most scenarios. Follow these recommendations unless you have a
specific requirement that overrides them.

_Include considerations for deploying or configuring the elements of this architecture._

## Scalability considerations

_Identify and address scalability concerns relevant to the architecture in this scenario._

## Availability considerations

_Identify and address availability concerns relevant to the architecture in this scenario._

## Manageability considerations

_Identify and address manageability concerns relevant to the architecture in this scenario._

## Security considerations

_Identify and address security concerns relevant to the architecture in this scenario._
```

```
## Deploy the solution

_Describe a step-by-step process for implementing the reference architecture solution. Best practices are to add the solution to GitHub, provide a link (use boilerplate text below), and explain how to roll out the solution._

A deployment for a reference architecture that implements these recommendations and considerations is available on [GitHub](https://www.github.com/path-to-repo).

1. First step
2. Second step
3. Third step ...
```

```
---
title: <Article title>
titleSuffix: Azure Example Scenarios
description: <Write a 100-160 character description that ends with a period and ideally starts with a call to action.>
author: <github username>
ms.date: <publish or update date - mm/dd/yyyy>
ms.service: architecture-center
ms.topic: conceptual
ms.subservice: example-scenario
---

# Article title

The title is a noun phrase that describes the scenario.

> Example: "Insurance claim image classification on Azure"

Avoid naming the scenario after the Azure technologies that are used.

(Introductory section - no heading)

> This should be an introduction of the business problem and why this scenario was built to solve it.
>> What industry is the customer in?
>> What prompted them to solve the problem?
>> What services were used in building out this solution?
>> What does this example scenario show? What are the customer's goals?

> What were the benefits of implementing the solution described below?

## Potential use cases

> Are there any other use cases or industries where this would be a fit?
> How similar or different are they to what's in this article?

These other uses cases have similar design patterns:

- List of example use cases

## Architecture

_Architecture diagram goes here_

> What does the solution look like at a high level?
> Why did we build the solution this way?
> What will the customer need to bring to this? (Software, skills, etc?)
> Is there a data flow that should be described?

### Components

A bullet list of components in the architecture (including all relevant Azure services) with links to the product documentation.

> Why is each component there?
```

```
> What does it do and why was it necessary?

- Example: [Resource Groups][resource-groups] is a logical container for Azure resources. We use resource groups to organize everything related to this project in the Azure console.

### Alternatives

Use this section to talk about alternative Azure services or architectures that you might consider for this solution. Include the reasons why you might choose these alternatives.

> What alternative technologies were considered and why didn't we use them?

## Considerations

> Are there any lessons learned from running this that would be helpful for new customers?  What went wrong when building it out?  What went right?
> How do I need to think about managing, maintaining, and monitoring this long term?

### Availability

### Scalability

> Are there any size considerations around this specific solution?
> What scale does this work at?
> At what point do things break or not make sense for this architecture?

### Security

> Are there any security considerations (past the typical) that I should know about this?

## Deploy this scenario

> (Optional, but greatly encouraged)
>
> Is there an example deployment that can show me this in action? What would I need to change to run this in production?

## Pricing

> How much will this cost to run?
> Are there ways I could save cost?
> If it scales linearly, then we should break it down by cost/unit. If it does not, why?
> What are the components that make up the cost?
> How does scale affect the cost?
>
> Link to the pricing calculator with all of the components in the architecture included, even if they're a $0 or $1 usage.
> If it makes sense, include small/medium/large configurations. Describe what needs to be changed as you move to larger sizes

## Next steps

> Where should I go next if I want to start building this?
> Are there any reference architectures that help me build this?

## Related resources

> Are there any relevant case studies or customers doing something similar?
> Is there any other documentation that might be useful?
> Are there product documents that go into more detail on specific technologies not already linked

<!-- links -->

[calculator]: https://azure.com/e/
```

Cloud Adoption Framework content review process

7/12/2021 • 2 minutes to read

Prerequisite requirements

Follow these steps to prepare your environment:

1. If you don't already have a GitHub account, create one.
2. Link your GitHub account to your Microsoft account (<alias>@microsoft.com):
   a. Go to Microsoft Open Source organizations, and sign in.
   b. Complete the steps to link your accounts.

      If your accounts are already linked, you see **Link OK**.
3. Join the **MicrosoftDocs** organization:
   a. Go to Microsoft Open Source organizations.
   b. In the **Available Microsoft GitHub organizations** section (right-hand side), scroll down until you see **MicrosoftDocs**.
   c. Select **Join**.

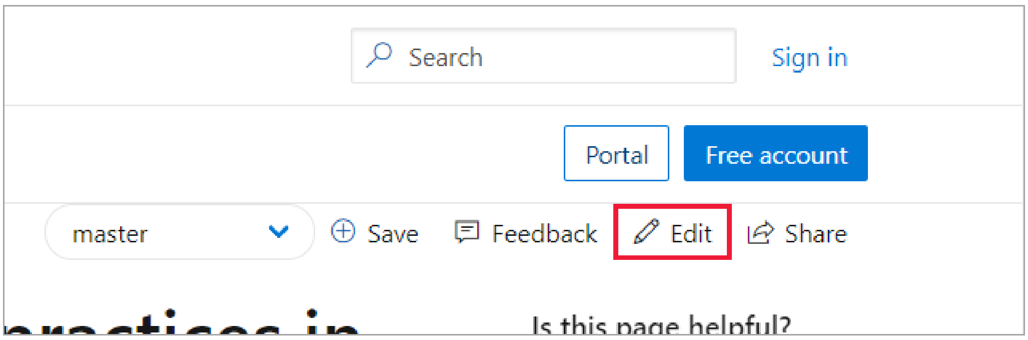
Review content and submit suggested changes that the author can review

Watch this 2-minute video to see how to edit staged content.

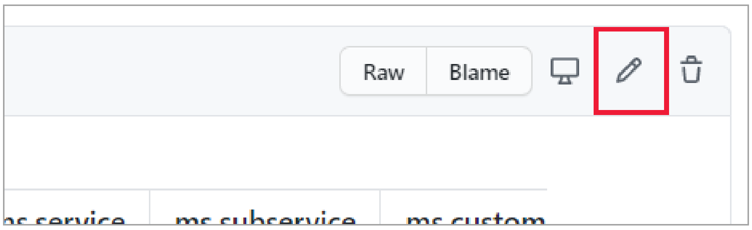
Or follow these steps:

1. If the author provided direct links, go to the links provided.

   If the author didn't provide a direct link, go to the Microsoft Cloud Adoption Framework for Azure docs and find the article.
2. Select **Edit**.

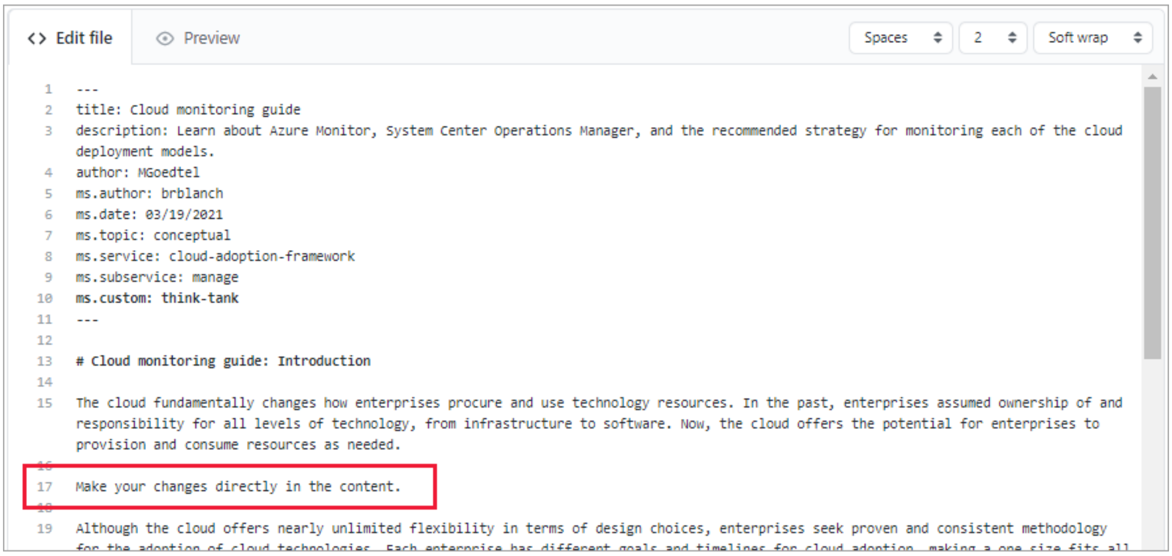
   

   The link takes you to the GitHub source of the article.
3. Select the pencil icon to edit the source content.

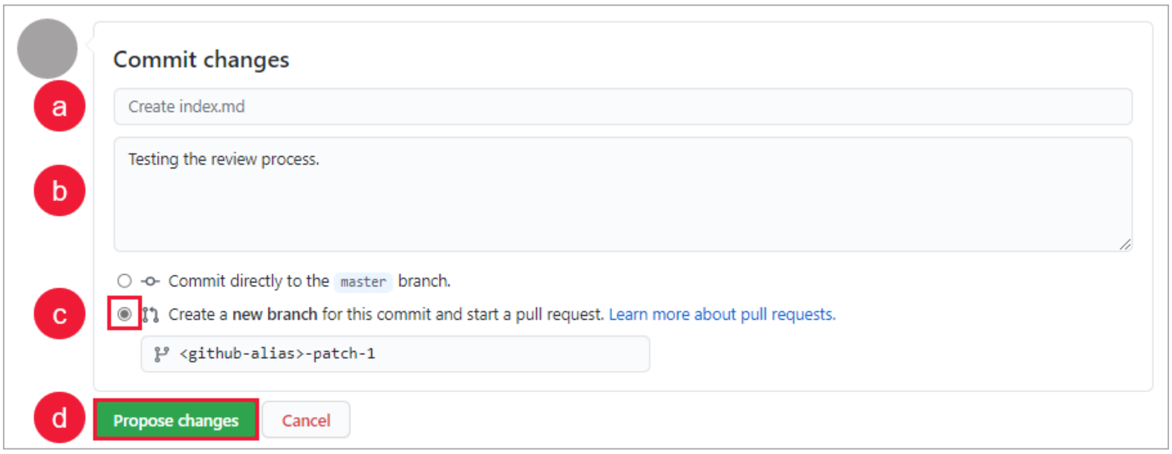


4. Make your changes directly in the content.

   You can add, remove, or change text. But you must make an alteration to the text before you can leave a comment and propose a change.



5. After you make your change:



   a. Give your change request a title.

   b. Leave a comment describing your changes in the text box.

   c. Select **Create a new branch** ... so you can propose your new changes.

   d. Select **Propose changes** to go to the **Open a pull request page**.

6. On the **Open a pull request** page, scroll down to review the changes that you have made.

7. If you're happy with your changes, select **Create pull request**.

Watch for GitHub notifications. The author may @mention you in the pull request to have a conversation about the changes.

It's OK to propose changes even if you aren't 100% sure of the change. Your pull request is the start of a conversation.

See also

- Adding line comments to a pull request.
- Collaboration with Git and GitHub.

Troubleshooting

If you select a **Edit** in the review article and you get a 404 message, we recommend you check two things:

- Are you actually signed into GitHub with the account you linked to your Microsoft identity?
- Are you a member of the MicrosoftDocs organization in GitHub? Visit the Open Source portal to check your memberships.

Work tracking for Azure DevOps content

7/7/2021 • 2 minutes to read

The Azure DevOps content team plans and tracks their work in Azure Boards on the following backlogs and boards:

- **Azure DevOps Features backlog**: `https://dev.azure.com/mseng/TechnicalContent/_backlogs/backlog/Azure%20DevOps/Features?showParents=true`
- **Azure DevOps User Stories backlog**: `https://dev.azure.com/mseng/TechnicalContent/_backlogs/backlog/Azure%20DevOps/Stories?showParents=true)`
- **Azure DevOps User Stories board**: `https://dev.azure.com/mseng/TechnicalContent/_boards/board/t/Azure%20DevOps/Stories`

We use the backlogs to track significant work for documenting new scenarios and tasks. We also use the backlogs to track changes to the UI, and to review and update content to support accessibility, SEO, and general content freshness.

> **TIP**
> If you are adding new content or making significant changes, please include a member of the content team on this work to avoid duplication and to align with ongoing content strategies. You can find the member of the content team who oversees a specific content area in Azure DevOps repositories, content areas, content owners. Or email the Azure DevOps content team.

Add a user story or doc defect bug

For new content to document a new scenario or address a feature or UI change, use the following template:

- Azure DevOps User story

To address an inaccuracy or doc defect in the content, use the following template:

- Azure DevOps Content Bug

Some specific fields to provide:

- Area path: `TechnicalContent\Dev Tools and Services\Jill Reinauer Team\Azure DevOps`
- Iteration: `TechnicalContent\3WeekSprints\Future`
- Description: Explain the issue or requirement. Provide a link when possible. If there's a due date, explain why.

Other resources

- Azure Boards (work tracking) documentation

Azure DevOps content review checklist

7/7/2021 • 3 minutes to read

Contributors and content owners can use this checklist when writing or reviewing content contributed to Azure DevOps repositories.

Other resources include:

- Writing checklist
- Microsoft Cloud Style Guide
- Term Studio
- Accessibility guidelines

Structure

- [ ] Is the content appropriate for a technical article, as opposed to a blog post or white paper?
- [ ] Does the introduction make clear what the article is about?
- [ ] Is it clear who the target audience is?
- [ ] Is the goal of the article clear? How does it help customers? What problem does it solve?
- [ ] Is there any unnecessary content that doesn't support the intent or help solve the problem?
- [ ] Does the article follow the standards for its content type, is it the right content type?
- [ ] Does the flow of the article logical?
- [ ] Does the article file name meet our conventions? Is it located in the right folder?

Metadata

- [ ] All required metadata filled out.
- [ ] Review metadata for SEO.
- [ ] Update the `ms.date` for:

- New articles.
- Articles that you've updated in a significant way.

- [ ] `monikerRange` check.

Text

- [ ] Review of staged content for any poorly formatted text.
- [ ] You provided Header/Platform/Version.
- [ ] Headers - No gerunds, sentence capitalization, short to cause least amount of **In this article** wrapping.
- [ ] Check all headers for sentence capitalization.

- [ ] Check all headers for use of active verbs in Quickstarts, Tutorials, and How-to guides.

- [ ] Use of **Prerequisites** section where needed.

- [ ] Modern voice

- Are the voice and tone of the text following modern voice guidelines?
- Is the content concise, conversational, and engaging--but not colloquial?

- [ ] A Tutorial has a checklist.

- [ ] You used Fictitious names as needed.

Terms and acronyms

- [ ] Check Term studio as needed around common use terms.

- [ ] Check suspect terms against the Microsoft Cloud Style Guide.

- [ ] Terminology check: are terms used and spelled correctly?

- [ ] Are technical terms and concepts introduced and explained?

- [ ] Check that you've spelled out acronyms on first usage.

SEO

- [ ] See Guidance for using the SEO cheat sheet.

- [ ] Do these parts of the article reflect the words the customer will use when searching for this content:

- Metadata title
- Metadata description
- Page title (H1)
- Summary (first paragraph)

- [ ] Do those parts of the article clearly capture the relevance of the article?

- [ ] Are the expected search keywords and phrases used in the content?

Art and images

- [ ] See Create and format screenshots in documentation.

- [ ] Review staged content for any poorly formatted images.

- [ ] Check that screenshot size is within height (> 750 pixels) and width (> 725 pixels) limits.

- [ ] You've provided well-formed Alt text for screenshots.

- [ ] Call outs in screenshots use established color and format.

- [ ] All screenshots have a gray border.

- [ ] Screenshots use fictitious names and don't display sensitive information.

Tags: Moniker Tags and Conceptual

- [ ] Have you applied moniker tags correctly?

- [ ] Review staged content for each version to check format and that no moniker labels appear in the text.

- [ ] Are tabbed conceptual tabs applied correctly using the existing tags defined for Azure DevOps?

Cross linking

- [ ] Are there helpful links to related resources?
- [ ] Quickstarts and tutorials should contain **Next steps** section.
- [ ] Do all external links resolve correctly? Do any need update?
- [ ] Are internal links formatted correctly?
- [ ] Should you create links from anywhere else or other product docsets to this article?
- [ ] Disambiguation – Is there a need to refer the user to some other article to disambiguate terms? For example Work Item Tags versus Git Tags, Azure DevOps Projects and Azure DevOps Project.

TOC

- [ ] Article added to the main TOC YAML file.
- [ ] Article added to secondary TOC YAML files where appropriate with shared TOC tagging.
- [ ] Review the TOC entires check that you didn't use gerunds.
- [ ] TOC entry makes sense even if shortened from the article title.
- [ ] Filtering TOC yields results when using article-specific terms.

Aliases for Azure DevOps technical content

7/7/2021 • 2 minutes to read

Azure DevOps contributors to technical documentation need to belong to certain aliases and be aware of others for information sharing.

ALIAS NAME	ALIAS	REQUIRED?	PURPOSE
Content & Learning Azure DevOps Content Devs	AzDevContentDev	For FTE content developers	FTE communications for content developers (CPub discipline) who oversee major content areas and docsets for Azure DevOps
Azure DevOps Content Contributors	vsocontc		For PMs and devs who contribute to Azure DevOps content
Azure DevOps	azuredevops		Email notifications about Azure DevOps

Azure DevOps `azure-devops-docs-pr` repo and branch policies

The following groups are added to the branch policies for one or more folders in the **azure-devops-docs-pr** repository. By joining a group, you can approve your PRs so that they're merged into the main branch. You join these groups by going to IDWEB.

SECURITY GROUP	WEB ID ALIAS	OWNER	PURPOSE
Azure Boards Content Approvers	ts-work-cnt-approve	kaelli	For content contributors to **Azure Boards** and the `/boards` , `/organizations/settings/work` , and `/reference` folders
Azure DevOps Artifacts Approvers	az-artifacts-approve	phwilson	For content contributors to the `/artifacts` folder
Azure DevOps Collaborate Content Approvers	ts-collbrt-cnt-appro	kaelli	For content contributors to the `/project` and `/notifications` folders
Azure DevOps Release Notes	AzDevRelNotesApprove	alexn	For content contributors to the `/release-notes` folder
Azure DevOps Report Content Approvers	ts-repor-cnt-approve	kaelli	For content contributors to the `/report` folder

SECURITY GROUP	WEB ID ALIAS	OWNER	PURPOSE
Azure DevOps Setup-admin Approvers	vsts-setup-approvers	kaelli	For content contributors to the `/organizations` folder
Team Services CI/CD Content Approvers	ts-cicd-doc-approve	Julia.KullaMader	For all contributors to **Azure Pipelines** content and the `/pipelines` and `/deploy-azure` folders
Team Services Integrate Approvers	ts-integrate-approve	chcomley	For all contributors to content to the following folders: `/integrate`, `/extend`, `/service-hooks`, and `/marketplace`
[TechnicalContent]/Azure DevOps Test Content Approvers	N/A	sdanie	For content contributors to the `/test` folder
DevOps Doc Approvers	vsts-vc-docapprovers	sdanie	For content contributors to the `/repos` folder

These folders are missing from Branch policies: `/demo-gen` and `/java`.

Azure DevOps `azure-devops-server-docs-pr` repo and branch policies

The following groups are added to the Branch Policies for one or more folders in the `azure-devops-server-docs-pr` repository. By joining a group, you can approve your PRs so that they're merged into the main branch.

SECURITY GROUP	WEB ID ALIAS	OWNER	PURPOSE
Azure DevOps TFS Content Approvers	tfs-setup-approvers	elbatk	For content contributors to the `tfs-docs-pr` repository

Azure DevOps repositories, content areas, content owners

11/2/2020 • 2 minutes to read

This article provides specific guidance for Microsoft employees to contribute to Azure DevOps documentation accessible from here:

Git and GitHub repositories

DOCUMENTATION	REPOSITORY (PRIVATE)	GITHUB REPOSITORY (PUBLIC)
Azure DevOps Documentation	azure-devops-docs-pr	MicrosoftDocs/vsts-docs
Azure DevOps Server Documentation	azure-devops-server-docs-pr	Not set up
DevOps Resource Center	devops-learn-pr	Not set up
Azure DevOps Project Documentation	MicrosoftDocs/azure-docs-pr/articles/devops-project	MicrosoftDocs/azure-docs/articles/devops-project/
Azure DevOps CLI	MicrosoftDocs/vsts-docs-cli-python	Not applicable
REST APIS	TBD (Owned by the Product team)	Not applicable

Main product docsets

DOCSET	FOLDER	CONTENT OWNER
New User Guide	/user-guide	Chrystal Comley (chcomley);
Azure Artifacts (formerly Package Management	/artifacts	Rami Bououni (ramiMSFT)
Azure Boards **Reporting & Analytics**	/report	Kathryn Elliott (kaelli)
Azure Repos	/repos	Tier 3 support
Azure Pipelines	/pipelines	Steve Danielson (sdanie) Julia Kulla-Mader (Julia.KullaMader)
Azure Test Plans	/test	Tier 3 support

Analytics and social tools docsets

DOCSET	FOLDER	CONTENT OWNER
Reporting	/report	Kathryn Elliott (kaelli)
Notifications	/notifications	Chrystal Comley (chcomley)
Wikis	/project/wiki	Chrystal Comley (chcomley)
Search	/project/search	Kathryn Elliott (kaelli)
Feedback	/project/feedback	Kathryn Elliott (kaelli)

Settings and administration docset

DOCSET	FOLDER	CONTENT OWNER
Manage organizations	/organizations/accounts/	Chrystal Comley (chcomley)
Billing	/organizations/billing	Chrystal Comley (chcomley)
Manage projects	/organizations/projects/	Chrystal Comley (chcomley)
Security, Access, & Billing	/organizations/security	Kathryn Elliott (kaelli)
All Settings	/organizations/settings	Chrystal Comley (chcomley)
Scale and configure teams	/organizations/settings/scale/	Kathryn Elliott (kaelli)
Work Item Customization	/organizations/settings/work /reference	Kathryn Elliott (kaelli)
Azure DevOps Server Administration	azure-devops-server-docs-pr repo	Kathryn Elliott (kaelli)
Data Warehouse & SQL Reports	/report/sql-reports	Kathryn Elliott (kaelli)

Developer docsets

DOCSET	FOLDER	CONTENT OWNER
Custom Extensions	/extend	Chrystal Comley (chcomley)
Marketplace	/marketplace	Chrystal Comley (chcomley)
Service Hooks	/service-hooks	Chrystal Comley (chcomley)
REST APIs	TBD	TBD

Deprecated resources

Azure DevOps Archive documentation

DOCSET	FOLDER	CONTENT OWNER
SharePoint Integration	/report/sharepoint-dashboards	Kathryn Elliott (kaelli)
TFS-Project Server integration	/reference/tfs-ps-sync	Kathryn Elliott (kaelli)

Azure DevOps metadata guidance

7/7/2021 • 4 minutes to read

We invest a lot to create content for users, decision-makers, influencers, and potential customers. Metadata multiplies that investment by:

- Helping search engines display our content in search results to help our customers.
- Guiding customers to the information they seek, so they can finish tasks with our services and products.
- Enabling us to track and measure how well our content performs and what needs improvement.

General guidance provided here:

- Metadata overview
- Metadata attributes

Sample Azure DevOps metadata

All articles include metadata at the top. You delimit metadata with three dashes `---` as shown in the following example:

```
---
title: Add built-in charts to a team dashboard
titleSuffix: Azure DevOps & TFS
description: Add system-generated charts or query-based charts to a team dashboard for Azure DevOps or Team Foundation Server
ms.prod: devops
ms.technology: devops-analytics
ms.reviewer: greggboe
ms.author: kaelli
author: KathrynEE
ms.topic: quickstart
monikerRange: '>= tfs-2013'
ms.date: 09/20/2018
---
```

For an `/includes` file, provide this metadata:

```
---
ms.topic: include
---
```

Required tags

TAG	USAGE	EXAMPLE
`title`	SEO focused title, 60 total char max, including spaces Docs Build adds the site identifier, Microsoft Docs, to the end of each title.	`Add built-in charts to a team dashboard.`

TAG	USAGE	EXAMPLE
`description`	SEO focused description. It's a minimum 115 characters long and a maximum 145 characters, including spaces. Use important keywords in the first 100 characters.	`Add system-generated charts or query-based charts to a team dashboard.`
`ms.prod`	Enter the value associated with the product.	`devops`
`ms.technology`	Enter a value associated with the content area.	See **ms.technology** list below.
`ms.topic`	Enter a value that represents the article type.	See **ms.topic** list below.
`manager`	Microsoft alias of Content team manager. `ms.manager` works as well.	`douge`
`ms.author`	Microsoft alias of the primary writer of the content area.	`alewis`
`author`	GitHub alias of the primary writer of the content area.	`alewis`
`ms.date`	Date of first publish or with a significant update or change in content. Follow the date pattern shown in the example. Enter the `ms.date` value without time. Docs Build interprets the time as 0:00 in the UTC time zone. Docs converts the date so it's displayed to users in their own time zone.	`09/20/2018`

For more SEO guidance on titles and description, see the Guidance for using the SEO cheat sheet.

Optional tags

TAG	USAGE	EXAMPLE

`titleSuffix`	Use to specify another site identifier associated with your product brand. Product name that will appear in the browser tab followed by **| Microsoft Docs**.	`Azure DevOps`
		or
		`Azure DevOps Services`

`ms.reviewer`	Enter the alias of the primary PM or Dev who reviews or helps author the content.	`greggboe`
`ms.assetid`	GUID, not required.	`B080CEFA-4D94-44B2-99E3-0E3E85616D04`

TAG	USAGE	EXAMPLE
`ROBOTS`	Crawl instruction for the page for web crawlers. The default value is `INDEX, FOLLOW`. All possible values are: • `INDEX, FOLLOW` • `NOINDEX, NOFOLLOW` • `INDEX, NOFOLLOW` • `NOINDEX, FOLLOW`	
`ms.custom`	Text field. You're allowed to add multiple values separated by commas. (**NOTE**: Don't change any `devx-track-*` tags. They're used by the Developer Experience team to track groups of articles across services and repositories. Contact B. Kess with any questions.)	
`ms.devlang`	You can find a list of values in the Taxonomies for Docs article.	`csharp, javascript`
`monikerRange`	Specifies the versions that apply to the article. For the home docset, the article will appear in the TOC only for the versions specified in the range.	`monikerRange: 'vsts'`

ms.technology (for ms.prod: devops)

MS.TECHNOLOGY	TECHNOLOGY NAME STRINGS
`devops-analytics`	Analytics
`devops-accounts`	Accounts
`devops-agile`	Agile
`devops-artifacts`	Artifacts
`devops-billing`	Billing
`devops-code-git`	Repos (Git)
`devops-code-tfvc`	Repos (TFVC)
`devops-collab`	Search and Collaboration, Wiki
`devops-cicd`	Pipelines
`devops-ecosystem`	Extensibility and Integration

MS.TECHNOLOGY	TECHNOLOGY NAME STRINGS
`devops-learn`	Azure DevOps Learn
`devops-marketplace`	Marketplace
`devops-new-user`	New User Guide and Navigation
`devops-public-projects`	Public Projects (pending)
`devops-ref`	Azure DevOps API Reference
`devops-security`	Security and Permissions
`devops-settings`	Settings
`devops-test`	Test
`devops-whitepapers`	DevOps Whitepapers

ms.topic values

MS.TOPIC VALUE	USAGE
`article`	Useful content that isn't specific to a product scenario, like migration or data security. For example, whitepapers like Azure DevOps Technical Articles.
`conceptual`	Any content linked in a TOC that isn't marked in metadata. For example: • `Quickstart` • `Tutorial` • `Sample` • `Reference` • `Overview`
`contributor-guide`	Any content authored specifically for inclusion in a content contributor guide.
`include`	Use in **[!INCLUDE]** files only. It's required so Docs Build, validation, and other tools to work correctly.
`interactive-tutorial`	Interactive content.
`overview`	Overview or User Guide articles. Typically live under an **Overview** node in a TOC, or high-level general article in a New User Guide or Onboarding guide.
`quickstart`	Any authored content placed under a **Quickstart** node in a TOC *and* that follows the specific guidelines and template for a Quickstart article.

MS.TOPIC VALUE	USAGE
`reference`	Any authored (not autogenerated) content placed under a **Reference** node in a TOC.
`sample`	Any authored (not autogenerated) content placed under a **Samples** or **Examples** node in a TOC.
`troubleshooting`	Any authored content placed under a **Troubleshooting** node in a TOC AND that follows the specific guidelines and template for a Troubleshooting article.
`tutorial`	Any authored content placed under a **Tutorials** node in a TOC *and* that follows the specific guidelines and template for a Tutorial article.

Azure DevOps Open Publishing workflow

7/7/2021 • 2 minutes to read

Unlike most Microsoft Docs content in the `MicrosoftDocs` GitHub repo, we maintain Azure DevOps content in several Git repos in Azure Repos. For a list of repos, see Azure DevOps repositories, content areas, content owners.

Workflow to author and publish content

> **TIP**
> If you're adding new content or making significant changes, please include a member of the content team on this work to avoid duplication and to align with ongoing content strategies. You can find the member of the content team who oversees a specific content area in Azure DevOps repositories, content areas, content owners. Or email the Azure DevOps content team.

The following steps describe the current process for authoring and publishing content:

1. Join your GitHub account to the `Microsoft` and `MicrosoftDocs` organizations.
2. Clone the azure-devops-docs-pr in Visual Studio or Visual Studio Code.
3. Create a new local branch from the `/master` branch. You can create the branch from Visual Studio or Visual Studio Code. The new working branch is for your content additions and updates.
4. Make your changes to the content and commit them to this working branch.
   a. If you're working from a cloned repo, push the changes to the remote repo for your working branch.
   b. Create a pull request (PR) from your branch to the master branch.
   c. Once you've committed your content to the remote repo, a build kicks off. Builds can take 10-15 min. If you're not sure if you're in the build queue, you can check the build queue portal. Log in to the build portal with your GitHub alias to reach this page. You'll receive an email that tells you where the content is staged and if there are broken links. A link to the build report is in the email.
   d. Check that your content displays correctly on the staging site.
   e. Once your content displays correctly with no build errors, it's ready for review. Create a comment with the text "#sign-off" to trigger the PR Merger review.
   f. PR Merger approves some PRs automatically and requests reviews for others. After the PR is approved, PR Merger merges the PR.
5. The team creates PRs at least twice a day. These PRs pull the content changes from `master` into `live`. When the team merges the PRs, your changes are published.

Contributor's guide resources

- **Follow content standards**
  - Write an overview
  - Write a quickstart
  - Write a tutorial
  - Write a sample
  - Edit a TOC

- Improve your writing
  - Follow writing principles
  - Review the style guidelines

Staged content

Whenever you push your working branch to the cloud repo, the branch builds. You'll get an email when the build completes along with a link to an error report. The email also includes a link to the staged content that looks similar to the following example:

```
https://review.docs.microsoft.comazure/devops/File-path-and-name?branch=branchname
```

When you push your branch, you can see where your build is in the OPS Portal build queue. You'll receive emails when the build completes after you create a PR or push your changes.

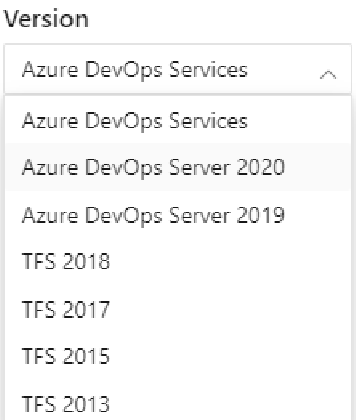
Each content area has its own table of contents (TOC) that uses markdown syntax within a yml file (`toc.yml`). The TOC on the published site only shows one area's TOC at a time. To navigate across each area in the hub, use the breadcrumb navigation in the top navigation bar.

Azure DevOps versioning guidance and scenarios

6/24/2021 • 4 minutes to read

Many Azure DevOps articles require the application of moniker tags to support versioning. Users can select between Azure DevOps Services and the version of TFS (will become Azure DevOps Server) they are using. The content displayed will only show that content which applies to their selected version, based on the monikers defined within the article.

Users select the Azure DevOps Services or Azure DevOps/TFS version from the left-hand TOC to scope the content that is displayed. The **azure-devops-docs-pr** repository contains content for the cloud product, Azure DevOps Services, and the on-premises product, Azure DevOps Server (previously named Team Foundation Server). Versions for TFS 2013 and all later versions are supported.



As much as possible, we want to minimize topic versioning. That is, we want each article to address whatever platform/versions that support the task or scenario covered by the content.

Features and scenarios introduced during an update, such as TFS 2017.2, are handled through notes. These notes are encapsulated within monikers specific to the main version.

As much as possible, we want to maintain the same article (with URL juice) and allow it to cover features introduced first on Team Services and then later into a TFS Update.

- Articles should be scenario and task-oriented, features are secondary
- Only when the tasks diverge fully (or lets say 60% or more) between platforms or versions should you consider branching an article to address the differences.

Azure DevOps monikers

Azure DevOps and product versions are associated with monikers. The following table indicates the current monikers that you can use in Azure DevOps articles to control versioning:

MONIKER	PRODUCT/VERSION
azure-devops	Azure DevOps product version
azure-devops-2020	Azure DevOps Server 2020
azure-devops-2019	Azure DevOps Server 2019

MONIKER	PRODUCT/VERSION
tfs-2018	Team Foundation Server 2018
tfs-2017	Team Foundation Server 2017
tfs-2015	Team Foundation Server 2015
tfs-2013	Team Foundation Server 2013

Versioning is implemented at three levels: Zone, Article, and Folder.

Only major level versioning is supported. If new features are added to a version update, for example TFS 2017.2, then you need to make that explicit in the content.

Zones within articles

Inside an article, you can version specific **zones** of text. Consider the following taken from the Add and manage dashboards. Zone monikers uses the same moniker range concepts, but they apply to zones inside the article.

```
::: moniker range="tfs-2015"
> [!NOTE]
> Multiple team dashboards and the [widget catalog](widget-catalog.md) are available from TFS 2015.1 or
later versions. For TFS 2015 and earlier versions, you don't have access to multiple team dashboards.
Instead, your home page serves as a [single team dashboard](team-dashboard.md). For information on
SharePoint dashboards, see [Project portal dashboards](../sharepoint-dashboards/project-portal-
dashboards.md).
::: moniker-end

::: moniker range="tfs-2017"
> [!NOTE]
> For information on SharePoint dashboards, see [Project portal dashboards](../sharepoint-
dashboards/project-portal-dashboards.md).
::: moniker-end


::: moniker range="azure-devops"
[!INCLUDE [temp](../includes/dashboard-prerequisites-vsts.md)]
::: moniker-end

::: moniker range=">= tfs-2017 <= tfs-2018"
[!INCLUDE [temp](../includes/dashboard-prerequisites-tfs-2017-18.md)]
::: moniker-end

::: moniker range="tfs-2015"
[!INCLUDE [temp](../includes/dashboard-prerequisites-tfs-2015.md)]
::: moniker-end
```

Article level metadata

Article-level versioning is applied when the article applies to select versions. You add a `monikerRanger` metadata value to the article. This overrides the folder-level defaults. Consider the following portion of metadata:

```
ms.topic: quickstart
monikerRange: '>= tfs-2015'
ms.date: 07/21/2018
---
```

In this example, the article applies to all versions of TFS 2015 and greater (`>= tfs-2015`) including Azure DevOps Services

In the following example, the article applies to only versions TFS 2013 and TFS 2015.

```
monikerRange: '>= tfs-2013 <= tfs-2015'
ms.date: 08/11/2016
---
```

Folder level defaults

Each folder in the repository has a versioning default assigned in the **vsts-docs-pr/docs/docfx.json** file in the repository. This means that new articles are automatically versioned just by their location.

Folder-level monikers have been applied to the following folders:

```
    "fileMetadata": {
      "monikerRange": {
        "report/admin/**": "< azure-devops",
        "report/extend-analytics/**": ">= azure-devops-2019",
        "report/powerbi/**": ">= azure-devops-2019",
        "report/sql-reports/**": "< azure-devops",
        "reference/witadmin/**": "< azure-devops",
        "reference/mapping/**": "<= tfs-2018"
```

Platform version header

On many articles, you'll see a platform-version header right below the article title. As the version selector is often ignored by our customers, we want to give them another visual cue as to which platforms and versions an article applies to. Add this type of header to your article right below the title.

You should use the INCLUDE files located under /docs/includes folder or in the /includes folder for your specific vertical.

For articles that apply only to Azure DevOps Services:
Azure DevOps Services
`\docs\includes\version-vsts-only.md`

For articles that apply only to Azure DevOps Services and Azure DevOps Server 2020:
Azure DevOps Services | Azure DevOps Server 2020
`\docs\includes\version-cloud-plus-2020.md`

For articles that span all product versions:
Azure DevOps Services | Azure DevOps Server 2020 | Azure DevOps Server 2019 | TFS 2018 - TFS 2013
`\docs\includes\version-vsts-tfs-all-versions.md`

For articles that are specific to select versions of TFS:
TFS 2017 | TFS 2015 | TFS 2013 `\docs\includes\version-tfs-2013-2017.md`

For articles that are specific to a product/service, such as Azure Boards:

Azure Boards | Azure DevOps Server 2020 | Azure DevOps Server 2019 | TFS 2018 - TFS 2013

`\docs\boards\includes\version-vsts-tfs-all-versions.md`

General guidance for applying monikers

- Do apply article monikers where needed
- Do apply zone monikers where needed
- Don't nest monikers
- Don't apply monikers around conceptual tabs

When a new feature is introduced, it is usually on the cloud version first, so you apply monikers accordingly.

When a new version of Azure DevOps Server is released, you need to review the moniker applications and update accordingly.

Related articles

- Conceptual Versioning
- SQL versioning guidance and scenarios for writers

Azure DevOps tabbed conceptual tags usage and guidance

7/7/2021 • 2 minutes to read

Use tabbed conceptual tags to provide different versions of content depending on selections made by the user. That selection might cover:

- Different features supported by the product, such as new navigation and previous navigation
- Git repo versus TFVC repo
- Browser view versus a client view, such as Visual Studio or Team Explorer Everywhere
- OS differences
- New work item form versus old form

> **NOTE**
>
> Tabbed conceptual tags differ from versioning moniker tags which are used to support differences in content among different product versions.
>
> The zone pivot feature is often a competitive alternative to the tabbed conceptual feature.

Syntax for usage

The syntax for using tabbed conceptual tags is simple, as shown in the following example. This syntax was updated to support markdig build engine.

```
#### [Browser](#tab/browser)

<!--- Content here  -->

#### [Visual Studio](#tab/visual-studio)

<!--- Content here  -->

---
```

The last three dashes close out the tabbed conceptual entry.

Examples used in articles

- Uses **Browser** and **Visual Studio**:
  - Use work item templates, Markdown file
- Uses **New form**, **Old form**, **Visual Studio**, and **Team Explorer Everywhere**:
  - Link work items to support traceability and manage dependencies

Guidance on using tabbed conceptual tags

- If you use tabbed conceptual tags, you need to use the same set throughout the article. You can't use different sets of tags in the same article. Likewise, you can't use two tags in one place and three tags in another place within the same article.

Guidance on using Tabbed Conceptual tags and Conceptual Versioning Moniker tags

- Can use moniker tags within a tabbed conceptual tag, but not the other way around. You can't surround a tabbed conceptual set of tags with monikers.

Tags to use

Here are the tags that are in use. Use these tags whenever possible and be sure to use the same tags everywhere.

If you have to add new tags, add them to this list.

LABEL	TAG	NOTES, ARTICLE WHERE USED
New web form	new-web-form	/boards content
Old web form	old-web-form	/boards content
Designer	designer	/pipelines/build content
YAML	yaml	/pipelines content
Web	web	/pipelines/agents content
Windows	windows	/pipelines/agents content
Linux	linux	/pipelines/agents content
macOS	unix	/pipelines/agents content
Azure DevOps CLI	azure-devops-cli	/organizations/settings/set-area-path
Azure Repos or TFS repo	vsts	/pipelines/apps content
Azure Repos	gitvsts	/pipelines/apps content
GitHub repo	github	/pipelines/apps content
Install them during the build	apple-install-during-build	\pipelines
Preinstall them on a macOS build agent	apple-preinstall	\pipelines
Team Explorer Everywhere	tee	Used in a few /boards content
Visual Studio	visual-studio	/boards and /repos content
Command Line	command-line	/repos content
MSTest	mstest	/pipelines content
NUnit	nunit	/pipelines content

LABEL	TAG	NOTES, ARTICLE WHERE USED
Batch	batch	/pipelines content
Bash	bash	/pipelines content
PowerShell	powershell	/pipelines content
Shell	shell	/pipelines content
Browser	browser	Use to support the web portal view
HTTP	http	/extend content
C# (client library)	csharpclient	/extend content
C# (generic)	csharpgeneric	/extend content
Node.js (generic)	nodejsgeneric	/extend content
Legacy URLs	vsts	DUPLICATE USAGE - /artifacts/quickstarts
New URLs	azuredevops	/artifacts/quickstarts

Tags to not use

LABEL	TAG	NOTES
Web	web	Use **Browser** tab instead
Web Portal	vsts-tfs-web-portal	Use **Browser** tab instead

Azure DevOps terminology and branding guidance

7/7/2021 • 12 minutes to read

This article provides usage examples and guidance for brand names and specific terms used in Azure DevOps content.

Our umbrella name is **Azure DevOps**. For official naming guidance, see Azure DevOps in the cloud style guide.

Generally, though, use the following statements as streamlined guidance:

1. Use the official service names. For example, use *Azure Pipelines* when talking about the product.
2. Use *Azure DevOps* when talking about the collection of services.
3. When required, use *Azure DevOps Services* and *Azure DevOps Server* to disambiguate between the cloud and on-premises offerings. But where possible, use *Azure DevOps* as the umbrella term.

Don't abbreviate the official name (*Azure DevOps*). For example, don't use *ADS*, *ADOS*, and so on. Internally, our shortened team name is *AzDev* but don't use this abbreviation externally or in hashtags on social media apps. *Azure DevOps* or *#AzureDevOps* are the only forms of the name you can use. *DevOps* should always be CamelCase.

You don't need to prefix the name with *Microsoft*. For example, you don't need to say *Microsoft Azure Pipelines*. There are no trademark symbols required in licenses or console output. Always include *Azure* unless you have naming or repetition constraints. Contact Ori Zohar and Martin Woodward if you think you might have a reason to not precede with *Azure*.

Product/service names and usage

IMPORTANT

We need to address both the **New Navigation** elements and the **Previous Navigation** elements (previous names). So, when adding or editing content, think through the names used in the user interface based on the navigation steps you're documenting.

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
Azure DevOps		Umbrella name of Microsoft's DevOps solutions. It includes five cloud services, collectively called Azure DevOps Services and an on-premises product currently called Team Foundation Server (TFS).	*To create an Azure DevOps organization with a personal Microsoft account or a work or school account, go to the [Azure DevOps Services](#) site.*	Limit usage as follows: Azure DevOps organization, Azure DevOps project, Azure DevOps group. Procedural steps documenting a user interface that uses Azure DevOps. Otherwise, **DO NOT USE** in docs until after TFS is rebranded to *Azure DevOps Server* in March 2019 unless you're documenting a user interface with *Azure DevOps* in the user interface string.
Azure DevOps Services	Visual Studio Team Services, VSTS, Team Services, Visual Studio ALM, Visual Studio Team Services (VSTS), Visual Studio Online, VSO	Umbrella suite name for the hosted or cloud offering.	*In this quickstart, you learn how to view your permissions or another person's permissions defined in Azure DevOps Services or Team Foundation Server (TFS).*	Use only to refer to *the collection of 5 services*. When referring to functionality or features of a specific service, use the name of that service. Don't find and replace previous names with Azure DevOps Services; it's important to identify the correct individual service. Use to disambiguate the cloud offering from the on-premises offering. Don't shorten or abbreviate, for example, *DevOps Services*, *ADS*, *ADOS*, and so on.

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
Azure DevOps Server	Team Foundation Server and TFS	Name of the on-premises Azure DevOps product.	**Example 1** You have documentation that references on-premises in general (no specific version). Continue to call on-premises *TFS* until March 2019. **Example 2** After 9/5, you blog about a new feature on the service that will be in the next on-premises version. Say it will be available in Azure DevOps Server. **Example 3** You have an error message that is specific to TFS 2018 and earlier. Always keep this as *TFS* since we won't rebrand these shipped versions of TFS.	We won't be making that change yet in the product or in the docs until we're close to launching Azure DevOps Server 2019. We won't go back and update TFS 2018 or earlier. TFS branding will remain for all versions from TFS 2013 through TFS 2018 (including TFS 2018.3). For things that reference TFS on-premises in general (clients, tools, documentation, blogs, and so on), we should keep them as *TFS* until Azure DevOps Server 2019 ships in March. Until then, the latest on-premises is still TFS 2018, so we should call it *TFS*. For references to the next version of on-premises, you can use *Azure DevOps Server*.
Azure Pipelines	Build & Release hub, build and release hub, Visual Studio Team Services Build Service, Visual Studio Team Services Build & Deployment Service	Name of the service/page that hosts the CI/CD functionality.	*Go to the Azure Pipelines service in the web UI...*	Always use *Azure Pipelines*. Don't shorten to *Pipelines*. If you come across phrasing like *CI/CD with VSTS*, replace with *CI/CD with Azure Pipelines* rather than *CI/CD with Azure DevOps*. The goal is to be Azure DevOps service-specific now whenever possible based on the context of the doc. For more terminology specific to Azure Pipelines, see Azure Pipelines terms.

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
Azure DevOps Project(s)		A specific Azure service released long before the Azure DevOps brand.	*Configure continuous integration (CI) and continuous delivery (CD) for your .NET core or ASP.NET application with Azure DevOps Project.*	First reference: *Azure DevOps Project(s)* Later references: *DevOps Project(s)* Always capitalize *Project*. Singular or plural *Project* depends on context. Only use Azure DevOps Project(s) when referring to the named Azure service. Don't capitalize when referring to the concept formerly known as *team project* in VSTS. When referring to a single, user-created Azure DevOps Projects project, use *project* unqualified by the service name.
Azure Boards	Work	Work management for teams with work item visualization, such as Kanban boards.	*To manage your backlog, go to **Boards**...*	Always use *Azure Boards*. Don't shorten to *Boards*, except when referring to the user interface. If you come across phrasing like *Work item tracking with VSTS*, replace with *Work item tracking with Azure Boards* rather than *Work item tracking with Azure DevOps*. The goal is to be Azure DevOps service-specific now whenever possible based on the context of the doc.

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
Azure Repos	Code	Code Repositories for Git and TFVC.		Always use *Azure Repos*. Don't shorten to *Repos*. If you come across phrasing like *Get a VSTS Git repo*, replace with *Get a Git repo with Azure Repos* rather than *Get an Azure DevOps Git Repo*. The goal is to be Azure DevOps service-specific now whenever possible based on the context of the doc.
Azure Test Plans	Test	Capabilities for manual testing of apps.		Always use *Azure Test Plans*. Don't shorten to *Test Plans*. The goal is to be Azure DevOps service-specific where possible.
Azure Artifacts	Package, Package management, Package extension	Package management and distribution.		Always use *Azure Artifacts*. Don't shorten to *Artifacts*. The goal is to be Azure DevOps service-specific now whenever possible based on the context of the doc.
Team Foundation Version Control (TFVC)				Always capitalize *Version Control*. Don't use *Team Foundation version control*.

UI element terminology

The intent of this table is to capture the current state of terminology and plan for the next few sprints as we transition to the new branding. This transition is to ensure consistency between design, product, docs, and marketing.

TERM	PREVIOUS TERM	DEFINITION AND NOTES ON USAGE	SAMPLE USAGE IN DOCS
breadcrumb		Navigation in global header	Global breadcrumb

TERM	PREVIOUS TERM	DEFINITION AND NOTES ON USAGE	SAMPLE USAGE IN DOCS
flyout		A panel that appears in the right to take further inputs or provide additional information about the selected item.	My Work flyout is available in the upper-right corner of every page. The flyout provides quick access to the items you care about, no matter where you are in Azure DevOps services.
global header		Header present on every page, shows location context and provides access to global controls. Includes homepage, breadcrumb, search, my work, marketplace, user profile.	
homepage		https://dev.azure.com Use lowercase *project* to refer to a project in Azure DevOps. Use *Azure DevOps Project(s)* to refer to the named Azure service. Don't prepend *Azure DevOps* unless you really need to disambiguate. For example, use *project* rather than *Azure DevOps project* or *Azure DevOps Services project*. If you do need to disambiguate, use *Azure DevOps project*.	
lists		Secondary navigation used to search and filter within *pages*.	

TERM	PREVIOUS TERM	DEFINITION AND NOTES ON USAGE	SAMPLE USAGE IN DOCS
organization	account, VSTS account, Team Services account	A shared space where users can collaborate on projects. See https://blogs.msdn.microsoft.com/devops/2018/07/19/adopting-the-word-organization/. Don't prepend *Azure DevOps* unless you really need to disambiguate. For example, use *organization* rather than *Azure DevOps organization* or *Azure DevOps Services project*. If you need to disambiguate, use *Azure DevOps organization*.	This article is about adding a project to an Azure DevOps organization or a Team Foundation Service project collection. If instead you want to create a project in Azure DevOps Projects, see Azure DevOps Projects.
pages	tabs, hubs, hub groups	Sections below services and anything else below.	
pivot	tabs	Navigation between two or more content panes and rely on text headers to label the different sections of content.	
project	Team Project, VSTS project, Team Services project	A project provides a repository for source code. A project also provides a place for a group of people to plan, track progress, and collaborate on building software solutions. Since *project* is such a generic term, make sure that the context is clear. For example, avoid confusion with Visual Studio Project.	This article is about adding a project to an Azure DevOps organization or a Team Foundation Service project collection. If instead you want to create a project in Azure DevOps Projects, see Azure DevOps Projects.
project page		`https://dev.azure.com/your-org/your-project`	
selectors		Dropdown menus provided to select from a favorite, filter a list of options, or browse all options.	To navigate to a feature or artifact quickly, such as a dashboard, repository, product backlog, Kanban board, build pipeline, you can use breadcrumbs or selectors.
services	Verticals, hub groups, products	Top-level services we sell.	Go to the **Services** page and enable **Azure Boards**.

TERM	PREVIOUS TERM	DEFINITION AND NOTES ON USAGE	SAMPLE USAGE IN DOCS
settings		There are several levels of settings: personal, team, project settings, organization (and possibly enterprise).	
sidebar		Vertical pane that hosts the left navigation menu.	

Pipelines terminology changes

The following table captures some of the changes made in terminology for pipelines, builds, and releases. Terminology of unified pipelines isn't addressed.

CONCEPT	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
build pipeline	build definition	A process that describes how your app should be built and tested.	(a) To create a pipeline, go to ... (b) When you're done, save the pipeline and queue a build. (c) Add the following snippet of code to your `.azure-pipeline.yml` file and push the changes to your Git repository.	The concept of build pipeline is more prominent when you use the web designer to author it. When using YAML files, you just author and push a YAML file (for example, `.azure-pipeline.yml`), and users don't think about "creating" a pipeline. Avoid repeatedly calling out *build pipeline* and just say *pipeline* when the context is clear. The verb that is used the most with this noun is *create* or *edit*.
build	build	The result of running a build pipeline once	(a) Queue a build ... (b) View the build logs ... (c) If the build fails, check if you made any changes to the pipeline (or to the YAML file) ...	This term can be used as a verb in a more general context, for example, *build your app*. When referring to gestures in Azure Pipelines, it's typically used as a noun, for example, *queue a build*. The verbs that are used the most with this noun are *view* or *queue*.

CONCEPT	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
artifact	artifact	A build publishes zero or more artifacts.	(a) Add a *Publish artifacts* task to your build pipeline... (b) Your build published an artifact... (c) Download the artifacts in a release...	A build publishes artifacts, a release downloads and consumes them. Common verbs: *publish*, *download*.
job	Phase or Job	A collection of tasks that run as one unit on an agent in a build or deployment	(a) Define a job in your pipeline... (b) You can configure dependencies between jobs... (c) Configure a matrix for your job... (d) Each job in the build...	Earlier, we used the term *Phase* as a definition-time concept, and *Job* as a run time concept. Now, we use the term *job* to mean both. Common verbs: *define*, *run*.
agent	agent	Software installed on a machine to run build or deployment jobs	(a) Use the Microsoft-hosted agent... (b) Install an agent...	Wherever ambiguous, we used to call this software *Build and Release agent*. We can continue to do so.
self-hosted agents	private agents	Agents that you set up and manage to run build and deployment jobs.	(a) You can use self-hosted agents to...	You can drop *self-hosted* if it's clear from the context. We're moving away from *Private agent* since sentences, such as *Use a private agent in a public project* are awkward.
Microsoft-hosted agents	hosted agents	Agents that are set up and managed by Microsoft to run customers' build and deployment jobs	(a) The following software is installed on Hosted VS2017 agents ...	Microsoft runs four types of Microsoft-hosted agents - Hosted VS2017, Hosted, Hosted Linux, and Hosted Mac. Use Microsoft-hosted agents when referring to any of these agents. Otherwise, refer to the specific type. This change has been made to be consistent with the change to self-hosted agents.
parallel jobs	concurrent pipelines	A unit of licensing in pipelines	(a) Ensure that you have enough concurrent jobs...	We changed the licensing recently from concurrent pipelines to concurrent jobs.

CONCEPT	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
release pipeline	release definition	A process describing how your app should be deployed to multiple stages.	(a) To create a pipeline, go to... (b) When you're done, save the pipeline, and create a release.	The verb that is used the most with this noun is *create* or *edit*.
release	release	The result of running a release pipeline once (or) A collection of artifacts that are is being deployed.	(a) Create a release... (b) Deploy the release to a stage...	This term is used as a noun. Common verbs: *create*, *deploy*
stage	environment	A logical break in the pipeline needed to separate concerns.	(a) Add a stage to the release pipeline... (b) Deploy the release to the QA stage.	
deployment	deployment	The result of running a release targeting a single stage.	(a) Deploy a release to a stage... (b) Check the deployment logs. (c) If the deployment fails, redeploy the release...	We often use *release* and *deployment* interchangeably when the context is clear. But, it would be better to keep them straight.
service connection	service endpoint	The credentials used to connect to a service outside of Azure DevOps services.	(a) Configure an Azure service connection... (b) Configure a GitHub service connection...	

Docs URLs

While all links to VSTS docs should redirect, update URLs to the new Azure DevOps URLs.

Project/Service URLs

OLD URL	NEW URL
{organization}.visualstudio.com/{project}	dev.azure.com/{organization}/{project}
{account}.visualstudio.com/{project}	dev.azure.com/{organization}/{project}

Other URLs

OLD URL	NEW URL	LINK TEXT	NOTES
https://twitter.com/vsts	https://twitter.com/AzureDevOps	@AzureDevOps	
https://azure.microsoft.com/case-studies/	https://azure.microsoft.com/case-studies/	Azure DevOps Case Studies	

OLD URL	NEW URL	LINK TEXT	NOTES
http://visualstudio.marketplace.com/vsts	http://visualstudio.marketplace.com/azuredevops	Azure DevOps extensions	Marketplace for both cloud and on-premises so ok to use Azure DevOps without qualifier
https://azure.microsoft.com/support/devops/	https://azure.microsoft.com/support/devops/	Azure DevOps support	en-us can be replaced with any locale
https://social.msdn.microsoft.com/Forumshome?forum=TFService	n/a	Visual Studio Team Services	No change!
https://visualstudio.uservoice.com/forums/330519-team-services	n/a	Visual Studio Team Services	No change!
https://stackoverflow.com/questions/tagged/vs-team-services	https://stackoverflow.com/questions/tagged/azure-devops	https://stackoverflow.com/questions/tagged/azure-devops	

Azure DevOps navigation guidelines

6/24/2021 • 2 minutes to read

With Azure DevOps, users will have a choice of using Horizontal (traditional) navigational model and the New navigation (with a vertical sidebar). To support users in navigating the UI, here are the recommended guidance elements to provide.

Provide explicit guidance for Quickstarts and Tutorials

For Quickstarts and Tutorials and other high page view articles, you should provide instructions on how to navigate to the starting page. These procedures should address both the new navigation experience and the previous navigation experience exemplified by the horizontal blue bar. For TFS 2013 through TFS 2018, there is ONLY the horizontal navigation experience. Azure DevOps Server 2019 (previously TFS) and later versions will only support the New Navigation experience.

For an example of usage, see the following Navigation basics (MD file).

Also, provide a link to Navigation basics for more information. For example: *To learn more, see Navigation basics.*

Provide a link to Navigation basics

For all topics that aren't Quickstarts or Tutorials, you can provide a link to the Navigation basics in place of explicit instructions.

Use INCLUDE files as much as possible

To cut down on redundancy and increase maintainability, use INCLUDE files as much as possible. In particular, use INCLUDE files when providing introductory content that is the same across several articles.

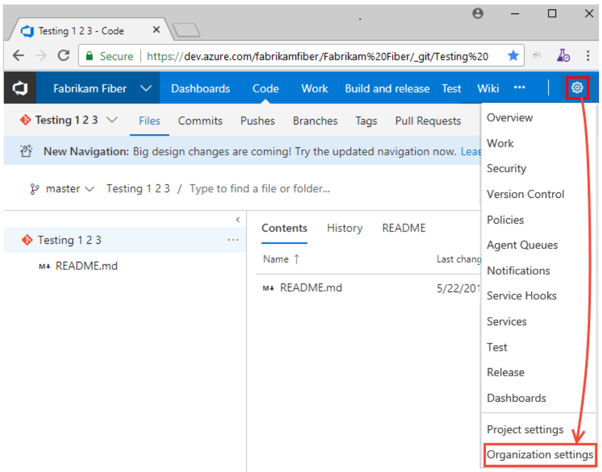
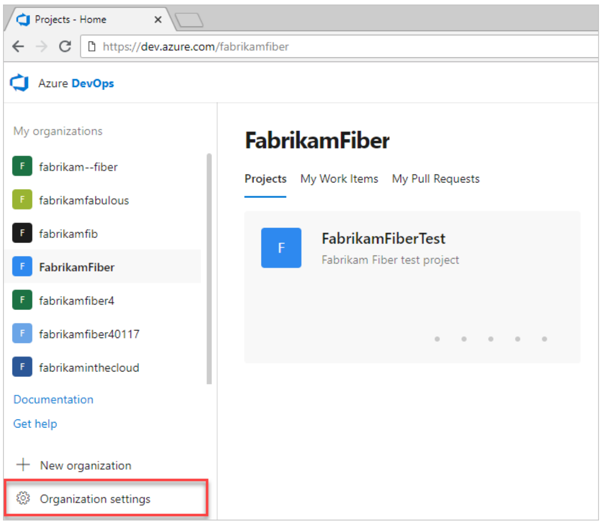
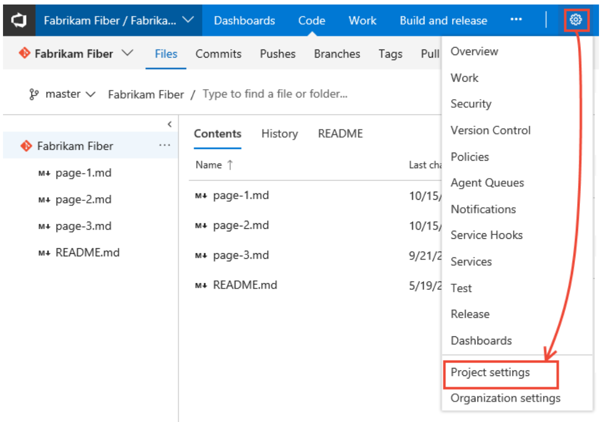
Use shared /includes image files as much as possible

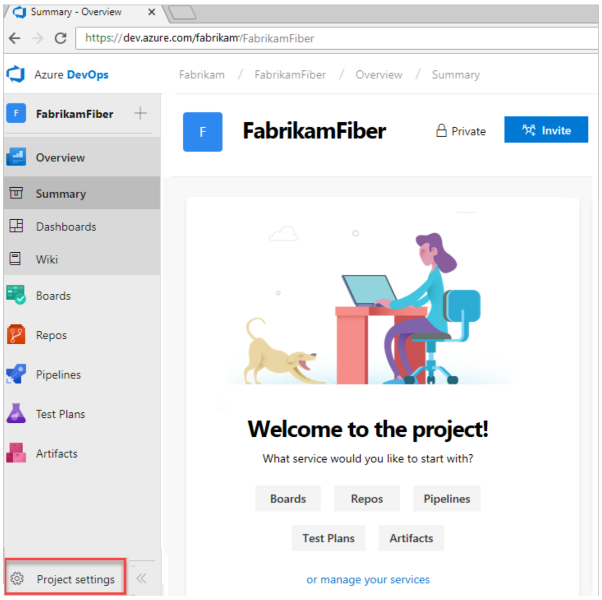
To reduce content redundancy, it's important to reuse images and content where it makes sense.

The **/docs/includes/media/settings** folder contains a number of common images used to open Organization and Project settings.

Examples are:

NAME	IMAGE

<table>
<tr><th>NAME</th><th>IMAGE</th></tr>
<tr><td>open-account-settings-horz-brn.png</td><td>
</td></tr>
<tr><td>open-admin-settings-vert.png</td><td>
</td></tr>
<tr><td>open-project-settings-horz.png</td><td>
</td></tr>
</table>

NAME	IMAGE
open-project-settings-vert-brn.png	

Publish Azure Databricks content

3/5/2021 • 6 minutes to read

Azure Databricks is a first-party Azure service that is primarily developed by Databricks Inc. The content for Azure Databricks is maintained by the Databricks Inc. documentation team in a private repository they own. Aside from a few cloud-specific differences, the documentation for Azure and AWS is the same and is reused on both documentation sites. The Databricks docs team updates content in a single source repo, and that repo is stripped of all AWS content before it is published on Microsoft Docs.

Workflow

1. The Databricks docs team makes updates in the source repo they own - `Databricks/docs`.
2. When the Databricks docs team is ready to publish, a CircleCI job strips the articles of all AWS content and puts the sanitized content in a new repo - `Databricks/azure-databricks-docs`.
3. Run the publishing script in the next section to move the new/updated Databricks content from `Databricks/azure-databricks-docs` to `MicrosoftDocs/databricks-pr`.
4. Open a pull request to the default branch using the output link from the publishing script.
5. Address any warnings by asking the Databricks team to update the content in the source repo and repeat the previous publishing steps. If there are no warnings, open a PR from the default branch to live.
6. Merge to publish.

Prerequisites

- Install PowerShell.
- Install Git.
- Access to the Databricks organization on Git Hub. Contact Leif Brown if you need access to publish.
- Clone the following repos to the same directory (save directory path for later – it's needed to run the script)
  - databricks/azure-databricks-docs
  - Your fork of MicrosoftDocs/databricks-pr
  - After your clone your fork, run the following command:

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

Script

Save the script as `Publish-Databricks.ps1`. Use the following command to run the script. Use the directory path that contains both prerequisite repos as the `-gitpath`.

```
Publish-Databricks.ps1 -gitpath "c:\docs"
```

- Windows
- Mac

```
#requires -version 5
<#
.SYNOPSIS
```

```
  Prepares Databricks content for publishing.
.DESCRIPTION
  Gets the latest changes from the azure-databricks-docs repo, moves them to the databricks-pr repo, cleans
up unwanted files, and commits the changes.
.NOTES
  Version:        1.0
  Author:         Mary McCready (mamccrea)
  Creation Date:  06/30/2020
  Purpose/Change: Initial script development
.EXAMPLE
  Publish-Databricks.ps1 -gitpath "c:\docs"
#>

#-----------------------------------------------------------[Parameters]------------------------------------
----------------------
param(
    [Parameter(Mandatory=$true)][string] $gitpath
)

#---------------------------------------------------------[Initializations]--------------------------------
----------------------

#Set Error Action to Silently Continue
$ErrorActionPreference = "Stop"

#----------------------------------------------------------[Declarations]----------------------------------
----------------------

# Script Path/Directories
$azuredbksdocs = $gitpath + '\azure-databricks-docs'
$databrickspr = $gitpath + '\databricks-pr'
$azuredbksdocsarticles = $azuredbksdocs + '\docs'
$azuredbksdocskb = $azuredbksdocs + '\kb'
$azuredbksdocsscenarios = $azuredbksdocs + '\scenarios'
$azuredbksdocstoc = $azuredbksdocs + '\databricks-pr\TOC.yml'
$databricksprarticles = $databrickspr + '\databricks'
# Date/Time format
$date = Get-Date -UFormat "%m-%d-%Y"
$time = Get-Date -Format "HHmmss"
# Git variables
$commitmessage = 'merge from source ' + $date
$branchname = 'merge-' + $date + $time

#-----------------------------------------------------------[Execution]------------------------------------
----------------------

# Get latest changes to azure-databricks-doc repo (owned by Databricks)
Write-Host "Getting latest changes to azure-databricks-doc repo"
Set-Location $azuredbksdocs
git checkout master
git pull
Write-Host "Finished getting changes to azure-databricks-docs"


# Sync databricks-pr repo (owned by Microsoft) with master and create new branch
Write-Host "Syncing databricks-pr with master and creating new branch"
Set-Location $databrickspr
git checkout master
git pull upstream master
git checkout -b $branchname
Write-Host "Finished Syncing databricks-pr with master and creating new branch"

Write-Host "Removing old content from databricks-pr"
# Get list of all items in the databricks-pr\databricks directory
[System.Collections.ArrayList]$oldtberemoved = Get-ChildItem $databricksprarticles | ForEach-Object {
$_.FullName }

# Remove items in the excludes list from the items to be removed
[System.Collections.ArrayList]$excludes = ($databricksprarticles + '\docfx.json'), `
```

```
    ($databricksprarticles + '\breadcrumb'),`
    ($databricksprarticles + '\index.yml')

Write-Host $excludes

foreach($e in $excludes){
    $oldtberemoved.Remove($e);
}

# Remove all items in the to be removed list
foreach($item in $oldtberemoved){
    Remove-Item $item -Force -Recurse
}

Write-Host "Finished removing old content from databricks-pr"

Write-Host "Copying new content to databricks-pr"
# Copy the docs directory to databricks-pr
Get-ChildItem -Path $azuredbksdocsarticles | Copy-Item -destination $databricksprarticles -Recurse -
Container

# Create a new kb directory in databricks-pr and copy the kb directory contents to databricks-pr
$newkb = New-Item -Path $databricksprarticles -Name "kb" -ItemType "directory"
Get-ChildItem -Path $azuredbksdocskb | Copy-Item -Destination $newkb -Recurse -Container

# Create a new scenarios directory in databricks-pr and copy the kb directory contents to databricks-pr
$newscenarios = New-Item -Path $databricksprarticles -Name "scenarios" -ItemType "directory"
Get-ChildItem -Path $azuredbksdocsscenarios | Copy-Item -Destination $newscenarios -Recurse -Container

# Copy the TOC.yml to databricks-pr
Get-ChildItem -Path $azuredbksdocstoc | Copy-Item -Destination $databricksprarticles -Recurse -Container

Write-Host "Finished copying new content from databricks-pr"

# Delete all AWS/S3 references, the sitemap, the index.md, and the /spark/latest/faq directory
Write-Host "Deleting unwanted content"
Get-ChildItem -Path $databricksprarticles -Include *aws*.* -Recurse | ForEach-Object { $_.Delete()}
Get-ChildItem -Path $databricksprarticles -Include *s3*.* -Recurse | ForEach-Object { $_.Delete()}
Get-ChildItem -Path $databricksprarticles -Include *sitemap*.* -Recurse | ForEach-Object { $_.Delete()}
Remove-Item -LiteralPath $databricksprarticles\index.md
Remove-Item -LiteralPath $databricksprarticles\spark\latest\faq -Force -Recurse
Write-Host "Finished deleting unwanted content"

# Commit changes and delete local branch
Write-Host "Committing changes"
git add .
git commit -m $commitmessage
git push origin $branchname
git checkout master
git branch -D $branchname
Write-Host $branchname " committed. Go to GitHub to open a PR: https://github.com/MicrosoftDocs/databricks-
pr"
```

Points of contact

- Leif Brown leif.brown@databricks.com - Databricks Director of Documentation
- Stephanie Bodoff stephanie.bodoff@databricks.com - Databricks Staff Technical Writer
- Santosh Perla Santosh.Perla@microsoft.com - Azure Databricks Program Manager

Update Azure Databricks content

3/5/2021 • 2 minutes to read

Azure Databricks content is updated by a one-way pipeline from a Databricks-owned source repo to Microsoft Docs. Because of this, you can't make updates directly in the MicrosoftDocs/databricks-pr repo. To make updates, reach out to one of the following points of contact:

- Leif Brown leif.brown@databricks.com - Databricks Director of Documentation
- Stephanie Bodoff stephanie.bodoff@databricks.com - Databricks Staff Technical Writer
- Santosh Perla Santosh.Perla@microsoft.com - Azure Databricks Program Manager

Metadata

Metadata is automatically generated/updated by the CircleCI job. Because of this, custom metadata is not supported and all topic types are `conceptual`.

Exceptions

The following folders in the databricks/azure-databricks-docs repo do not exist in the source repo and must be updated outside of the CircleCI process.

- databricks-pr: contains the TOC.
- scenarios: contains all top-of-funnel content (overviews, quickstarts, tutorials, and workspace and SQL Analytics landing pages).

The following files in the MicrosoftDocs/databricks-pr repo do not exist in the source repo or the databricks/azure-databricks-docs repo and must be updated directly in MicrosoftDocs/databricks-pr.

- databricks/docfx.json
- databricks/index.yml: overall Databricks service landing page

Repository and publishing information for the Partner Center business

6/28/2021 • 4 minutes to read

There are several separate GitHub repositories for the Partner Center docs. The following table describes the purpose of each repo, and shows the public/private pairing if any. The public docs URL is also shown where applicable.

REPO NAME	PURPOSE	PRIVATE REPO	PUBLIC REPO	DOCS URL
Partner Center	Conceptual docs for Partner Center and customer docs for the commercial Marketplace.	partner-center-pr	None	Docs URL
Azure Docs	Conceptual docs for Azure commercial Marketplace and Billing.	azure-docs-pr	azure-docs	Docs URL
Office store Docs	Conceptual docs for commercial Marketplace offering in the Office store.	NA	office-store-docs	Docs URL
Partner Center SDK	Conceptual docs and hand written reference docs for Partner Center. Overlaps with some of the autogenerated reference docs as well.	partner-center-sdk-pr	partner-center-sdk	Docs URL
Partner Center Rest	Partner Center Rest API Reference	partner-center-rest	NA	Docs URL
Partner Rest	Partner REST API (different from Partner-center)	partner-rest-pr	partner-rest	Docs URL
Partner Center Swagger Specs	Swagger specs to publish automated REST documentation. There is a partner-center-swagger-admin group to join to get access to publish the swagger specs.	partner-center-swagger	NA	NA

REPO NAME	PURPOSE	PRIVATE REPO	PUBLIC REPO	DOCS URL
Partner .NET SDK	Partner Center .NET SDK Reference	Windows-partner-center-dotnet	None	Docs URL
Partner Java SDK	Partner Center Java SDK Reference	None	partner-center-java-docs	Docs URL
Partner PowerShell SDK	PowerShell reference docs for Partner Center	None	partner-center-docs-powershell	Docs URL
Partner Center China Redirect	Hosts several redirects from URLS to old MSDN URL that points to 21Vianet, such as this url	partner-china-redirect	None	redirects only
Partner Center Red Tiger	Hosts .pdf and .docx files for downloads. Several agreements are hosted here that are not widely available on docs. The URL is a pattern such as `https://query.prod.cms.rt.microsoft.com/*` . Though one contributor can post and approve an asset, a second person has to publish it.	RedTiger	None	Varies - Example

Pull Request Reviews

1. **Azure-docs-pr** pull request reviews are handled by the Aquent managed team. Type `#sign-off` in the PR, and the PRMerger bot will add the `ready-to-merge` label, which the reviewers monitor for.

2. **Partner-center-pr**, **Partner-center-sdk**, and **Partner-center-sdk-pr** pull requests are also handled by the Aquent managed team.

   You can request a technical review before signing off. In the PR comments, type @partnerdocs or mentioning an individual, and assign a review at the upper right of the pull request.

   Type `#sign-off` in the PR, and the PRMerger bot will add the `ready-to-merge` label, which the reviewers monitor for. They will review within an hour usually, but worst case 1 day.

   If the reviewers give you blocking feedback that prevents them from merging, they will type `#hold-off` and it resets the `ready-to-merge` label. You should adjust the source markdown, and `#sign-off` again to get their attention to request the merge.

3. The remaining repos listed above are not monitored for `#sign-off` regularly. Follow the instructions in the Pull Request opening template, and @ mention the repo admins if in doubt in the comments to get their attention.

Publishing schedules

The busiest repos are published live on a schedule. The other are on-demand.

1. **Azure-docs-pr** publishing is handled by CPS. Generally happens three times a day. Schedule here
2. **Partner-center-pr** and **Partner-center-sdk-pr** docs publish twice a day.

   There is a set of custom Azure DevOps Pipeline jobs that run on a schedule. The scheduled times are 9am and 3pm PST. The job pulls from main branch and merge into live branch. Occasionally the publication fails if there is a merge conflict or a build error.

   Repo admins can also merge from main branch to live branch manually if needed.

3. The other docs are all published on demand only by the repo admin (merge main branch to live branch).
4. Public repos that are paired to a private repo use a **repo_sync_working_branch** branch provided by OPS to keep the two repos in sync. The repo-sync branch is used to forward the public commits from main branch in the public repo in a pull request into the main branch of the private repo. A pull request shows up in the private repo when the sync mechanism has forwarded public changes. However, that PR does not automatically merge, so an administrator needs to be monitor and sign-off or merge it.

Repo security

The members of the group @MicrosoftDocs/partnerdocs help with the content in the Partner Center Docs.

The group @MicrosoftDocs/partner-center-docs-admins can help with repo administration, such as setting up branches, resolving build issues, and publishing out of band. Content production Service (CPS) helps with Azure-docs-pr, but does not assist with branches and administration for the other partner repos.

Use the Docs Support general channel in Microsoft Teams if there is a problem with the repo that requires advanced troubleshooting.

Autogenerated REST docs

1. Swagger specs (.json files) go into this repo:

   https://github.com/MicrosoftDocs/partner-center-sdk-pr/blob/master/rest-swagger-specs/

   Anyone in MicrosoftDocs org can submit a PR, but only repo admins or PR Reviewers can merge the PR.

   Developers who want to publish new swagger specs for Partner Center can discuss with the folks in this team: @MicrosoftDocs/partner-center-docs-admins

2. Run a CI/CD pipeline to pull the Swagger specs in and build the autogenerated YAML page output.

   Join the security group on ID Web: pcswaggerdoc

3. Publication

   Rest API Docs are staged here: https://review.docs.microsoft.com/rest/api/partner-center/audit?branch=main

   Once public, the docs render here: https://docs.microsoft.com/rest/api/partner-center/audit.

Use applies to and appropriate roles in Partner Center

5/27/2021 • 2 minutes to read

Specific attributes in the meta-data

Assign values for both ms.service and ms.subservice to your article. Generally, the values for the articles in Partner Center include the following (there are a few exceptions for topics related to Marketplace, etc.):

ms.service:

- partner-dashboard

ms.subservice:

- partnercenter-mpn
- partnercenter-csp
- partnercenter-incentives
- partnercenter-sdk (for developer docs only)

Requirements for every article

An article may contain an **Applies to** section. However, for articles that just apply to the general Partner Center, you do not need **Applies to**. Use **Applies to** only if you are calling out cloud instances that are different from the standard Partner Center. In this case, also include Partner Center as a value (for example: **Applies to**: Partner Center | Partner Center for Microsoft Cloud Germany). Otherwise, you don't need to tell readers that the article applies to Partner Center. If **Applies to** is needed it should be at the top, directly following the H1.

Next, with few exceptions, each Partner Center article needs an **Appropriate roles** section at the top, either directly following **Applies to**, if needed, or the H1.

Values for both **Appropriate roles** and **Applies to** should be on a single line and separated by a pipe (|).

Special instructions for the SDK

In the SDK, we currently do not use **Appropriate roles**. We do use the **Applies to** section, although the values used may be different than the values described in this guideline article. You should format the values for **Applies to** in the same way as for the Partner Center documentation (values on a single line and separated by a pipe).

Possible values for Applies to

Applies to refers to websites, not Partner programs such as CSP. Accepted terms used here include (but are not limited to):

- Partner Center for Federal partners
- PartnerCenterforMicrosoftCloud for US Government
- PartnerCenterforMicrosoftCloudGermany
- Partner Center (when used with another value)

(Remember, for articles that just apply to the standard Partner Center, there is no need to include an **Applies to** section.)

Possible values for Appropriate roles

Appropriate roles refers to the Partner Center user role that can accomplish the task being explained in the article or that can see the information in the dashboard. The author of the content should be aware of which type of user will be able to make use of the content. We don't want partners to waste their time trying to do tasks they don't have the needed credentials to do.

Roles can overlap--an employee may have more than one role.

There are a few articles, such as the overview article **What is Partner Center?**, that are appropriate for all partners. The correct terminology for this is:

- All partners interested in Partner Center *or*
- All partners interested in learning more about

Otherwise, valid roles include (but are not limited to) the following values:

Manages commercial transactions and Azure AD:

- Global admin
- Account admin
- User management admin

CSP specific:

- Billing admin
- Admin agent
- Sales agent
- Helpdesk agent
- Referrals admin
- Incentives admin
- Incentives user
- Executive report viewer
- Report viewer

MPN specific:

- MPN partner admin
- Account admin

PSC specific:

- Partner Sales Connect (PSC) seller
- Partner Sales Connect (PSC) admin
- Partner Sales Connect (PSC) deal manager

Additional roles:

- Guest user
- CPV admin
- Advisor
- System admin or system customizer on the CRM
- Referrals user

General roles

- All Partner Center users
- Direct bill partners
- Indirect resellers
- CSP partners

Examples of correct usage

Applies to: Partner Center | PartnerCenterforMicrosoft Cloud for US Government

Appropriate roles: Global admin | Admin agent

Write a Partner Center announcement

7/7/2021 • 8 minutes to read

The purpose of an announcement is to inform Microsoft partners about offer promotions, upcoming new features, changes to policy, or other changes that might affect their businesses.

Treat an announcement like a news story. That is, get right to the point. Briefly cover only what partners need to know, and avoid providing the details you might ordinarily handle in a longer article, such as a Help topic. Announcements are not where partners learn how to complete a task. Ideally, your announcement would link to a Help topic or to the Operations gallery for more in-depth information.

Required sections

- **Categories**: Include the date of the announcement.
- **Summary**: Provide high-level information, including the date when the feature or the change becomes effective.
- **Impacted audience**: List who is affected by this information.
- **Date published**: Specify when the announcement goes live.
- **Details**: Describe the announcement in detail.
- **Next steps**: Provide resources to help partner prepare for the change, release, or event.
- **Questions**: (Optional) Link to Partner Center support, Yammer communities, or other resources so that partners can ask questions about the announcement.
- **Change log**: (Optional) List dates of changes or updates to the announcement.

Heading formatting

- **Heading length**: For the main heading, or announcement title, use no more than 60-75 characters, including spaces. The heading is displayed as a link on the main Partner Center announcements page and as an internal TOC entry on the announcements page for a specified month.

- **Heading size**: The announcement title is an H2 heading (prefixed with ## and a space); the section headings are H3s (prefixed with ### and a space).

Heading examples

- An effective heading:

  Audit log changes to include two new operation types for DAP (Why: this heading is short and to the point.)

- A less effective heading:

  Phase 1 Now Live: EU/EFTA & UK Change of Partner Billing Currency for new commerce offers (Why: this heading contains unnecessary information, and the timing ["Phase 1"] and audience ["new commerce"] are unclear.)

  - Here's an improved version of the same announcement:

    Billing currency will change for some EU/EFTA offers (Relevant details are placed in the body of the announcement.)

Announcement examples

New offer promotion or offer change

(H2) Dynamics 365 offers for February 2021

(H3) Categories

- Date: 2021-01-04
- Offers

(H3) Summary

New Dynamics 365 offer changes will be launched in February 2021.

(H3) Impacted audience

Partners participating in the Cloud Solution Provider (CSP) direct bill partner, indirect provider, and indirect reseller incentive programs

(H3) Details

Microsoft is announcing upcoming new Dynamics 365 offer changes that will be launched in February 2021 for:

- Dynamics 365 Customer Voice Additional Response

...

(H3) Next steps

Review the resources about this topic and share them with the appropriate stakeholders in your organization.

(H3) Questions?

For any further questions about these offers, check your relevant Yammer communities.

Business process change

(H2) New updates to the Referrals module in Partner Center

(H3) Categories

- Date: February 5, 2021
- Capabilities

(H3) Impacted audience

- Partners with Co-sell Ready offers
- Partners with a transactable offer using an Azure incentive-eligible solution in the

...

API change

(H2) Reminder: Introducing API throttling to partners calling Partner Center APIs

(H3) Categories

- Date: 2021-02-16
- Grow your Business

(H3) Summary

Microsoft will be implementing API throttling to allow a more consistent performance for partners calling the Partner Center APIs.

(H3) **Impacted audience**

All partnerstransactingthrough the Cloud Solution Provider (CSP) program

(H3) **Details**

Beginning in February 2021, Microsoft will be implementing API throttling to allow a more consistent performance for partners calling the Partner Center APIs.

...

(H3) **Partnerbenefits**

This update will reduce the complexity and overhead with multi-currency invoicing in the EU/EFTA & UK for the new commerce experience.

- Partners will receive a consolidated invoice in a single currency and will no longer receive an invoice for each customer location currency
- Incentive payouts will be in the same currency as the partner's invoice currency

...

Deprecation notice

(H2) **Deprecation of PUT and GET Qualification APIs for Education customer validation**

(H3) **Categories**

- Date: February 4, 2021
- Capabilities

(H3) **Impacted audience**

Partners selling Academic, Non-Profit, and GCC offers through the Cloud Solution Provider program using the Partner Center API

...

Pricing change

(H2) **Microsoft 365 E3 price changes coming for March 2021**

(H3) **Categories**

- Date: February 11, 2021
- Offers/Markets

(H3) **Impacted audience**

All partners transacting through the Cloud Solution Provider (CSP) program in EUR, AUD, GBP, and JPY.

Walkthrough: Creating a Partner Center announcement from soup to nuts

Step One: Receive the announcement

1. The process begins when you get an email (usually from Katie Kent) with the announcement attached as a Word document. The email Subject line should also contain the announcement number, title, contact, and publish date. They usually want it published in 48 hours, which is the lead time we've asked them for.
2. Within the hour if possible, create a deliverable for the announcement here, using the following:

FIELD	TEXT
Title	ANNOUNCEMENT 9315: Product launch calendar – March 25
Assigned to	(yourself)
State	Started
Iteration	(current)
Description	Publish attached announcement by Thursday, 3/25
Attachment	(original attachment that came in email)

3. Do a "Reply all" to the announcement email that looks something like this:

```
Hi, Katie,

We've created deliverable 32223758 to track announcement 9315: Product launch calendar, and it should go
live Thursday as requested. I'll keep you posted.

Pete
```

If you can't create a deliverable within the hour, send something like this and follow up later:

```
Thanks, Katie,

We should able to publish this Thursday as requested. I'll keep you posted.

Pete
```

...and then create the deliverable when you can.

Step Two: Clean up and structure the announcement

Here's where we get the content ready for markdown. I do this in Word – you do what you like.

4. Open the announcement email, and save the attached Word document to a convenient working folder. Include the four-digit announcement number in the filename if it's not already there, such as **9315 Product launch calendar.docx**.
5. Open the document, select and copy all content, paste into a new document as unformatted text.
6. Save that new document – I usually call mine something like **PA 9315.docx**.
7. Working in that new document now, remove extraneous text at the end, such as **"Let's do great work together"** and any content after that.
8. Remove any extra spaces, line breaks, etc.
9. Structure the content into the standard announcement sections below (this is usually done already):

(H2)
[ANNOUNCEMENT TITLE]

(H3) **Categories**

[date]

[content]

(H3) **Impacted audience**

[content]

(H3) **Details**

[content]

(H3) **Next steps**

[content, bulleted list]

(H3) **Questions?**

[content]

Step Three: Edit the announcement

10. Once you've got it cleaned up and structured, there are generally four things to edit. Refer back to that original Word document as needed.

- Content
  - At the very least, make it Acrolinx-ready
  - Tone down anything that sounds too much like marketing
  - Watch for acronyms that aren't spelled out the first time – this is common
  - Improve the content as needed and time permitting
- Lists (bulleted and numbered)
- Tables
- Links – Go to Word document, right-click link => Edit Link => ctrl+c, go to announcement document, put [] around linked text followed by () with link inside. Make sure /en-us/ is edited out

Now you've got a Word document that's markdown-ready.

Step Four: Add the announcement to the announcements file

This is just standard Gitbash, Github, VS Code, however you like to do it. I use Gitbash and VS Code.

11. Create a branch, such as `pete-announce-9315`.

12. Open VS Code.

13. Open this month's file: `2021-april.md`.

14. Copy your content from Word and paste into the file, right after this text:

```
This page provides the announcements for Microsoft Partner Center for April 2021.
```

15. Do whatever cleanup VS Code requires – lists will probably need spaces redone, etc.

16. Look at what had been the latest announcement, see what number is in its title, and make yours the next number. For example, if `15` used to be at the top, you'll add yours at the top as number `16`.

17. Preview, fix, clean up, etc. and save.

18. Do the two things I always forget:

- Update the date on the article
- Search for "en-us" and remove if found

19. You're done editing. Save your file but keep VS Code open – next you'll do the index.

Step Five: Update the index

The index is essentially a list of all announcements, with the most recent one at the top. That's where yours is going to go.

20. Open the file called **index.md**.

21. You'll see a table with a lot of rows that look like this:

```
\|  Grow Your Business  |  [ANNOUNCEMENT TITLE]  (2021-april.md#7)  |  April 19  |
```

22. Create a new row for your announcement. I usually just copy an existing row, paste it at the top, then update it with the right **category**, **title**, **filename** and **#**, and the **date** you want it to go live.

23. Preview, fix, and save. When you're done, close VS Code.

24. The usual: add-commit-push, open PR, watch for build results, fix as necessary, sign off.

Step Six: Confirm publication and update the team

25. When the time and day of publication arrive, go to the **Announcements** page and confirm that yours is there.

26. When it's live and looking good, do another "Reply all" to that original email thread and say something like this:

```
Hi, Katie,

Announcement 9315: Product launch calendar is live as of this morning. Let me know if you have any
questions. Thanks!

Pete
```

27. Open your deliverable, mark it as **COMPLETED**, then save and close.

28. You're done!

Finally: A note on organization

It's quite common to be editing and publishing numerous announcements at once. It can be very useful to create a table for yourself to track the following for each announcement:

- Announcement number
- Deliverable number
- Owner or PM
- Subject
- Pull request number
- Requested publication date
- Status

...

How to contribute to Power BI documentation

7/7/2021 • 2 minutes to read

This article provides specific guidance for Microsoft employees to contribute to Power BI documentation.

GitHub repository

Use MicrosoftDocs/powerbi-docs-pr for internal contributions:

- Internal pull requests
- To get permissions to contribute to the repository, see Set up a GitHub account.

Use MicrosoftDocs/powerbi-docs for public contributions:

- Public pull requests

Email contact

Use the PbiSqlBiContent email alias to discuss technical content and pull requests related to the powerbi-docs-pr and powerbi-docs repositories.

Customer feedback channel

The Power BI docs use GitHub issues for customer feedback.

Visual Studio work item tracking

Use this new work template to track work items or bugs for this repository:

- Area Path: `TechnicalContent\Data and Analytics\BI\Power BI`
- Iteration Path: `TechnicalContent\3WeekSprints\Future`
- State: `Proposed`
- Tags: `Power BI` and other variations such as `Power BI Desktop`, `Power BI Embedded`, `Power BI Mobile`, and `Power BI Pro`.
- Description: Explain the issue or requirement and provide a link when possible. If there's a due date, explain why.
- Story Points: The number of days estimated to do the work. For example, 1.0, 0.5, or 7 (days).

Example article metadata

The following metadata block is an example for a Microsoft Power BI article. For detailed metadata information, see Metadata overview. Recommendations for Power BI docs:

- Mention the product name, Power BI, in the **title** and **description** tags for SEO.
- Specify the **services**, **ms.service**, and **ms.workload** tags by using technology-specific values such as `powerbi` or `power-bi-embedded`.
- Specify a **ms.subservice** value, such as `powerbi-desktop`, `powerbi-service`, `powerbi-admin`, `powerbi-gateways`, `powerbi-mobile`, `powerbi-developer`, `powerbi-report-server`, `powerbi-template-apps`, or `powerbi-custom-visuals`.
- Refresh the **ms.date** tag each time you make a significant change to the article for freshness.

```
---
title: Overview of Power BI Desktop
description: Power BI Desktop lets you build advanced queries, models, and reports that visualize data.
ms.service: powerbi
ms.workload: powerbi
ms.subservice: powerbi-desktop
author: githandle
ms.author: msalias
ms.topic: conceptual
ms.date: 04/23/2021
---
```

Free sign-up recommendation

Use this sentence in each quickstart that you write to encourage users to try a free trial:

```
If you're not signed up for Power BI, [sign up for free](https://app.powerbi.com/signupredirect?
pbi_source=web) before you begin.
```

Open issues

- Docs team info: OneNote

Get started contributing to SQL documentation

7/7/2021 • 2 minutes to read

This article provides guidance on how Microsoft employees can contribute to SQL documentation published at https://docs.microsoft.com/sql/ and https://docs.microsoft.com/azure/. SQL Server documentation is managed in two main GitHub repositories:

- sql-docs-pr : SQL Server on-premises content.
- azure-docs-pr: SQL content on Azure.

General tasks

If you need to make a minor change to an existing article, see Quickly edit an existing article. For all other edits, use the following resources in the Contributor's Guide:

TASK	CONTRIBUTOR'S GUIDE RESOURCES
Setup a local repository for larger or frequent edits	Sign up for a GitHub account Install Git and VS Code Set up a Git repository locally
Submit larger or frequent edits	Create working branches and submit pull requests Edit content in Visual Studio Code
Follow content standards	Write an overview Write a quickstart Write a tutorial Write a sample Edit a TOC
Improve your writing	Follow writing principles Use Acrolinx Review the Microsoft style guide

> **TIP**
> If you are adding new content or making significant changes, please include a member of the content team on this work to avoid duplication and to align with ongoing content strategies.

SQL tasks

There are also writing tasks and resources specific to SQL documentation related to the tasks in the previous section.

TASK	SQL DOCS RESOURCES
Ask a question	Email the SQL content team
File a new content request	Open a new Azure Boards user story

TASK	SQL DOCS RESOURCES
View current work	View the SQL content dashboard
See staged content	Visit the review site and select a branch
SQL writing tasks	Edit metadata in SQL articles Use applies-to includes Manage versioning Work with SQL release branches

Next steps

For more information, see the Contributor's Guide.

SQL documentation guidance

6/24/2021 • 7 minutes to read

This article provides guidance for Microsoft employees who contribute to SQL documentation in `sql-docs-pr` and `azure-docs-pr`.

How to contribute

You can contribute to directly to SQL docs either by editing in your browser or by working locally with VS Code and GitBash. For more information, see Get started contributing to SQL Documentation.

> **IMPORTANT**
>
> If you need to make changes for a future release, please use an appropriate release branch and follow the steps on working with release branches in this article.

Author publishing process

Authors refer to doc owners, such as content writers, or program managers (PMs) from the product group. Authors should follow the publishing process outlined in this section. Writers that do not fit into these two categories should follow the contributor guidance.

To submit your documentation change, open a GitHub pull request (PR). For more information, see How to create a pull request. Please note the following:

- After you complete your changes, you must add #sign-off in the comments of the PR.
- The PR is reviewed for standards before merging. You will be asked to fix any issues.
- Any SQL PR that does not show activity for 30 days will be closed. You can reopen it at any time.

After your changes are merged, they are ready to publish. Twice a day, the `master` branches of `sql-docs-pr` and `azure-docs-pr` are pushed to the public repositories. This publishes the content live. Publishing happens at roughly 10:30am and 3:30pm PST.

> **NOTE**
>
> If you are working in a release branch, then the content is not published until that release branch is merged into `master`.

Email contact

Use the sqldocsPR email alias to discuss technical content and pull requests related to `sql-docs-pr` and `sql-docs`.

Visual Studio workitem tracking

To see what is happening in SQL Server content, access the SQL Azure Boards Dashboard.

If you have a new content idea or need to partner with the content team in any way, file a new Azure Boards work item.

SQL docs guidance

The following section provides some additional guidance on working in the **sql-docs** and **sql-docs-pr** repositories.

> **IMPORTANT**
>
> The information in this section is specific to **sql-docs**. If you are editing a SQL article in the Azure documentation, see the Readme for the azure-docs repository on GitHub and the Azure section of the internal contributor's guide.

The sql-docs-pr repository uses several standard folders to organize the content.

FOLDER	DESCRIPTION
docs/	Contains all published SQL Server content. Subfolders logically organize different areas of the content.
docs/includes/	Contains include files. These files are blocks of content that can be included in one or more other topics.
./media/	Each folder can have one **media** subfolder for article images. The **media** folder in turn has subfolders with the same name as the topics that the image appears in. Images should be .png files with all lower-case letters and no spaces.
toc.yml	A table-of-contents file, in the newer YAML format. Each subfolder has the option of using one toc.yml file.

Contributor guidance

Product group program managers or content developers approve all updates in the private repository. Contributors who are not in the product group may add PMs or content developers as reviewers, but they may not sign off on new content or updates. This policy ensures that content aligns with product group intent and supportability.

Procedure

Once a Pull Request (PR) has been submitted, you must:

1. Check the Acrolinx scorecard and ensure to implement the suggested changes.
2. Review the staged content to ensure formatting is okay and your changes look good. Do this by selecting the "View" link in the **Validation status: Passed** comment under **Preview URL** within your Pull Request.
3. Check the metadata for who the ms.author of the article is, and email them directly with the link your PR. CC the content team (rogue1@microsoft.com). The author or someone on the content team will sign off.

Guidelines

These are general guidelines that all authors must follow, but that are not explicitly stated elsewhere.

- When providing a link to content, content on docs take priority. Do not provide a link to a blog if a docs article exists. If providing a link to a blog, the blog should be vetted by the article author for technical accuracy. Ideally, the content of the blog should be migrated to docs, but the author needs to make that call.

Article metadata

Each article has a metadata block at the top. For detailed metadata information, see Metadata guidance. In the near future, we will post more specific metadata guidance for SQL topics here. For now, use metadata in a similar way as existing SQL articles in your area.

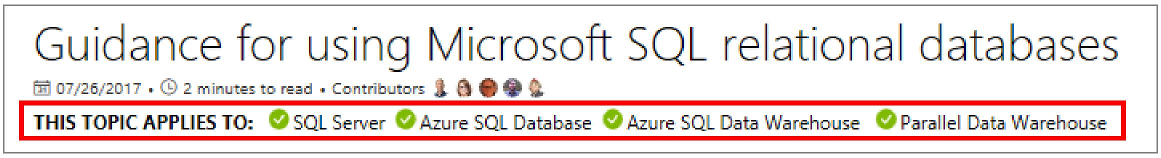
Applies-to includes

Each SQL Server article contains an **applies-to** include file after the title. This indicates what areas or versions of SQL Server the article applies to.

Consider the following Markdown example that pulls in the **appliesto-ss-asdb-asdw-pdw-md.md** include file.

```
[!INCLUDE[appliesto-ss-asdb-asdw-pdw-md](../includes/appliesto-ss-asdb-asdw-pdw-md.md)]
```

This adds the following text at the top of the article:



To find the correct applies-to include for your article, use the following tips:

- For a list of commonly used includes, see SQL Server version and applies-to include files.
- Look at other articles that cover the same feature or a related task. If you edit that article, you can copy the Markdown for the applies-to include link (you can cancel the edit without submitting it).
- Search the docs/includes directory for files containing the text "applies-to". You can use the **Find** button in github to quickly filter. Click on the file to see how it is rendered.
- Pay attention to the naming convention. If there are x's in the name, they are usually placeholders indicating the lack of support for a service. For example, **appliesto-xx-xxxx-asdw-xxx-md.md** indicates support for only Azure SQL Data Warehouse, because only **asdw** is spelled out, whereas the other fields have x's.
- Some includes specify a version number, such as **tsql-appliesto-ss2017-xxxx-xxxx-xxx-md.md**. Only use these includes when you know the feature was introduced with a specific version of SQL Server.

There is some overlap between applies-to includes and versioning, but it is not complete. For now, there is a need to use both applies-to includes and versioning to help customers navigate the SQL documentation.

Versioning

In addition to applies-to includes, SQL Server content has implemented versioning. Users can select a SQL Server service or product version from the left-hand TOC to scope the content that is displayed. To understand how to use versioning in SQL topics, see SQL versioning guidance and scenarios for writers.

Branches

SQL Server content uses the following branches for releasing content. The instructions in this guide use the **master** branch, which is for immediate release. But there are also release branches for delayed releases.

BRANCH	DESCRIPTION
master	Changes publish twice a day from this branch.
release-sql2017	Changes for the next cumulative update of SQL Server 2017.

BRANCH	DESCRIPTION
release-sqlseattle	Changes for the next CTP of SQL Server vNext.

Work in a release branch

Use the following steps to work in a release branch like **release-sql2017** or **release-sqlseattle**.

1. If you have not locally cloned the **sql-docs-pr** repository and installed gitbash and VS Code, see Setup a local repository for larger or frequent edits.

2. Follow the steps in Check out a release branch to check out a working branch for your release branch. For example, if you want to make changes to SQL Server vNext, you could run the following command to create a working branch **mychanges** for **release-sqlseattle**...

```
cd /sql-docs-pr/docs
git checkout -B mychanges upstream/release-sqlseattle
git pull upstream release-sqlseattle
```

3. Make all of your changes and then Commit and push your working branch.

4. Then create a pull request for your changes. It helps to add `@githubusername` to the comments for any content lead or PM associated with the change to facilitate awareness and collaboration.

   Caution

   You must insure that your pull request pulls your changes into the corresponding release branch, such as **release-sqlseattle**, and not **master**. If you accidentally pull your changes into **master**, all of the pre-release content will go live (not just your PR). Having a lot of extra commits that are not associated with your change is a clue that you have made this mistake.

If you have any questions or need any help on how to work with release branches, email sqldocsPR.

SQL Syntax colorizer

Most SQL Server articles about Transact-SQL statements contain two types of code *blocks*, sometimes called *fenced code blocks*. A fenced code block is a set of several lines, with a triple-backtick as the first line, and another triple-backtick line as the last line.

The first triple-backtick line should have a trailing *colorizer code*, sometimes called a *language specifier*.

- *Regular executable SQL example* - In a code block for a regular SQL code example that the customer could perhaps execute, the colorizer code is `sql`.
- *Syntax description of SQL* - In a code block for an SQL syntax description, the colorizer code is `syntaxsql`.
  - `syntaxsql` became available on March 17, 2020.
  - An example follows.

````
```syntaxsql
DROP DATABASE [ IF EXISTS ] { database_name | database_snapshot_name }
    [ ,...n ] [;]
```
````

For a list of available language codes for fenced code blocks, see the devlang allowlist. To request new allowlist values, email DevRelIA as described in How to request metadata changes.

Contributor resources

- Contributor Guide for Docs
- Markdown basics

Next steps

Find an article, submit a change, and help improve our SQL Server content for customers.

Questions? Email sqldocsPR.

Thank you!

SQL versioning guidance and scenarios for writers

7/7/2021 • 14 minutes to read

This article explains the effects of the OPS Conceptual Versioning system to Writers, emphasizing the particular esoteric needs of the **MicrosoftDocs/sql-docs-pr** GitHub repo.

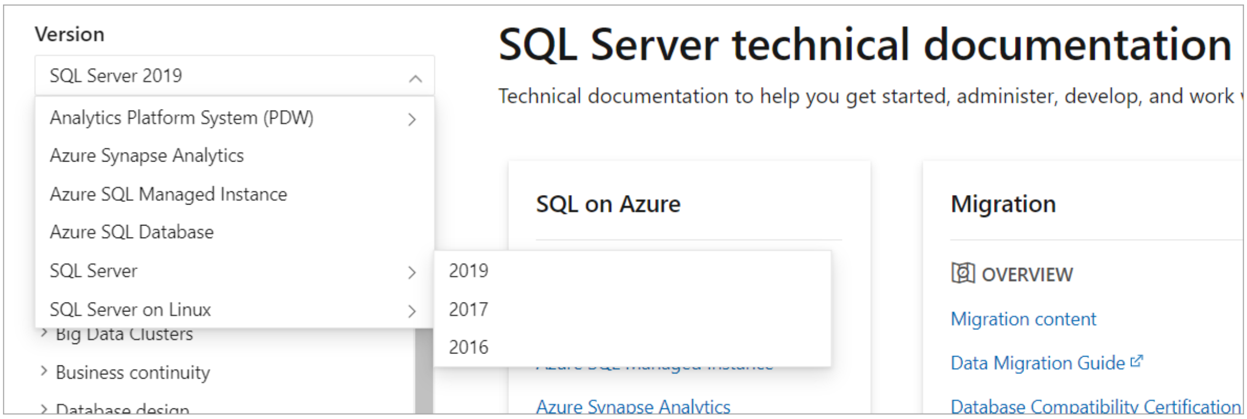
The emphasis is on the *decisions* and *actions* that Writers must take to cooperate with the Versioning system. However, you have to gain and understand basic knowledge about the Versioning system in **sql-docs-pr** before we can explain those decisions and actions.

If you want a quick overview of how we've implemented versioning or if you have an update you need to make in a single article, see the SQL versioning quickstart section.

For regular contributors, it's important to review how we implemented versioning and the special scenarios in which we use versioning. See the SQL versioning guidance section.

SQL versioning quickstart

SQL Server content has implemented versioning. Users can select a SQL Server service or product version from the left-hand TOC to scope the content that Docs displays.



You can associate SQL services and product versions with monikers. The following table explains the current monikers that you can use in SQL articles to control versioning:

AREA	MONIKER	ASSOCIATED SERVICE OR PRODUCT VERSION
All	`sqlallproducts-allversions`	Everything. No filter.
SQL Server	`sql-server-ver15`	SQL Server 2019
	`sql-server-2017`	SQL Server 2017
	`sql-server-2016`	SQL Server 2016
	`sql-server-previousversions`	SQL Server previous versions
SQL Server (Linux)	`sql-server-linux-ver15`	SQL Server 2019 for Linux

AREA	MONIKER	ASSOCIATED SERVICE OR PRODUCT VERSION
	`sql-server-linux-2017`	SQL Server 2017 for Linux
SQL Database	`azuresqldb-current`	Azure SQL Database
	`azuresqldb-mi-current`	Azure SQL Database Managed Instances
SSAS	`sql-analysis-services-2016`	Analysis Services for SQL Server 2016
SQL Server PDW	`aps-pdw-2016`	APS and PDW for SQL Server 2016-AU6
	`aps-pdw-2016-au7`	APS and PDW for SQL Server 2016-AU7
SQL DW	`azure-sqldw-latest`	Azure SQL Data Warehouse

Versioning happens at three levels:

- Folder
- Article
- Zone

Folder-level defaults

Each folder in the repo has a versioning default assigned in the **sql-docs-pr/docs/docfx.json** file. As a result, new articles are automatically versioned just by their location.

If you think there should be changes to folder defaults, contact the SQL content team. Changes are often more complex than they appear at a surface level.

Article-level metadata

When the folder default versioning doesn't work for an article, you can add a **monikerRange** metadata value to the article. The metadata value overrides the folder-level defaults. Consider the following portion of metadata:

```
author: "craigg-msft"
ms.author: "craigg"
monikerRange: ">= sql-server-linux-2017 || >= sql-server-2016 || = sqlallproducts-allversions"
---
```

In this example, the article applies to:

- Linux for SQL Server 2017 and greater: `>= sql-server-linux-2017`
- All versions of SQL Server 2016 and greater: `>= sql-server-2016`
- User selected All SQL: `= sqlallproducts-allversions`

Zone-level versioning within articles

Inside an article, you can view version-specific **zones** of text. Consider the following examples:

```
::: moniker range="=sql-server-2017"
|What's new  | Release notes  |
|---------|---------|
|[What's New in SQL Server 2017](../sql-server/what-s-new-in-sql-server-2017.md) |[SQL Server 2017 Release Notes](../sql-server/sql-server-2017-release-notes.md)|
::: moniker-end

::: moniker range="=sql-server-2016"
|What's new  | Release notes  |
|---------|---------|
|[What's New in SQL Server 2016](../sql-server/what-s-new-in-sql-server-2016.md)| [SQL Server 2016 Release Notes](../sql-server/sql-server-2016-release-notes.md)|
::: moniker-end
```

The examples use the same moniker range concept, but it applies to zones inside the article. Docs displays a different table when it's scoped to SQL Server 2017 versus SQL Server 2016. Docs customizes the user's experience.

> **TIP**
>
> For a complete example, see the markdown for the main SQL Server Documentation article. Experiment with the versioning selector on the left. Then select the **Edit** button on the right to view the markdown (just don't submit any edits).

SQL versioning guidance

The following sections go into more detail about concepts behind SQL versioning and scenarios involving versioning. It's important that all regular contributors review and learn this information.

The **sql-docs-pr** repo has the documentation content for Microsoft SQL Server 2016 and all later versions. Much of the documentation also applies to related products or services like Azure SQL Database. And before June 2018 the repo will also have the content for SQL Server 2014. Writer decisions and actions must account for the added residence of SQL 2014, sometimes.

> **NOTE**
>
> Versions 2005-2012 are available in the SQL Server previous versions documentation. But these much older versions do not participate in Versioning.

If you have read this article before, and just want a quick refresher, see the short Summary of actions near the end of this article.

SQL has complex Versioning requirements

Most other documentation repos have simpler requirements for Versioning than does our **sql-docs-pr**. The reasons include:

- Our high-level directories or folders under **sql-docs-pr/docs/** were never designed to be strict groupings of articles by the Versioning attributes. It isn't clear whether such a goal would even be realistic. But many other repos inherently have such high-level folders.

- The family of SQL Server and the different closely related products is more diverse and nuanced than the families of other repos. SQL includes:

  - On-premises.
  - Two products in the Azure cloud, one being Azure SQL Database Managed Instance.
  - Offshoot products like Azure SQL data warehouse.

- A Linux-based product.
- Several tools that might or might not quite be a direct part of the primary SQL Server product.
- Tools you can install like SSMS. They're designed so that customers never have any motivation to use an older version.

- Many articles apply only to SQL Server on-premises, and don't apply to Azure SQL Database.
- In the **sql-docs-pr** repo, the high-level folders under **docs/** were never designed to group articles by their Versioning attributes.

Prerequisites

OPS documentation

Per the rules of the internal Contributor Guide, this article contains as little redundancy as possible with the general OPS Versioning system docs. You should read the general Versioning documentation first:

- Conceptual Versioning

If you already know a little about OPS Versioning, you might learn everything else you need to know just by reading this article.

Terminology

As you read this article, you'll come across several precisely used terms. Whenever you're unsure about a term, see Terminology of OPS Versioning.

docfx.json encodes the default monikers

The following snippet of JSON code is an abbreviation of the **fileMetadata** element in our real **docs/docfx.json** file.

```
"fileMetadata": {
  "monikerRange": {

    "2014/**/*.md": "=sql-server-2014 || =sqlallproducts-allversions",

    "relational-databases/**/*.md": ">=sql-server-2016 || =sqlallproducts-allversions",

    "t-sql/**/*.md": "=azuresqldb-current || >=sql-server-2016 || =sqlallproducts-allversions"
  }
},
```

> **WARNING**
>
> By glancing at the high-level folder names in the JSON **fileMetadata**, you can see how dangerous it could be for a Writer to rename a high-level folder.

> **NOTE**
>
> Many other GitHub repos that implement OPS Versioning do not use the **fileMetadata** element. Instead, the majority rely on the **Groups** JSON element to assign default moniker ranges to high-level folders, or to their equivalent of our **docs/** folder. Our **docs/docfx.json** does use the **Groups** JSON element, but only lightly to support **fileMetadata**.

Action: Move a file

In this section, we refer to the earlier JSON that has the **fileMetadata** element from **docs/docfx.json**.

Suppose we have an article **sql-article-1.md** under the **docs/t-sql/** folder. The article has no moniker range specified within its MD file. So system assigns the moniker range we've specified in the **fileManager** JSON file of the **t-sql/** folder. The default moniker range is correct for the file. The article applies to both SQL Server on-premises and Azure SQL Database.

Then we want to move the **sql-article-1.md** file laterally, from **docs/t-sql/** to **docs/relational-databases/**. This move puts the file under a different high-level folder. In the **fileMetadata** element, its new folder has a different moniker range than its old folder has. And the new range doesn't exactly match the Versioning needs of the article. So when we move the MD file, we must add a moniker range to the metadata of the file. In this particular case, a copy of the moniker range from **t-sql/** is exactly the range we should put inside the article that we move. See the **monikerRange:** name in this example YAML metadata:

```
---
title: All about SQL article 1 | Microsoft Docs
monikerRange: =azuresqldb-current || >=sql-server-2016 || =sqlallproducts-allversions
.....More metadata here.....
---
# SQL Article 1
```

> **NOTE**
> Avoid adding a **monikerRange** into an article's YAML metadata when it would merely duplicate all the same moniker comparators that the article otherwise inherits from its high-level folder.

> **WARNING**
> For now we are ignoring an important coordination consideration that exists because of the existence of SQL Server 2014 documentation under **docs/2014/**. See the later section about SQL 2014.

Move a file lower

If you move a file from **docs/relational-databases/** further down to **docs/relational-databases/in-memory/**, there are no Versioning considerations related to monikers. But there could be considerations involving SQL 2014 under **docs/2014/**, as explained later.

Action: Add a file

Given the preceding explanation of *moving* a file, the following steps should suffice for *adding* a new file:

1. Decide under which high-level folder to store the new MD file in the **sql-docs-pr** repo.
2. Go to the **fileMetadata** element in the **docs/docfx.json** file.
3. Assess whether the default moniker range of the high-level folder exactly matches the needs of your new article.
   - If yes, then you have no more Versioning actions to take. Just add the file as usual.
   - If no, then add a **monikerRange** into the YAML metadata of the new article.

Which moniker comparators can you use?

The only valid list of moniker comparators you can choose from exists in the **Groups** element of the **docs/docfx.json** file!

Action: Rename a file

Renaming a file has Versioning considerations only if the same *path/filename.md* exists for both SQL 2016+ and SQL 2014. See the later section about SQL 2014.

Action: Remove a file

There are no Versioning considerations when you remove an obsolete MD file. SQL Server 2014 documentation under **docs/2014/** causes no considerations either.

SQL Server 2014 and Versioning complications

All the documentation for SQL Server 2014 is under **docs/2014/**. No documentation for any other product or version is under **docs/2014/**.

The only regular moniker comparator for the **2014/** folder is the equality to SQL Server 2014: = **sql-server-2014**.

> **NOTE**
>
> Ideally the set of 10,000 articles under **docs/2014/** would be entirely self-contained. However, all SQL 2014 articles that contain links to a Transact-SQL article link out to articles under **docs/t-sql/**. Also, as of May 2018, some 2014 articles link to **docs/mref/** (although when OPS Reference Versioning is fully implemented, this might change).

Paths of SQL 2014 and 2016 must correspond

About 80% of the 10,000 *path/filename.md* items under **docs/2014/** have an exact match in SQL Server 2016 documentation under **docs/**. By *match* we mean both of the following two conditions:

- The **assetGuid** from CAPS for the articles that migrated to **docs/2014/** matches the YAML **ms.assetid** in a SQL 2016 article elsewhere under **docs/**.
- From the end of their respective HTTP root folder paths in the OPS system, the rest of their HTTP addresses are identical to each other.
  - In the **sql-docs-pr** repo, for all articles under **docs/2014/**, the OPS portal at https://ops.microsoft.com/ shows that the GitHub folder **docs/2014/** maps to the HTTP address node of **sql/**.
  - For all articles elsewhere under **docs/**, the OPS portal shows that the GitHub folder **docs/** maps to the *same* HTTP address node of **sql/**.
  - For example, https://docs.microsoft.com/**sql/**relational-databases/in-memory/cool-sql-2.

These matches or *article version pairs* work in the Versioning system only because both articles have the exact same complete HTTP address as each other. Well, the one difference is their value for the URL parameter **?view=**. Here *view* means *the product and version of the article to display*. The parameter value is a moniker known to the OPS system.

Next is an informal example that shows an *article version pair* as it could look in GitHub. The two GitHub folder paths match identically where the **relational-database/** folder starts.

```
docs/     relational-databases/in-memory/cool-sql-2.md

docs/2014/relational-databases/in-memory/cool-sql-2.md
```

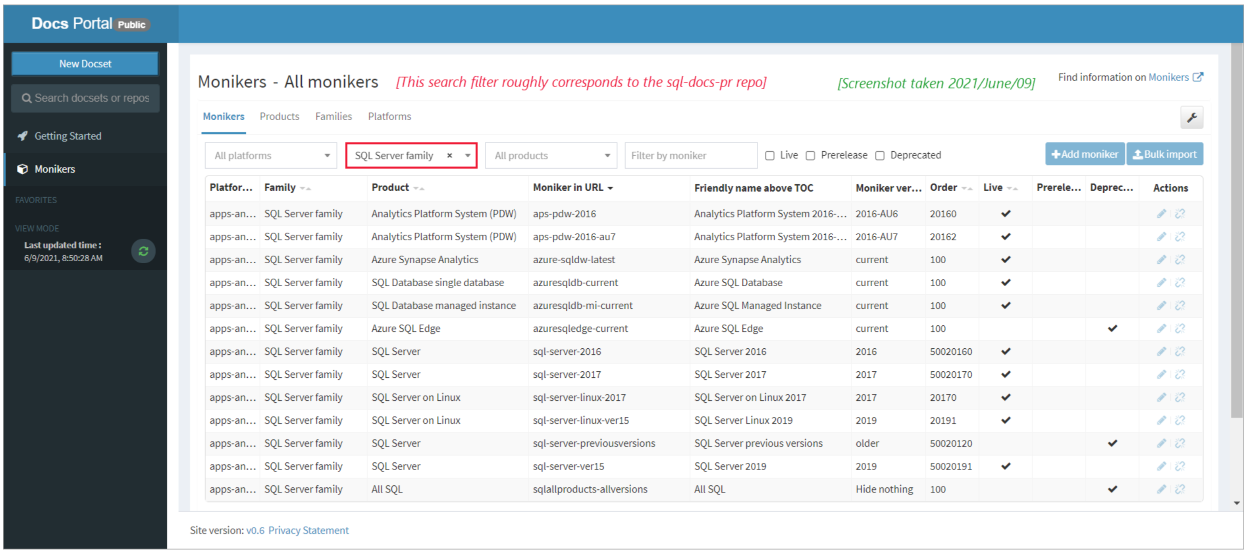
SQL 2014 migration guided by correspondence

As we migrate the **SQL14** portfolio from CAPS into repo *sql-docs-pr* in April-May 2018, we're careful to match on SQL 2014's articleGuid from CAPS to the **ms.assetid** metadata left from the earlier February 2017 migration of portfolio **SQL Server 2016**. Whenever we find a guid match, we carefully place the SQL 2014 article under **docs/2014/** in the same corresponding path where the SQL 2016 article already is elsewhere under **docs/**.

OPS portal for our packages

The only monikers that the Versioning system recognizes and honors are the monikers that we've tagged as *Live* on the Packages display in the OPS portal. The monikers that are useful to our **sql-docs-pr** repo are part of our *Family* named **SQL Server family**. Technically we of the **sql-docs-pr** repo don't officially *own* the **SQL Server family**, but in practice we should and do own it. Outsiders shouldn't be adding or changing monikers in **SQL Server family**.

Next is a screenshot of our packages in the portal.



Platfor...	Family	Product	Moniker in URL	Friendly name above TOC	Moniker ver...	Order	Live	Prerele...	Deprec...	Actions
apps-an...	SQL Server family	Analytics Platform System (PDW)	aps-pdw-2016	Analytics Platform System 2016-...	2016-AU6	20160	✔			
apps-an...	SQL Server family	Analytics Platform System (PDW)	aps-pdw-2016-au7	Analytics Platform System 2016-...	2016-AU7	20162	✔			
apps-an...	SQL Server family	Azure Synapse Analytics	azure-sqldw-latest	Azure Synapse Analytics	current	100	✔			
apps-an...	SQL Server family	SQL Database single database	azuresqldb-current	Azure SQL Database	current	100	✔			
apps-an...	SQL Server family	SQL Database managed instance	azuresqldb-mi-current	Azure SQL Managed Instance	current	100	✔			
apps-an...	SQL Server family	Azure SQL Edge	azuresqledge-current	Azure SQL Edge	current	100			✔	
apps-an...	SQL Server family	SQL Server	sql-server-2016	SQL Server 2016	2016	50020160	✔			
apps-an...	SQL Server family	SQL Server	sql-server-2017	SQL Server 2017	2017	50020170	✔			
apps-an...	SQL Server family	SQL Server on Linux	sql-server-linux-2017	SQL Server on Linux 2017	2017	20170	✔			
apps-an...	SQL Server family	SQL Server on Linux	sql-server-linux-ver15	SQL Server Linux 2019	2019	20191	✔			
apps-an...	SQL Server family	SQL Server	sql-server-previousversions	SQL Server previous versions	older	50020120			✔	
apps-an...	SQL Server family	SQL Server	sql-server-ver15	SQL Server 2019	2019	50020191	✔			
apps-an...	SQL Server family	All SQL	sqlallproducts-allversions	All SQL	Hide nothing	100			✔	

Beyond our family

We own monikers beyond just the ones in the **SQL Server family**. Other monikers we own and use include:

- azure-sqldw-latest
- aps-pdw-2016

When you need to add a moniker range in the YAML of an article's MD file, you're restricted in which moniker comparators you can use. You can find the list of moniker comparators that you can choose from in the **Groups** element of the **docs/docfx.json** file.

Not involving editions in our Versioning

For the **sql-docs-pr** repo, we designed our monikers to provide filtering at the level of *product* and *version*. We found those two dimensions complicated enough. We believe it's unrealistic to reach for the more granular concept of *edition* which are the differences between the on-premises SQL Server editions:

- *Developer*
- *Standard*
- *Enterprise*

Most or all other teams at Microsoft have come to the same conclusion.

sqlallproducts-allversions is a special moniker

After the OPS Versioning implementation, each article webpage shows a drop-down box containing clickable lines of product and version combinations. When the customer selects a different line, Docs might instantly reduce the TOC to display only the subset of articles that have a moniker range that includes the chosen product and version line. The Versioning requirements for the **sql-docs-pr** repo are complex enough that sometimes the customer could become frustrated by their inability to make the whole TOC reappear (it's a long explanation).

To mitigate this ongoing frustration of a much hidden TOC, we added the special moniker *sqlallproducts-allversions* into our family of monikers. In the `sql-docs-pr` repo, we always include the moniker comparator of `= sqlallproducts-allversions` in every moniker range, at all precedence levels! Most other repos have no similar moniker, but at least one other repo does.

> **WARNING**
> All our moniker ranges, at all precedence levels, must include the moniker comparator of `= sqlallproducts-allversions`!

Live examples of precedence levels

How to find which articles contain a YAML moniker range

Find by using a cmd.exe window

```
cd sql-docs-pr
cd docs
findstr /s /l /i /n /c:"monikerRange" *.md
```

Find by using PowerShell

The following code example shows long syntax commented out, and then the equivalent short syntax.

```
# Get-ChildItem -Recurse:$true -Include "*.md" | Select-String "monikerRange";
gci -r -i "*.md" | sls "monikerRange";
```

Find by using a bash.exe window

```
grep 'monikerRange' -E -r *
```

How to find which articles contain a ::: zone

By using a `cmd.exe` window, you can find articles that contain a Versioning *zone*.

```
findstr /s /l /i /n /c:":::" *.md
```

Terminology of OPS Versioning

- *Conceptual Versioning:* Is distinct from *Reference Versioning*, which applies to autogenerated reference content.
- *moniker* The unique ID or unique name for each Package row in the OPS portal. A moniker is a combination of *product* and *version*, for example `sql-server-2017`. A moniker of a *shippable unit*.
  - By itself, a moniker doesn't include any relational operator or conjoining operator.
- *Moniker-Friendly Name:* A seemingly irrelevant column of values in the OPS portal for Packages. We care about the monikers as they appear in the column *Moniker Name*.
- *moniker comparator:* For example, `>= sql-server-2016`. A moniker comparator is the combination of a relational operator and moniker. It's equivalent to saying a node in a moniker range. See *moniker range*.
- *moniker range:* A string of one or more moniker comparators, conjoined by a conjoining operator.

- For example, the moniker range of >= **sql-server-2016 || = azuresqldb-current** has two *moniker comparators*. We've conjoined the two comparators by the conjoining operator OR.

- *operator:* There are two kinds of operators used with monikers to form ranges. Consider the example *moniker comparator* of >= **sql-server-2016 || = azuresqldb-current**:
  - *relational operator:* The >=, meaning later than or equal to. Absence defaults to =.
  - *conjoining operator:* The **||**, meaning boolean OR. Absence defaults to AND. The explicit syntax **&&** for AND is invalid. AND is rarely used.

- *OPS:* Open Publishing System of our Microsoft group named *Content & Learning*.

- *OPS portal:* https://ops.microsoft.com/

- *precedence hierarchy:* You can specify moniker ranges at increasingly granular levels. Levels that are more granular take precedence over levels that are less granular. Only one precedence level applies in any given scenario. There's *no additive effect* from the less granular levels when a more granular level takes precedence. Only the less granular moniker range applies.
  - *Folder-based:* The least granular level. It's specified in the **docfx.json** file.
    - Unless it's explicitly set up otherwise, the folder moniker range cascades down to all its child and other descendant folders in GitHub.
  - *Article YAML-based:* The metadata value **monikerRange** in the YAML at the top inside at article's MD file.
  - *Zone-based:* A ::: delimited section within the body of an article. It's the most granular level.

Summary of actions

- Ensure that any two sister articles, where one exists under **docs/2014/** and the other exists elsewhere under **docs/**, remain in equivalent corresponding *path/filename.md* values.

- Sometimes an article comes to exist under a high-level folder whose moniker range from docfx.json doesn't exactly match the Versioning needs of the article. In such cases, add a **monikerRange** vale to the YAML metadata inside the article's MD file.

- If one section of your article needs a different moniker range, use the ::: syntax.

Related links

- Versioning & Monikers

Update documentation for a servicing update

6/24/2021 • 3 minutes to read

This article explains how to submit a documentation update for a servicing update in the SQL Server repository, **MicrosoftDocs/sql-docs-pr**. A servicing update is either a cumulative update (CU) or service pack (SP).

The steps in the article demonstrate how to submit a pull request to a release branch with your browser.

Before you start

To complete these tasks, you need a GitHub account belongs to both the **Microsoft** and the **MicrosoftDocs** organizations. To configure your account, follow the instructions in GitHub account setup.

Identify your release branch

Update content for a servicing update in a release branch. When the product group releases the servicing update to general availability (GA), the SQL Server content team merges the release branch into the default branch. The updated content becomes public at the next normal publishing cycle. See Pull request review and publishing schedules.

> **NOTE**
>
> At this time, the default branch for sql-docs-pr repository is `master`. Eventually the name of the default branch will be `main`.

To update the documentation for a servicing update, locate the appropriate branch in GitHub. The sql-docs repository has servicing branches for CUs and SPs.

- SQL Server 2019: sql-docs-pr/release-2019-cu
- SQL Server 2017: sql-docs-pr/release-2017-cu
- SQL Server 2016: sql-docs-pr/release-2016-cu

For example, to submit an update to the machine learning (ML) cabinet (CAB) file for the next SQL Server 2019 CU, select the link above for SQL Server 2019.

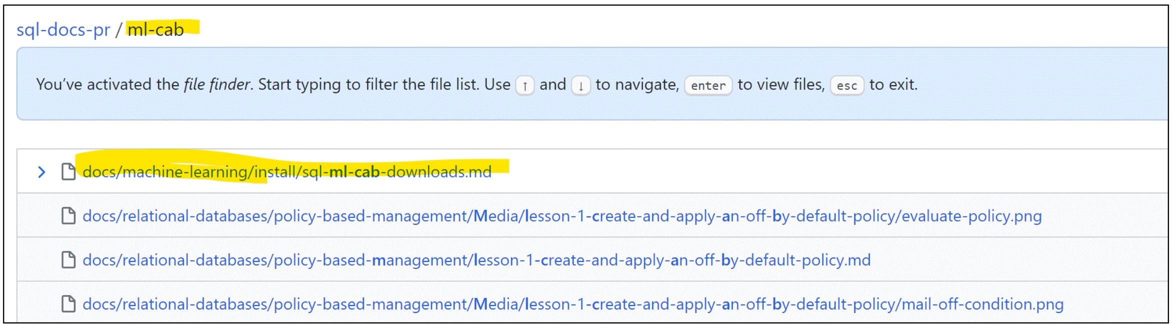
> **NOTE**
>
> **sql-docs-pr** has release branches for the most frequently updated products. The label **-cu** on the release branch name is applied because most updates are CUs. If your update is to a service pack, or other type of release, contact the SQL Server content developers for information about the appropriate release branch.
>
> Use these branches for whatever the next CU will be for a version. The branches publish when a CU releases. Stage the update in the appropriate branch, so that it publishes in sync with the update release to GA.
>
> Do not use these branches for a CU that is further in the future than next. All changes in these branches becomes public on the day the CU releases.

Locate the file you want to update

To locate the file in the repository on GitHub:

1. Verify that you are in the correct branch.
2. Select **Go to file**.

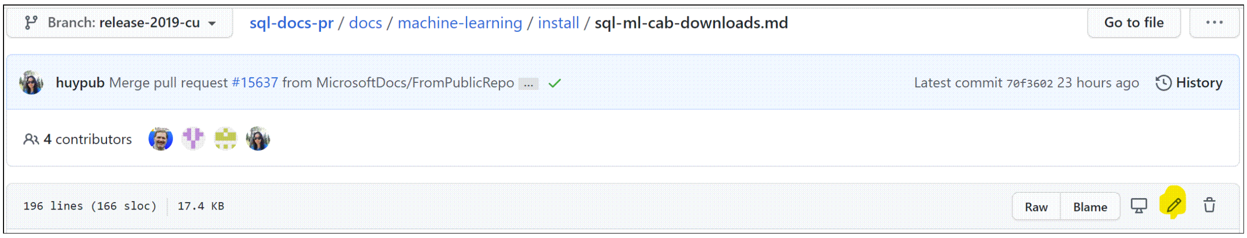
3. Type part of the file name to locate the file you want to update in this branch. For example, to locate *sql-ml-cab-downloads.md*, type `ml-cab`.



4. Select the file.

Update the file

In the GitHub page for the file, locate the pencil icon, and select it to edit the file.

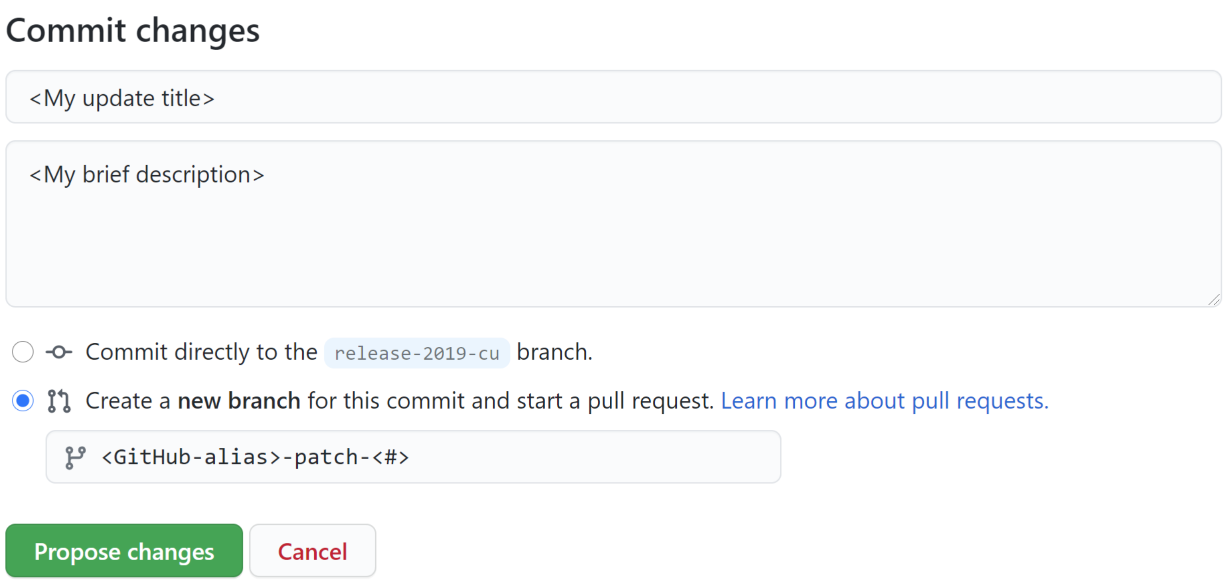


In the browser, GitHub provides an interface for you to update the content of the article in Markdown. For an introduction to Markdown, see Docs Markdown reference.

Submit the pull request

After you update the file, go to the bottom of the page in GitHub. Under **Commit Changes**, type a brief description of your update. If you need more space, use the second field.

Choose, **Create a new branch for this commit and start a pull request**.



Select **Propose changes**.

Check the pull request

After you submit the pull request, the open publishing system (OPS) validates the build. This process normally takes about 10 minutes. After validation, you can preview your updated content from the links in GitHub. The preview links are in the pull request under **Conversation**.

If your content validates correctly, you can sign-off on the content.

> **TIP**
>
> You can find all of your pull requests at https://github.com/pulls.

Sign-off and publish

To sign-off on a pull request, wait for it to validate. In the pull request **Conversation**, add `#sign-off` as a comment. After you sign-off, the open publishing system notifies the reviewers to review and merge the content. OPS automatically merges small changes. Larger changes require a human reviewer.

An automated workflow routes the pull request to the reviewers. The reviewer will either merge the update or block the update. If the update is blocked, the reviewer will add a comment to explain the blocking violations. For more information, see Pull request submission best practices.

If you have any questions, email the SQL content team (sqldocs).

Related links

Get started contributing to SQL Documentation

ILDC-specific publishing and pull request procedures

5/10/2021 • 2 minutes to read

Pull request review

To view the standard Redmond-based pull request review schedule translated into ILDC hours, see the PR review and publishing schedules article.

To facilitate the workflow on Sundays in Israel and to compensate for the usual 10-hour time differential, IDLC team members can request local pull request review.

The designated ILDC pull request reviewers are authorized to merge content in the following repositories:

- azure-docs-pr
- cloudappsecurity-docs-pr
- atadocs-pr

When the ILDC pull request reviewer merges a PR, they assign the **ILDC-PR** label in GitHub. The Redmond-based PR review team will review these merged PRs daily and provide any additional feedback necessary directly to the author via a comment in the merged PR with an @mention to the author requesting follow up.

Publishing

To view the standard Redmond-based publishing schedule translated into ILDC hours, see the PR review and publishing schedules article.

The ILDC team determines when publishing occurs for the following repositories because they are wholly managed by the content team:

- cloudappsecurity-docs-pr
- atadocs-pr

If the ILDC team needs to publish Azure content from the azure-docs-pr repository on Sundays or other days of the week, the team has a couple options that allow publishing to occur without pushing the entire contents of the Azure-docs-pr master branch live. These options preserve the ability of US-based writers to correctly manage their planned release times.

- **Sunday publishing branch**: There is an ongoing release branch named **release-ildc-sunday** that members of the ILDC team can use for all content in the Azure repo that they want to release during the Sunday business day in Israel. The designated ILDC pull request reviewers are authorized to publish on Sundays using these steps. This process ensures that the commits for Sunday publishing go into live and master without releasing any other pending content in master:

  1. Create a pull request from the **release-ildc-sunday** branch directly to **live** and merge after a successful build.
  2. Create a pull request from the **release-ildc-sunday** branch directly to **master** and merge after a successful build.

- **Hotfix publishing**: The designated ILDC pull request reviewers can follow the standard process for hotfix publishing to publish content as needed during the ILDC business day.

If necessary, the ILDC team can use the Sunday publishing branch model with release branches if an ILDC release needs to occur during a US holiday or to target a specific event that occurs during ILDC business hours.

Content freezes

The ILDC team must respect content publishing freezes that occur as part of a planned release. The team should work with Redmond-based release management if an exception to the publishing freeze is required. Contact the designated release manager for the event to work out a solution.

Get started contributing to Visual Studio documentation

7/15/2021 • 2 minutes to read

This article provides guidance on how Microsoft employees can contribute to Visual Studio documentation published at https://docs.microsoft.com/visualstudio/. Visual Studio and related documentation is managed in the following GitHub repositories:

- visualstudio-docs-pr: Visual Studio content
- visualstudio-docs-dotnet-api-pr: Visual Studio SDK reference content
- intellicode-pr: IntelliCode content
- liveshare-docs: LiveShare content

This guidance applies to the visualstudio-docs-pr repo.

A repo with the suffix `-pr` is a private repo and every private repo will have also have a public repo that is kept in sync with it. The public repo is intended for use by external contributors. For internal Microsoft employees, we recommend using the private repo.

> **NOTE**
> Visual Studio Code documentation is not on the docs platform. For authoring information, see Visual Studio Code documentation on GitHub.

General tasks

If you need to make a minor change to an existing article, see Quickly edit an existing article. For all other edits, use the following resources in the Contributor's Guide:

TASK	CONTRIBUTOR'S GUIDE RESOURCES
Setup a local repository for larger or frequent edits	Sign up for a GitHub account Install Git and VS Code Set up a Git repository locally
Submit larger or frequent edits	Create working branches and submit pull requests Edit content in Visual Studio Code
Follow content standards	Write an overview Write a quickstart Write a tutorial Write a sample Edit a TOC
Improve your writing	Follow writing principles Use Acrolinx Review the Microsoft style guide

TIP

If you are adding new content or making significant changes, please include a member of the content team on this work to avoid duplication and to align with ongoing content strategies.

Visual Studio tasks

In addition to the general writing tasks in the previous section, there are writing tasks and resources specific to Visual Studio documentation.

TASK	VISUAL STUDIO DOCS RESOURCES
Ask a question	Email the Visual Studio content team
View current work	View the Visual Studio and VS Code dashboard
See staged content	Visit the review site and select a branch
Visual Studio writing tasks	Edit metadata in Visual Studio articles Manage versioning Work with Visual Studio release branches

Next steps

For more information, see the Contributor's Guide.

Visual Studio docs guidance

7/15/2021 • 7 minutes to read

This article provides guidance for Microsoft employees who contribute to Visual Studio documentation in **visualstudio-docs-pr**.

How to contribute

You can contribute to directly to Visual Studio docs either by editing in your browser or by working locally with VS Code and Git Bash. For more information, see Get started contributing to Visual Studio Documentation.

> **IMPORTANT**
>
> If you need to make changes for a future release, please use an appropriate release branch and follow the steps on working with release branches in this article.

Author publishing process

Authors refer to doc owners, such as content writers, or members of the product group. Authors should follow the publishing process outlined in this section. Writers that do not fit into these two categories should follow the contributor guidance.

To submit your documentation change, open a GitHub pull request (PR) in the private repo. For more information, see How to create a PR. Please note the following:

- When you are viewing a page on the live site you want to edit, you can use the **Edit** button to switch to the page in GitHub, but this takes you to the public repo. You can easily switch to the private repo from there by editing the URL to add `-pr` to the repo name.
- After you complete your changes, you must add #sign-off in the comments of the PR.
- The PR is reviewed for standards before merging. You will be asked to fix any issues.

After your changes are merged, they are ready to publish. Twice a day, the **master** branches of **visualstudio-docs-pr** and are pushed to the public repositories. This publishes the content live. Publishing happens at roughly 10:30am and 4:30pm PST.

> **NOTE**
>
> If you are working in a release branch, then the content is not published until that release branch is merged into **master**.

Email contact

Use the vs-content email alias to discuss technical content and PRs related to **visualstudio-docs-pr** and **visualstudio-docs**.

Visual Studio workitem tracking

To see what is happening in Visual Studio content, access the Visual Studio and VS Code Azure Boards Dashboard.

If you have a new content idea or need to partner with the content team in any way, contact the Visual Studio

Content Team.

Visual Studio docs guidance

The following section provides some additional guidance on working in the **visualstudio-docs-pr** repository.

The visualstudio-docs-pr repository uses several standard folders to organize the content.

FOLDER	DESCRIPTION
docs/	Contains all published Visual Studio content. Subfolders logically organize different areas of the content.
docs/includes/	Contains include files. These files are blocks of content that can be included in one or more other topics.
./media/	Each folder can have one **media** subfolder for article images. The **media** folder in turn has subfolders with the same name as the topics that the image appears in. Images should be .png files with all lower-case letters and no spaces.
toc.yml	A table-of-contents file, in the newer YAML format. Each subfolder has the option of using one toc.yml file.

Contributor guidance

Product group PMs or content developers approve all updates in the private repository. Contributors who are not in the product group may add PMs or content developers as reviewers, but they may not sign off on new content or updates. This policy ensures that content aligns with product group intent and supportability.

Procedure

Once a PR has been submitted, you must:

1. Check the Acrolinx scorecard and ensure to implement the suggested changes.
2. Review the staged content to ensure formatting is okay and your changes look good. Do this by selecting the "View" link in the **Validation status: Passed** comment under **Preview URL** within your PR.
3. Check the metadata for who the ms.author of the article is, and email them directly with the link your PR.

Guidelines

These are general guidelines that all authors must follow, but that are not explicitly stated elsewhere.

- When providing a link to content, content on Docs takes priority. Do not provide a link to a blog if a Docs article exists. If providing a link to a blog, the blog should be vetted by the article author for technical accuracy. Ideally, the content of the blog should be migrated to Docs, but the author needs to make that call.

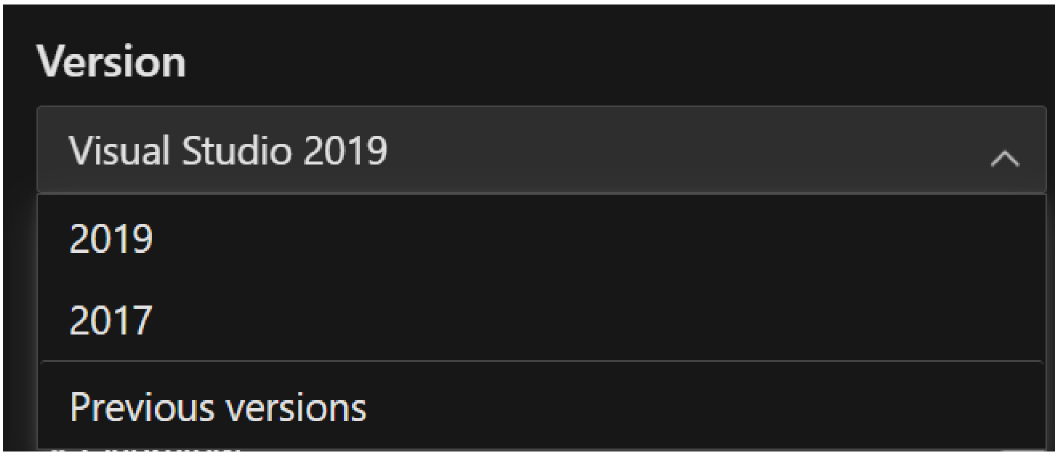
Article metadata

Each article has a metadata block at the top. For detailed metadata information, see Metadata guidance. In the near future, we will post more specific metadata guidance for Visual Studio topics here. For now, use metadata in a similar way as existing Visual Studio articles in your area.

Versioning

Visual Studio content has implemented versioning. Users can select a Visual Studio service or product version from the left-hand TOC to scope the content that is displayed.

In our documentation, we create content that applies to several recent versions of Visual Studio. Most of the time, that content is uniform across versions. But sometimes, it isn't. To make sure that folks can see content for the version of Visual Studio that they are using, we include a version drop-down control on the left side of the page so that they can do so.



To provide this versioning experience, we use monikers in Markdown to tag content that applies to a specific version. For example, we currently publish content for Visual Studio 2017, Visual Studio 2019, and Visual Studio 2022 (Preview as of this writing). So, we use the following monikers to offset sections or pages for specific versions.

- vs-2017
- vs-2019
- vs-2022

Here's an example of how we use them to set off text:

```
::: moniker range="vs-2017"

If you want to check that your unit test is working, chooseTest>Run>All Testsfrom the menu bar. A window
calledTest Exploreropens, and you should see that theTestGetCurrentDatetest passes.

::: moniker-end

::: moniker range="vs-2019"

If you want to check that your unit test is working, chooseTest>Run All Testsfrom the menu bar. A window
calledTest Exploreropens, and you should see that theTestGetCurrentDatetest passes.

::: moniker-end
```

When using monikers for product changes that are expected to persist to future versions, use the moniker range syntax that specifies versions greater than or equal to the current, as shown in the following example. Otherwise, the content in the moniker block will not be included in the docs for future releases.

```
::: moniker range=">=vs-2022"
```

NOTE

Monikers sometimes don't work well in lists, such as numbered lists or bulleted lists, or in a table. In many cases, it's best to repeat an entire section in a moniker tag rather than to try to disambiguate entries in a list or in a table.

Branches

Visual Studio content uses the following branches for releasing content. The instructions in this guide use the **master** branch, which is for immediate release. But there are also release branches for delayed releases.

BRANCH	DESCRIPTION
master	Changes publish twice a day from this branch.
release-16.xx	Changes for the next update of Visual Studio 2019.
release-dev17-previewN	Changes for previews of Visual Studio 2022 aka "Dev17".

> **NOTE**
>
> If you are working on documentation for a preview release, be sure to understand the general policies for preview content as well as the Visual Studio guidance for preview content: Add preview content to Visual Studio docs.

Work in a release branch

Use the following steps to work in a release branch like **release-16.xx** or **release-dev17-previewN**.

1. If you have not locally cloned the **visualstudio-docs-pr** repository and installed Git Bash and VS Code, see Set up a local repository for larger or frequent edits.

2. Follow the steps in Check out a release branch to check out a working branch for your release branch. For example, if you want to make changes to Visual Studio Dev17, you could run the following command to create a working branch **mychanges** for **release-dev17-preview2**:

```
cd /visualstudio-docs-pr/docs
git checkout -B mychanges upstream/release-dev17-preview2
git pull upstream release-dev17-preview2
```

3. Make all of your changes and then Commit and push your working branch.

4. Then create a PR for your changes. It helps to add `@githubusername` to the comments for any content lead or PM associated with the change to facilitate awareness and collaboration.

   Caution

   You must ensure that your PR pulls your changes into the corresponding release branch, such as **release-dev17-previewN**, and not **master**. If you accidentally pull your changes into **master**, all of the pre-release content will go live (not just your PR). Having a lot of extra commits that are not associated with your change is a clue that you have made this mistake.

If you have any questions or need any help on how to work with release branches, email the repo admin for visualstudio-docs-pr.

Contributor resources

- Contributor Guide for Docs
- Markdown basics

Next steps

Find an article, submit a change, and help improve our Visual Studio content for customers.

Questions? Email the repo admin for visualstudio-docs-pr.

Thank you!

Principles for publishing preview content for Visual Studio and VS Code on docs.microsoft.com

5/13/2021 • 4 minutes to read

As a rule, the Visual Studio & VS Code content team does not author preview content on docs.microsoft.com or code.visualstudio.com/docs. Preview content could be content associated with a specific preview release, or with an insider build, or an a/b test or experimental feature that only some users see.

Have questions? Reach out to: vs-content@microsoft.com.

Why we don’t publish preview content

We don't publish preview content because it:

- Changes more often. Product churn can result in costly doc churn, still too unstable for scarce writer resources
- Usually has a small audience, and therefore is lower impact and lower priority
- Can create a confusing experience to readers, for example if the UI or the available features don't match the current released version, or if the preview content interwoven into the existing content would create confusion
- Requires tracking so that we can remove preview text and caveats after the feature or service is no longer in preview

Where you might publish instead

When contributors want to cover preview scenarios, we recommend some or all of these options:

- **Option: Publish a blog post** Make a blog post with links or other info about how to join the preview. Blogs are useful when there is only a small amount of content to publish and it can fit in a single post (or maybe a series of a few posts). Also useful if the feature is simple to use and discoverable and therefore doesn’t need advanced documentation. You might want to link from related content on docs.microsoft.com to the blog post about the preview. We can convert blogs to official docs later when the release code is more stable.
- **Option: Publish on visualstudio.microsoft.com (VSCOM)** If the goal is to share value prop + high-level information, not technical details, consider the product site. Also, good when the feature is simple & discoverable. See example.
- **Option: Create a PDF to send to preview participants** For a limited, private preview where the information about the product or feature is not public, send a pdf (or use option 4). Also useful if there are security requirements, for example, the content must be watermarked, must not allow copying, etc. We can convert this content to official docs later when the release code is more stable.
- **Option: Publish on a separate (private) GitHub repo** When the engineering team already has a GitHub repo for source code, and especially if the project is open source, the repo can be a good solution. Also useful if there are security requirements, for example, the content must be kept in a private repo. We can convert this content to official docs later when the release code is more stable.

Exceptions

In some cases, however, we do publish preview content on docs.microsoft.com.

Exception 1

Preview release notes from the release team are permitted.

Exception 2

Updates to the build numbers page are permitted.

Exception 3

If the content does not fit into any of the previous options and all the following are true, then publishing on docs.microsoft.com is permitted.

- The preview is public. (See What can I publish on docs.microsoft.com page and in the Preview content page in this guide)
- The content aligns with a high priority strategy for the product group
- The product team expects to have many users and many pageviews for this preview feature, or a small set of strategic users
- The product group must be willing to do the following with advisory input from the VS/VS Code writing team:
    1. Own the repo (if required) and the content
    2. Go through the docs onboarding training
    3. Write, publish, and maintain the content
    4. Update the content for every change and churn of the preview feature, such as screenshots for UI changes, code snippet updates for API churn, etc.
    5. Resolve GitHub issues on the content. While you can disable GitHub issues, we usually want feedback during a preview, so we don't recommend turning off the docs feedback control.

Formatting and structure for exceptions

If you publish have an exception and can publish preview content on docs.microsoft.com, you must clearly let the reader know that this content is for a feature or service that is not yet available to everyone.

Best practices include:

- Append "(Preview)" to the H1 or H2 for a small feature that doesn't require a page
- Note any requirements for having a preview version installed in prerequisites or procedures as needed
- Include text near the top of the page, above the fold (or near the H2 preview text if it's a small feature), to let users know the feature is a preview, or requires a Preview version. The text could be something like: *"This feature is currently in preview. Previews are made available to you on the condition that you agree to the supplemental terms of use. Some aspects of this feature may change prior to general availability (GA)."*
- You might want to add a blurb about the preview feature and content to the What's New or Release Notes pages, to generate energy/buzz about the preview. In some cases, VS Code has included info about a preview for several months in a row in the release notes to build excitement. The VS Code release notes then indicate how to get the preview feature, either in the Insiders release or in the "Stable" release but requiring a setting switch. In either the VS What's New page or the VS Code Release Notes page, work with the page owner to determine if the preview info is appropriate.
- For Visual Studio preview releases, refer to the preview release name in this format: "Visual Studio 2019

version 16.7 Preview 4 or later." Note: It's ok to shorten to "16.x Preview Y."

Contribute to Xamarin documentation

6/24/2021 • 2 minutes to read

Xamarin documentation is available publicly at:

- https://docs.microsoft.com/xamarin
- https://github.com/MicrosoftDocs/xamarin-docs

For staging and internal contributions, visit:

- https://review.docs.microsoft.com/xamarin?branch=master
- https://github.com/MicrosoftDocs/xamarin-docs-pr

If you have a question about contributing, email the Xamarin docs team alias.

Article metadata

The Xamarin docfx.json defines default values for most required metadata (based on directory structure), such as:

- `ms.prod: xamarin`
- `ms.technology: xamarin-forms | xamarin-android | xamarin-ios | xamarin-mac | xamarin.essentials | xamarin-skiasharp | xamarin-crossplatform`
- `ms.topic: conceptual`
- `searchScope: Xamarin | Xamarin.Android | Xamarin.Forms | Xamarin.iOS | Xamarin.Mac | SkiaSharp`

These defaults mean the minimum required metadata block at the top of new Markdown files should be:

```
---
title: "Xamarin page title"
description: "Page description is required"
author: github-alias
ms.author: ms-alias
ms.date: 01/31/2020
---
```

Don't forget to update author info for new or significantly changed pages. Similarly, the date should only be updated if the article is considered *completely up to date* (don't update the date for a minor typo fix). Date format is mm/dd/yyyy.

Pull requests

Xamarin repos have PRMerger installed. Review your changes carefully, and ask for a review if appropriate. Use PRMerger commands like `#sign-off` when your PR is complete and ready for publishing.

Screenshots

Ensure screenshots are clear, readable, and include accessible alt text. Trim extraneous information from a screenshot, and if required add a higher-resolution lightboxed version.

Example of a regular image (less than 800 pixels wide) with alt text:

`![Create a new project.](images/win-2019.png)`

Lightbox example with a small (fewer than 800 pixels wide) image as well as a larger image to zoom in:

`[![Create a new project.](images/win-2019-sml.png)](images/win-2019-lg.png#lightbox)`

Tabs usage

Tabs are used to let customers view information specific to one or more platforms they are targeting: **Android**, **iOS**, or **UWP**.

You can see an example on this Xamarin.Essentials page (source).

```
# [Android](#tab/android)
Android content
# [iOS](#tab/ios)
iOS content
# [UWP](#tab/uwp)
Windows content
--------------
```

Pivot usage

Pivots are used to let customers switch between **Visual Studio 2019** and **Visual Studio for Mac**. The pivot will autoselect based on the operating system the customer is browsing on.

You can see an example pivot on this quickstart page (source).

```
::: zone pivot="windows"
Windows content
::: zone-end
::: zone pivot="macos"
Mac content
::: zone-end
```

Pivots are defined in zone-pivot-groups.yml – this file should not be updated unless we ship new versions of Visual Studio.

Product versions

Xamarin docs target the current stable version.

There is no support for monikers that let customers choose a version.

Release notes

Xamarin release notes are in a separate repo:

- https://github.com/MicrosoftDocs/xamarin-engineering-docs-pr

API reference

Xamarin docs use the xref syntax extensively for build-time-checked reference links, for example, `[Effect](xref:Xamarin.Forms.Effect)`.

You can test an xref in any browser by changing the `uid` in this query: `https://xref.docs.microsoft.com/query?uid=Xamarin.Forms.Effect`.

Reference source

Xamarin reference content, which drives the docs site and IntelliSense, is stored in the following repos:

- Xamarin.Forms reference
- Xamarin.Android reference from Google
- Xamarin.iOS and Mac reference
- SkiaSharp reference

Samples

Xamarin code samples can be browsed on docs.microsoft.com/samples. The source code is available in the following repos:

- Xamarin.Forms samples
- Xamarin.Android samples
- Xamarin.iOS samples
- Xamarin.Mac samples

Meriwether diff reporting tool guide

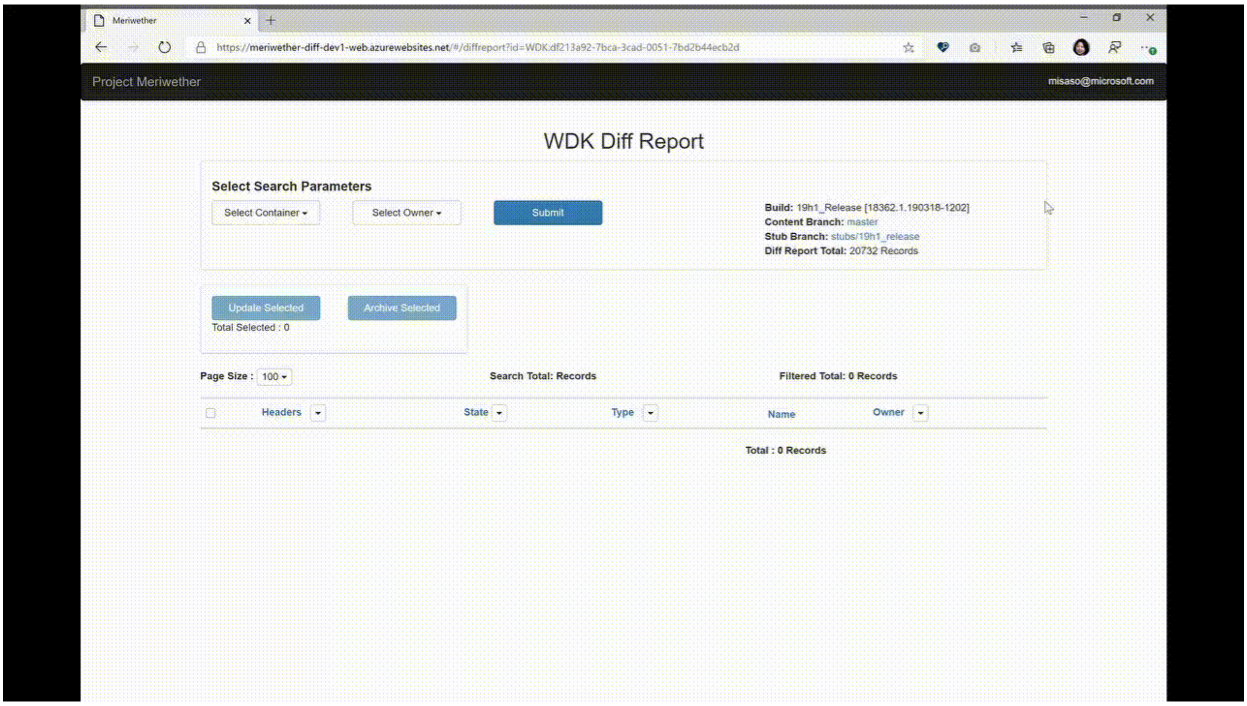
6/24/2021 • 3 minutes to read

The Meriwether diff reporting tool at https://aka.ms/meriwether is an internal tool that helps with Windows API content authoring. Using the tool, authors can:

- View APIs that are new, modified, or deleted, in a given release
- View APIs that are undocumented in previous releases
- View and select APIs by author, header file, or namespace
- Filter APIs by New, Changed, Delete, or Undocumented
- Put APIs in Hold state for further filtering
- Obtain raw markdown stubs for authoring

This tool supports the following API reference docs:

- *Windows Runtime*
- *Windows UI*
- *Win32 and COM APIs*
- *Windows Driver Kit DDIs*



Using the tool

No setup is required to use the diff reporting tool.

To use the tool, navigate to https://aka.ms/meriwether, and select the report you wish to work with. The title of the report indicates the name of the release, to pull diff information on. For example, WDK_Iron will pull up a diff report for the Iron release of Windows WDK DDI, and is a comparison of the stub branch of the Iron release, against the authoring branch of the Iron release. Note that the report does not diff against branches in different releases.

> **NOTE**
>
> The report will take a few seconds to load.

Each report contains a link to the content branch, a link to the stub branch, the version and build number, and two report totals. **Diff Report Total** represents the total changes, including New, Undocumented, and Changed APIs for the selected release. **Diff in Release** represents the total number of New APIs in the selected release, and not in the authoring branch for that release.

Under **Search Options**, select up to 10 headers from the **Select Container** drop-down text box, or select an alias from the **Select Owner** drop-down text box. To view API that are Undocumented in all previous releases, select the box next to **Include Undocumented**.

Once you've made your selection(s), choose **Submit**, and the report will display a list of APIs organized by headings: Headers, State, Type, Name, and Owner. You may filter these heading by clicking the drop-down arrow next to each heading. You can also order the APIs by a heading. API States are either New, Changed, Deleted, or *Undocumented* - if the **Include Undocumented** box was checked.

The following describes the API States in the report:

STATE	DESCRIPTION
New	APIs with unique IDs appear in the selected version stub branch, but NOT in the content authoring branch, and NOT in the previous version stub branch.
Undocumented	APIs with unique IDs appear in the stub branch of all versions prior to the selected version, but NOT in the content authoring branch, and NOT in the selected version's stub branch.
Changed	APIs with unique IDs appear in both the selected version's stub branch, and the authoring branch, but the keys differ.
Delete	Delete means the UID exists in the content branch but not in the stub branch.

Click each API name to view the API members.

The **Search Total** displays a total from the **Search Options** filter box, and the **Filtered Total** displays a total of the APIs filtered by the headings (Headers, State, Type, Name, Owner).

Get topic stubs

Topic stubs are autogenerated for each new/changed API. To get these stubs, select the checkbox next to the API, and choose **Update Selected**. A branch containing the stubs will be created for you in the content repo. To find your branch, navigate to the content repo to which you selected the report for, and on the left navigation, choose **Branches**, then choose **All**. The branch will be in the form of:

```
<alias>/diffrequest-on-<stub branch>-<yyyymmdd>-<hhmmss>
```

Note: the new branch won't be listed under **Mine** because it was created by the tool.

You can view the new or changed files by viewing the latest commit history of the new branch.

Mark as hold

You can use the **Toggle Hold** button to set state for selected APIs to Hold. This state can be used for APIs that you want to set aside for now and not have appear in the diff report.

Notes

- You can only select a maximum of 10 headers at a time.
- User state of the tool doesn't persist, so clicking the back button of the browser will navigate back to the start page.
- At this time, the branches created by the diff report can't be deleted.
- Only 2 reports are available for each API reference. For example, reports for the current release and the next release are the only ones available for WDK DDI.
- Under **Search Options**, a user may only select headers OR owners, but not both. Selections do not filter the other options.

Issues

File bugs or issues by using this bug template.

How to use Microsoft Docs Documentation

6/24/2021 • 6 minutes to read

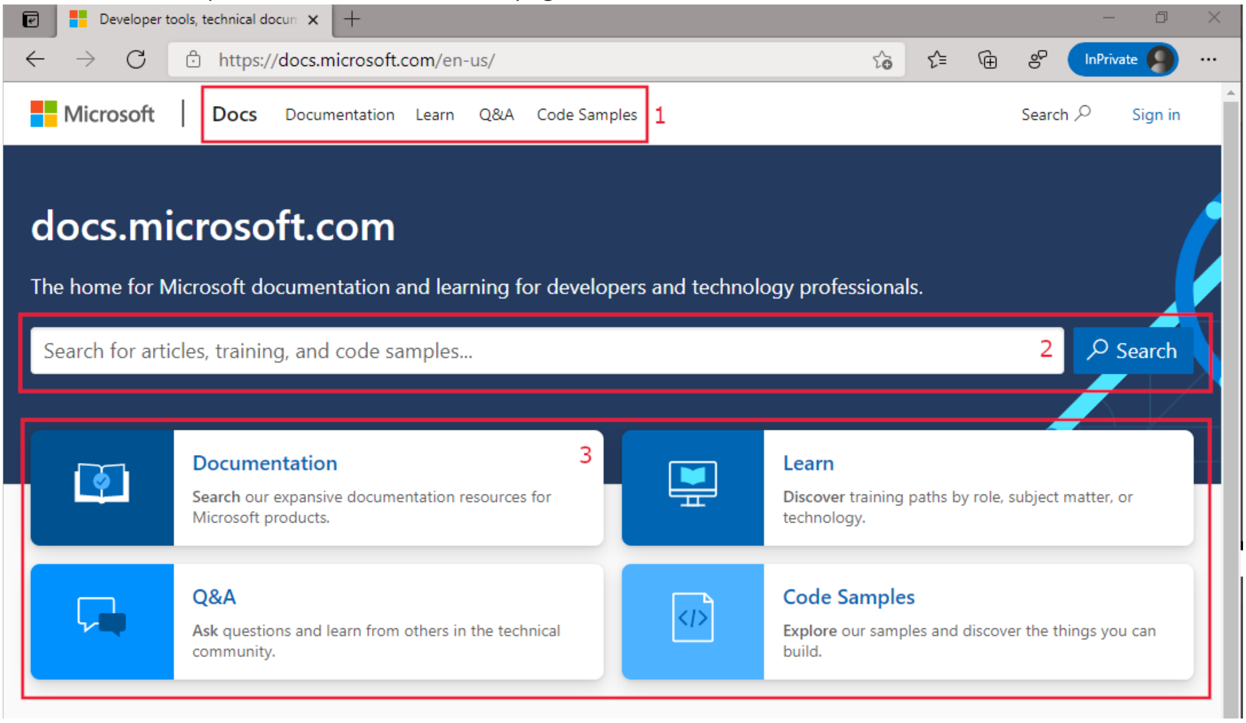
`docs.microsoft.com` (also known as the Microsoft Docs platform or DMC) is the home for Microsoft documentation and learning for developers and technology professionals. DMC launched in June 2016 as a replacement for the Microsoft Developer Network and TechNet libraries, which previously hosted this documentation. Over the years, DMC has expanded its charter and now includes four main services:

- **Documentation:** View conceptual articles, tutorials, guides, and references for our products and services.
- **Learn:** Discover training paths and certifications by role, subject matter, or technology.
- **Q&A:** Engage with others in the technical community.
- **Code Samples:** Explore our code samples to discover what you can build using Microsoft tools and technologies.

In this article, you'll learn how to navigate DMC's Documentation node. We'll also explore the common user interface shared by most Documentation articles.

DMC home page

Let's start at the top level with the DMC home page:



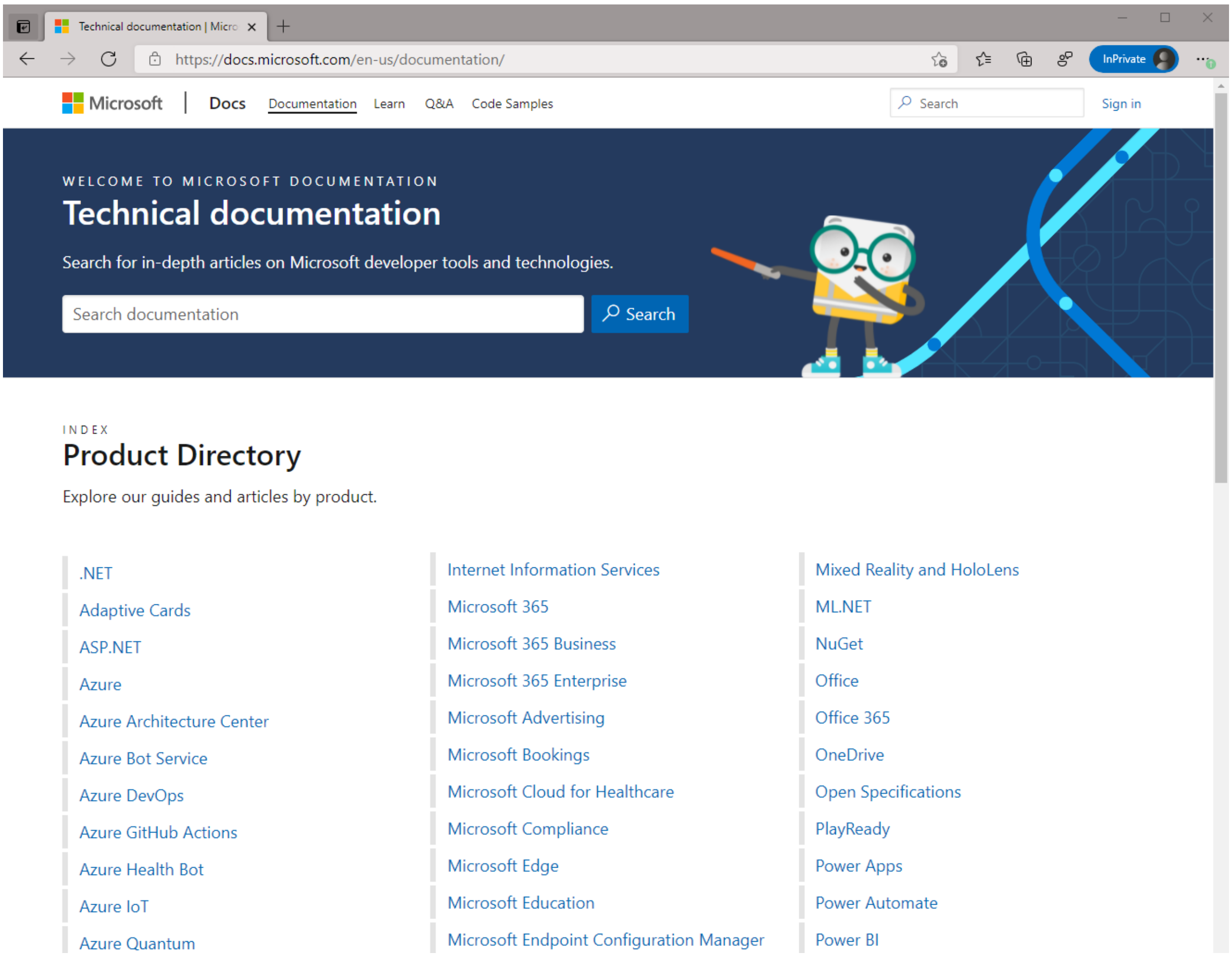
The DMC home page includes these elements:

1. **Global navigation:** All site content in DMC shares the global navigation (also referred to as Primary Nav). The global navigation provides access to the top-level subsites of Docs; to date, these subsites are Documentation, Learn, Q&A, and Code Samples.
2. **Search bar:** Use the search bar to search across all content in Documentation, Learn, Q&A, and Code Samples.
3. **Quick links to each DMC area:** If you know what you're looking for, jump to the content you need.

For the purposes of this article, we'll focus specifically on the **Documentation** area of DMC.

Documentation index

From the DMC home page, clicking **Documentation** in either the global navigation or the quick links brings you to the index for all Microsoft Technical Documentation:

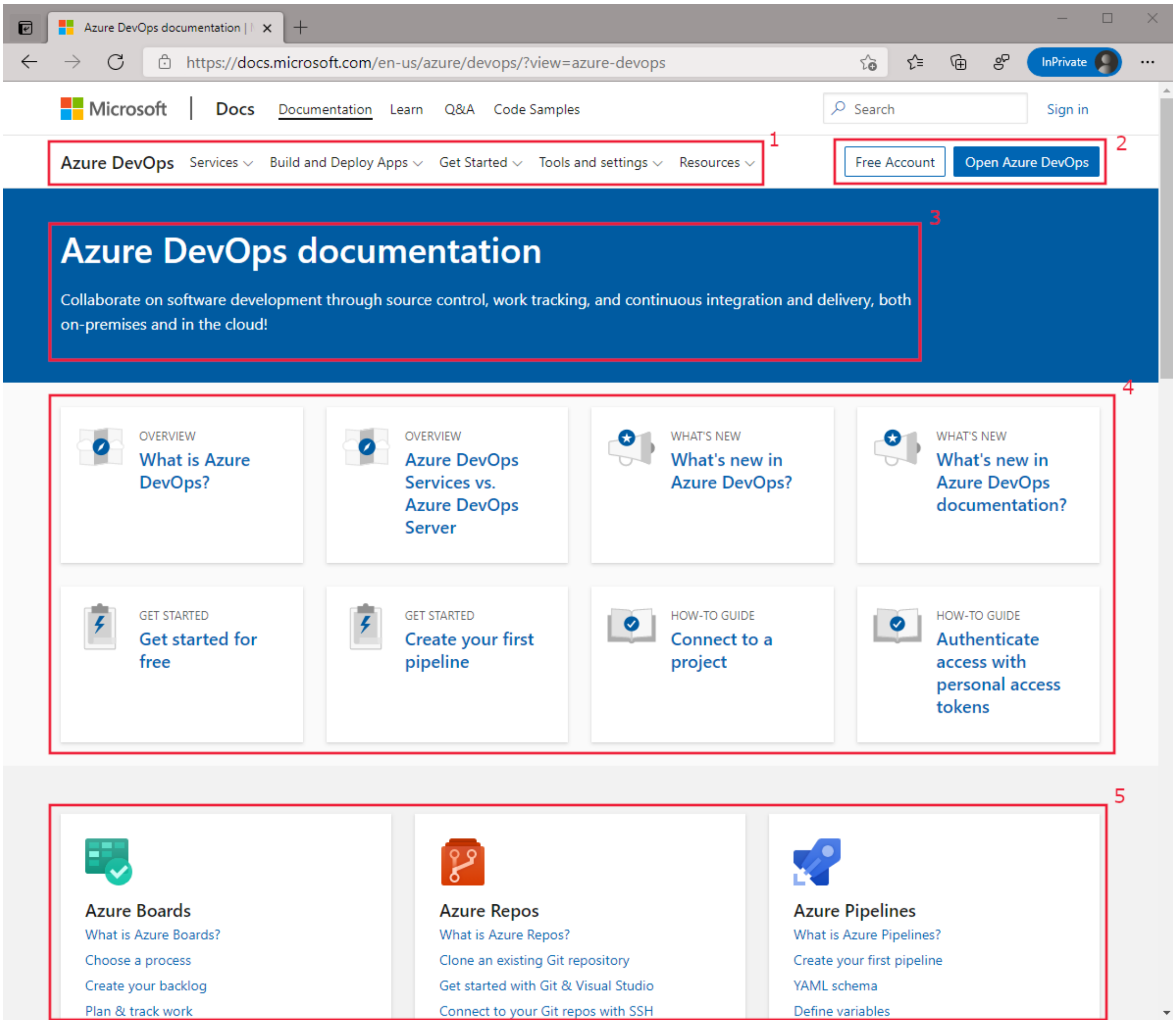


Notice how the global navigation remains and the **Documentation** link is now underlined. Use the search bar to search Documentation, or use the links in the Product Directory index to navigate to the documentation for a specific product.

Hub pages

Hub pages are collections of broader but related services, products, or languages. We use hub pages to help our customers better understand the bigger picture, see how components fit together, and quickly guide them to their area of interest.

From the Documentation index, select **Azure DevOps** from the product index to see an example hub page:



The Azure DevOps hub page includes these elements:

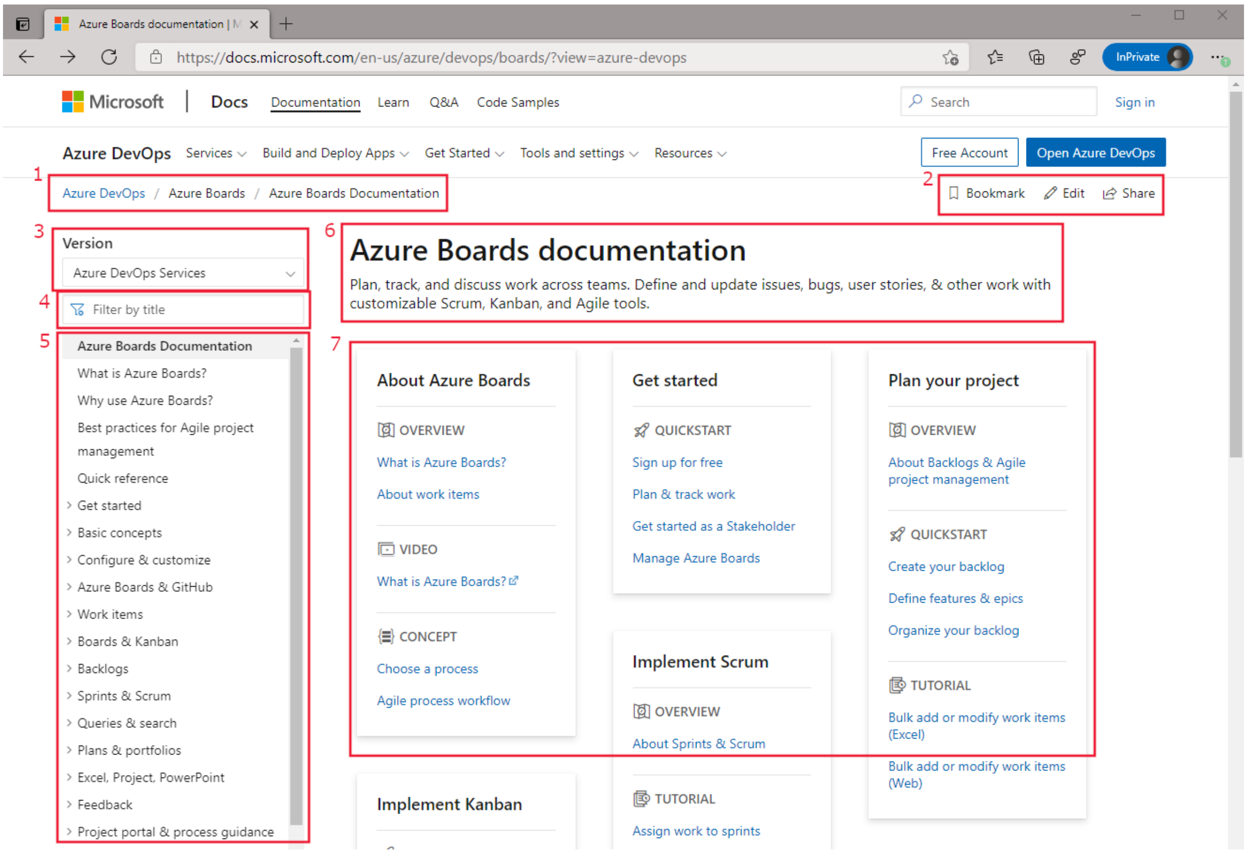
1. **Product anchor and product-level navigation**
   a. Most hub collections have a product anchor, which is usually the product name. If enabled, the anchor links to the top-level hub page.
   b. The product-level navigation helps users navigate product content. The Level 1 categories convey the breadth of documentation available for the product and should highlight meaningful areas of the product's documentation. Dropdowns in Level 1 provide links to Level 2 items.
2. **Action/Status bars:** Authors can add a blue primary button and a white secondary button to create a high-level action or indicate status. These buttons are part of the product-level navigation and persist across pages in the docset. In this example, the primary button opens Azure DevOps, and the secondary button helps the user create a free account.
3. **Root**: The root section contains the page title and summary for the hub collection.
4. **Highlighted content cards:** This section highlights content the authoring team thinks is most critical to its readers, like overviews, getting-started guides, how-tos, and content updates.
5. **Product directory cards:** The authoring team curates the directory to showcase product-related content.

Highlighted content and the product directory are optional sections and may not appear on every hub page. Other optional sections you may find include conceptual content cards, tools, and custom content.

Landing pages

Landing pages are an entry point for customers into a single docset. Usually, these pages map to a single table of contents (TOC) and appear at the top of the TOC. Authors use landing pages to surface top customer tasks or subjects for their product or service. Links on these pages point directly to articles within the docset. Landing pages can be a child of an overall hub page.

From the product navigation on the Azure DevOps hub page, select **Services --> Azure Boards** to see an example of a landing page:



Notice how the global and product-level navigation bars remain.

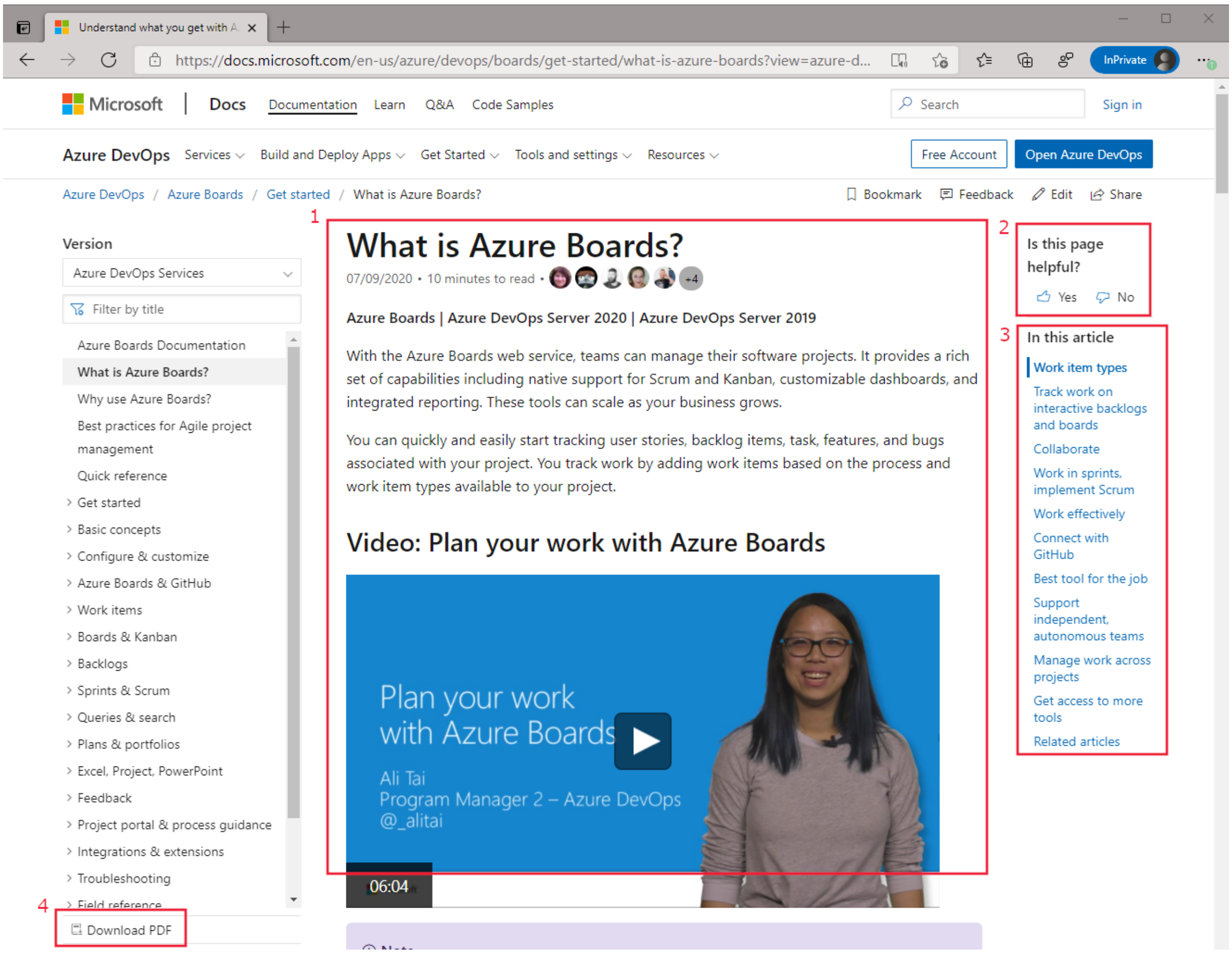
The Azure Boards landing page contains these elements:

1. **Breadcrumb**: Breadcrumbs are a series of links at the top of a page that convey the page's position in the overall site hierarchy. Breadcrumbs help set the context of the type of content you're reading, and they allow you to navigate back up the TOC tree. The page you're viewing appears on the far right, and each parent page appears to the left.
2. **Bookmark, contribute, and share**
   a. If you're logged in, bookmark a page to save it to your profile. Find all your bookmarks by clicking on your profile icon in the upper right-hand corner and selecting **Bookmarks**.
   b. If enabled by the authors, the **Edit** button allows anyone to contribute to the current page using GitHub for the web. Customers can use this contribution method to make small changes to an article.
   c. The **Share** button provides options to share an article via Twitter, LinkedIn, Facebook, or email.
3. **Version filter**: Some technical documentation covers content for more than one version of a product or service. Features can vary between versions, and so the content itself varies sometimes too. Use the version filter to select the appropriate version of the product you're viewing.
4. **TOC filter**: Search the titles within the TOC to find content faster.
5. **TOC**: Most services have one TOC with one landing page. For Azure documentation, the standard TOC nodes include Overview, Quickstarts, Tutorials, Samples, Concepts, How-to guides, Reference, and Resources.
6. **Title and summary**: Landing pages have a title and brief summary of the ideas covered by the docset.
7. **Top-task cards**: A landing page may have as many as 12 cards displaying the top customer tasks or top subjects, with supporting links.

Article pages

Articles make up the bulk of the pages within Documentation and cover top customer tasks. Articles can take the form of overviews, quickstarts, tutorials, concepts, how-to guides, FAQs, and more.

From the Azure Boards landing page, select **What is Azure Boards?** from the TOC to see an example of an article:



Notice how this page looks similar to the landing page. Landing pages and articles share many page elements, but there are a few elements that are unique to articles:

1. **Quality content:**
   a. All DMC content should align to our unified content model and writing principles. Documentation 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 *why*.
   b. Authors stamp each article with the date of the most recent major update to help you gauge freshness. You'll also find an estimated reading time and a list of contributors.
2. **Helpfulness feedback**: You can use the thumbs-up to give kudos to a helpful article, or you can use the thumbs-down to flag an unhelpful article. Selecting either will open a dialog box where you can provide optional comments before submitting your rating. Ratings and comments are private.
3. **Article navigation:** The right-hand navigation pane allows you to quickly navigate to sections within an article and identify your location.
4. **Offline documentation:** If enabled for the docset, you can download a PDF for offline reading. The PDF includes all articles within the entire TOC, not just the article you're viewing.

At the bottom of a Docs page, you'll find the Feedback control, if enabled:

Feedback

Submit and view feedback for

This product This page

The Feedback control allows you to submit product or documentation feedback.

- **This product** is the product-feedback button and directs you to the online location where you can provide feedback directly to the appropriate product team. For many products, feedback is collected in a UserVoice forum.
- **This page** is the page-feedback button and requires you to sign in to your GitHub account to submit feedback. Once signed in, you complete a feedback form, which is converted into a GitHub issue in the public GitHub repository where the content is published.

Content and Learning office hours

7/1/2021 • 2 minutes to read

Office Hours provide a dedicated time and place to get help with subjects related to our work. The current schedule of office hours, and guidance for hosting your own office hours, is below.

Current office hours schedule

Information architecture

Hosted by Rachel Price Thursdays 12-2pm Pacific

- Teams Meeting

A regular forum for partner teams, writers, onboarding PMs, etc. to bring questions about information architecture on Docs. Example questions could be:

- Is this TOC too deep?
- Are we using this feature correctly?
- What's a better way to organize this landing page?
- I think we need a hub page; how do we plan what it should look like?
- Wait, what's the difference between a hub and landing page?

Instructional design

Hosted by Mark Taparauskas Tuesdays 11am-12pm Pacific

- Teams Meeting

Whether you're currently writing content for Learn or just have some questions about how it works, stop by and say hello! The Learn Instructional Design team will be available to answer your questions about how to write titles, learning objectives, knowledge checks, etc. Our authoring guidelines are here: https://aka.ms/learn/id

PowerShell-Docs Office Hours

Hosted by Sean Wheeler every other Thursday 11:00am-12:00pm Pacific Time

Join the PowerShell-Docs Office Hours Teams channel

If your content set contains PowerShell examples or cmdlet reference, we are here to help. Come get help with your PowerShell content. Get answers from PowerShell specialists, share learnings and best practices, and get help onboarding your PowerShell content.

Get started by reading the PowerShell Docs Contributor Guide - https://aka.ms/PSDocsContributor

Email your questions to psdocsteam@microsoft.com.

Reference docs

Hosted by Robert Outlaw every Tuesday 12:30-1:30pm Pacific

- Calendar ics
- Teams meeting

Want to host office hours?

Anyone can host an office hours, and no permission is required to do so. However, we do ask that you follow these best practices.

Best practices

Always

- Always post your office hours in the Office Hours Schedule (above) and keep the information up to date.
- Always provide an ics link for the meeting in the Office Hours Schedule so people can add to their calendars as desired.
- Always provide a link to the Teams meeting in the Office Hours Schedule.
- Always record the meeting.
- Always send out brief notes after the session, with a link to the recording.

Consider

- Consider creating a page in the Contributor's Guide to describe the purpose in more detail and to provide links to related resources and information.
- Consider a specific agenda to avoid impractical discussions.
- Consider scheduling a conference room for in-person attendance.
- Consider scheduling at times where people in other time zones can attend. Earlier in the West Coast day is better.
- Consider a bi-weekly cadence. However, a weekly or monthly may be more appropriate.
- Consider running an office hour only for a few months as a pilot before establishing it for the long term. Consider ending an office hours series if fewer than 5 people are attending each session.

Content & Learning Vendor Supported Quality Programs

5/10/2021 • 2 minutes to read

Aquent and Acrolinx continue to provide quality-related support to the Content & Learning organization. The statement of work for managed editorial and authoring services is funded and scoped for the following work:

Editorial automation

Acrolinx provides instant automated feedback to authors about spelling, grammar, punctuation, terminology, and style. It runs automatically in every pull request and provides a link to results. You can also run Acrolinx locally in Visual Studio Code. Where Acrolinx is running, a minimum Acrolinx score of 80 is required to merge a pull request. Acrolinx is our first line of defense for baseline editorial quality.

- Acrolinx overview

Core editorial services

You can request a copy edit or developmental edit on technical documentation after you publish it. The vendor team also refers content for copy edit from the pull request queue if the content appears to need a copy edit. For copy edit, the editors review and provide feedback on spelling, grammar, punctuation, and clarity. They also review for basic Microsoft Style Guide items and product-specific terminology. For developmental edits, the team reviews the content for alignment to the C+L content model, to a clear customer intent, and to the the writing principles. Developmental edits include copy edits.

- Edit service description and instructions for requesting an edit
- Dev edit checklist for tutorials
- Editing Azure tech docs - resource for vendor editors

Pull request review

The resources on the managed service team review pull requests, provide authors with feedback, and merge pull requests that meet the defined quality criteria. This process applies to most Content & Learning conceptual content repositories, including the Architecture Center and the Learn repos.

The reviewers monitor pull requests across more than 20 repositories.

SLA

For pull requests that require human review, the target SLA is to pick up 50% of pull requests for review within 2 business hours of sign-off and the remaining 50% of pull requests within 4 business hours of sign-off. There is no SLA on how long it will take to review content or get it published after the author signs off because of variability in the volume of changes in pull requests, the amount of feedback needed, and the varied quality issues pull requests might have.

Resources

The following articles are related to this process:

- Pull request review and publishing schedules
- Criteria for merging
- Content release planning + process

- Content channel guidance
- Pull request submission best practices
- Public repo pull request review
- Query aggregation tool for supported repositories
- Pull request review process (private)
- Pull request review process (public)
- Aquent team wiki for PR review

Content authoring

A pool of technical writers is available for general content maintenance tasks such as freshness reviews, updates, and broken link fixing. The program functions under a fixed per-month budget. The team is focused on updating existing content that is low and medium priority and that has no fixed due date. These parameters ensure that the service can operate independently as a managed service. Items in this workflow are subject to prioritization by the Content & Learning leadership team and depend on organizational priorities, resourcing, and budget.

- Detailed content authoring program description
- Vendor team skillsets

Managed service contacts

Authoring, editing, and pull request review:

- Justin Chappell – Microsoft project owner
- Tami Fosmark – Aquent project manager

Acrolinx:

- Monica Rush – Microsoft project owner

Tutorial review checklist

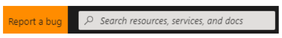
6/24/2021 • 5 minutes to read

The content model for docs.microsoft.com was created to support quality and consistency. The model provides the standards for several content types that work together to support a complete customer experience with our products and services. Tutorials are a part of that customer experience.

Tutorials are 100 – 200 level articles for an audience new to the service, product, or scenario. They're created from a list of top user tasks and provide the single best procedure for completing that task. Make sure you're familiar with the detailed guidance for writing tutorials before you continue with your review.

Clear cut criteria

CATEGORY	CRITERIA	ACTION	GUIDANCE
TOC	The tutorial is listed in the Tutorials node of the TOC.	Verify	Write tutorials - TOC
TOC	The article title in the TOC doesn't wrap to a second line when viewed in a full desktop browser screen.	Verify	Write tutorials - TOC
Customer intent	The customer intent statement is a markdown comment in the last line in the metadata section.	Verify	Write tutorials - Customer intent
Metadata	The ms.topic attribute has a value of "tutorial".	Verify	Write tutorials - Metadata
Metadata	The title attribute includes word "tutorial".	Verify	Write tutorials - Metadata
Metadata	The description attribute includes the word "tutorial".	Verify	Write tutorials - Metadata
H1	The H1 starts with "Tutorial: <verb>...".	Verify	Write tutorials - H1
Introduction	The introductory information is no more than 5 or 6 sentences.	Verify	Write tutorials - Introduction
Introduction	The introductory paragraph is followed by a sentence that says, "In this tutorial you'll do X..."		

CATEGORY	CRITERIA	ACTION	GUIDANCE
Introduction	The "In this tutorial you'll do X..." sentence is followed by a list of what is covered in the article using the green checkmark format.	Verify	Write tutorials - Introduction
Free account	There's a sentence that provides a link to a free account just before the first H2. For example, "If you don't have an Azure subscription, create a free account before you begin."	Verify	Write tutorials - Free account
Content	The content is referred to as a tutorial, not as a "guide", "article", or "topic".		
Screenshots	If there are screenshots of the portal, the first screenshot includes the full browser (Google Chrome or Safari (for Mac)).	Verify	Write tutorials
Screenshots	Screenshots aren't of an internal version of the product/service. Tip-off: Screenshots don't include these indicators in the banner at the top of the portal window:  	Verify	Screenshots
Prerequisites	If the article has prerequisites, **Prerequisites** is the first H2 in the article.	Verify	Write tutorials - Prerequisites
Links	There are no links to outside content after the **Prerequisites** section and before the **Next steps** section.	Verify	Write tutorials - Linking
H2s	The H2s don't wrap in the right-hand nav.	Verify	Write tutorials - H2s
H2s and H3s	The H2 and H3 headings aren't numbered.	Verify	Write tutorials - H2s
H2s and H3s	The H2s and H3s start with a verb.	Verify	Write tutorials - H2s

CATEGORY	CRITERIA	ACTION	GUIDANCE
Clean up resources	The article includes a **Clean up resources** H2 that immediately precedes **Next steps**. If there are no resources to clean up, the section should state that.	Verify	Write tutorials - Clean up resources
Next steps	The article includes a **Next steps** H2 section.	Verify	Write tutorials - Next steps
Next steps	There's a single link in the **Next steps** section that uses the "blue box" format.	Verify	Write tutorials - Next steps

Criteria that requires a judgment call

CATEGORY	CRITERIA	ACTION	GUIDANCE
Customer intent	The customer intent statement follows the *As a < type of user >, I want < what? > so that < why? >* format. Tutorials should be scoped to a single customer intent. Review the customer intent statement and ask yourself if it represents a task that a customer would likely need an answer to. For example, it's unlikely that a customer would need a series of unrelated management tasks as an answer to a single problem.	Verify	Write tutorials - Customer intent
Content type	The tutorial documents only one way to complete the task defined by the customer intent statement. Tip-offs: Optional steps, various ways to do the same task.	Verify	Writing principles checklist
Content type	The tutorial creates an example or proof of concept environment. Tip-offs: fictitious names, specific values for inputs, sample environments.		

CATEGORY	CRITERIA	ACTION	GUIDANCE
Content type	The bulk of the article is procedural steps. Conceptual content exists only to help the customer better understand how or why they would complete a step.	Verify	Choose a content type
Introduction	The introductory paragraph explains the customer intent and what the customer will accomplish in the article.	Verify	Write tutorials - Introduction
CLI	If the article uses the CLI, and if all commands run in Cloud Shell, the introduction includes the cloud-shell-try-it.md include file.	Verify	Write tutorials - Includes
PowerShell	If the article uses PowerShell, and if all commands run in Cloud Shell, the introduction includes the cloud-shell-powershell include file.	Verify	Write tutorials - Includes
CLI/PowerShell	If the article uses the CLI or PowerShell and all the commands don't run in Cloud Shell, the introduction includes instructions for the user to install the CLI or PowerShell locally.	Verify	Write tutorials - Includes
Screenshots	Screenshots offer value for the customer (explain complex UI, provide context, validate they've taken the correct steps or are in the right place, etc.).		
H2s	The H2s align with, but don'tnecessarily mimic the green checkmarks in the introduction.	Verify	Write tutorials - H2s
H2s	Each H2 is followed by one or two transitional sentences explaining why the steps need to be performed or how they contribute to the whole.		

CATEGORY	CRITERIA	ACTION	GUIDANCE
Alerts	If alerts are used, they're used sparingly. No more than two per article, and multiple notes should never be next to each other.		
Code blocks	Code blocks include language indicators.		
Body	CLI or PowerShell focused articles do not include numbers/bullets for the steps.	Verify	Write tutorials - Bullets
Everyday words	The article includes keywords that would the customer search for that would take them directly to this article.		
Everyday words	The article doesn't sound too formal.		
Everyday words	The article doesn't include unnecessary Microsoft jargon.		
Write concisely	The article has documented only what is necessary to achieve the customer intent.		
Write concisely	The article has no unnecessary information getting in the way.		
Write concisely	The article uses locators (>) for "click, click, click" instructions.		
Easy to scan	The article is not too long to allow for scanning.		
Easy to scan	The most important information is first in the article.		
Easy to scan	The section headers lead the customer through the article.		
Easy to scan	The article use tables and bolding to organize information or break up long blocks of text.		

CATEGORY	CRITERIA	ACTION	GUIDANCE
Easy to scan	Articles that focus on the CLI or PowerShell don'tinclude numbers or bullets.		
Empathy	The article reads like we're a partner to the customer.		

Vendor pull request review: public repos

5/10/2021 • 10 minutes to read

This article documents the responsibilities agreed to by Aquent and Content & Learning for the vendor team that reviews public pull requests in Content & Learning repos.

- Check each assigned public repository at least twice daily to review pull requests that have the `ready-to-merge` label assigned and merge them.
- Facilitate and unblock the workflow when people have minor issues with process, GitHub, or Git.
- Escalate issues and questions to the appropriate FTE through the Aquent team lead.

General process for reviewing PRs

1. Run the Public repos query for Aquent PR reviewers query to find all the signed off pull requests for the public repos. Run the query periodically throughout the day. The first round of reviews needs to be complete by 9:45 AM each day.
2. Check for email sent to azdocspr by a peer, colleague, lead, or manager who is signing off on a public PR for the listed author.
3. Review each signed off pull request from steps 1 and 2:
   - Review the PR to verify the changes are updates or corrections of a technical, editorial, or formatting nature.
   - The changes should align to the normal criteria used for PRs in the private repo.
   - If you see obvious editorial issues in a PR, fix them using the GitHub UI. Make sure to fix obvious problems with spelling, grammar, punctuation, and product names. Remove locales (en-us, en-gb) from Microsoft website links where the locale isn't needed.
   - Verify link fixes.
   - When you complete the review and the PR is ready to merge, add a brief but specific comment. Then, merge each pull request.
4. Run the Public repos query - all open PRs to see all open pull requests in the public repos. This shows pull requests where an employee intended to sign off but did not follow the defined process. Any PRs that are returned need to be carefully evaluated to determine intent and whether the person who attempted to sign off is qualified to do so (they must be a trusted source from the content or core PM team).
5. For new pull requests since the last time the team checked, review the pull request titles looking for indicators of minor editorial changes. For these sorts of pull requests, verify that the change is indeed trivial or purely editorial in nature, and merge. If a change seems truly editorial in nature and the affected text is a programming element or code block, the reviewer can merge the PR.
6. Check for the **review-team-triage** tag on any pull requests - these pull requests need to be manually evaluated and manually assigned.
   - If a pull request appears to affect many unrelated files, contact the repository admin and ask them for help. For the Azure repo, contact the PR review program PM.
   - Includes do not always contain metadata. If a PR is filed against an include file, search the repo for articles that use the include file. Assign the PR to the author of the articles that use the include file most frequently.

- If a PR is filed for images only, look up the article that uses the image. Assign the PR to the author listed for that article.

7. Check for sign-offs that don't follow the process.

8. Check for pull requests where the contributor license agreement (CLA) has not been completed. In the list of open PRs, the PR is marked with a yellow dot instead of a green check mark. If the CLA has been pending for more than 15 days, close the PR with the comment "Closing pending CLA completion."

Configuration file modifications in public repos

Public contributions should typically be against technical content in the repository. Any pull requests that update a configuration file need to be raised up to the repo admin and Cory Fowler. In general, we should not accept such changes from the public repos. Config files are usually identifiable because most include a period as the first element in the file name:

.gitignore .gitattributes .openpublishing.build.ps1 docfx.json

Specifically, we do not want to merge .gitattributes changes to manage line endings - this is work that needs to come from our internal engineering team. For more info, see https://github.com/MicrosoftDocs/SCCMdocs/pull/1732.

Understand sign-offs

There are two ways people can sign off on public pull requests:

- The author listed in the article metadata can type #sign-off in the pull request to assign the ready-to-merge label.
- An internal Microsoft colleague can send an email to the techdocprs alias to sign off for the author listed in the article metadata. This person can be a peer PM or writer, or it can be a content team lead or manager. These people are considered "trusted sources". Sending an email to request a merge is the exception process for cases where the listed author isn't available to sign off directly in the PR.

The sign-off automation is rigid because it's looking only for the author to sign off. If an author signs off the wrong way, merge the PR and then point them nicely to the correct process. If a peer PM or a content developer in Erin’s org signs off, merge. Ideally, they send mail, but we accept sign off from these folks even though the automation doesn’t. These process issues mean you need to query the repo for “sign-off”, “approved”, or other known variants to see if a trusted source signed off. Also query for “please-close” in case the automation isn't working.

Additionally, authors may list their GitHub account incorrectly in the article metadata. An incorrect GitHub account in the metada also breaks the automation.

Figuring out who is a trusted source for sign-off

- Anyone listed in the existing article metadata in the author or ms.author value is a trusted source.
- If a user is in Erin Rifkin's organization, they're considered a trusted source.
- Program managers on core product teams are trusted sources - usually, they are in C+AI. Cognitive Services PMs are in the Research org. Check the address book to see if their address book entry mentions the product area they work in. If the area they work in is "storage" and their address book title mentions "storage", consider them trusted. Check the address book or Dr. Whom to see if the PM who is signing off is a peer of the PM listed as the author. There are tools to help you determine identities and relationships:
  - Chrome extension for GitHub identities: https://repos.opensource.microsoft.com/settings/security/tokens/extension

- Dr. Whom: https://who
- If you are unsure, ask the Aquent team lead.

Merge conflict policy

- When a writer signs off on a pull request, but the PR is blocked by a merge conflict, the PR reviewer resolves the merge conflict by using the GitHub UI. The reviewer resolves the conflict by taking the public contributor's changes. In each conflict, the text above the divider is from the contributor's branch, and the text below the divider is what is in master. This may mean some changes from private master are lost or regressed. To avoid this problem, authors should pull the changes into the private repo and resolve the conflict to their satisfaction. Instructions are in Resolve merge conflicts article in this contributor guide.

- If a merge conflict occurs for the content of the entire article, it's usually because an author moved or deleted the article after the public PR was filed on the article. The reviewer can verify whether the article exists by searching the repo for the file name. (Tip - while you are at the base URL of a repo, press T on your keyboard to open the file filter for the repo. Type the file name to search for it in the repo.) If the file has been deleted or moved to a different folder in the repository, the pull request has to be closed. This typically happens only when a pull request sits open for too long in the PR queue. When you close a PR like this, make sure to explain why you closed it.

> **IMPORTANT**
> The GitHub merge conflict tool in the UI creates a commit back from the target branch to the source branch. Under certain conditions, it can cause an accidental content release into master from a release branch. **Never use it in the private repository** - use the command line to pull the merge conflict to your local clone and resolve.

Formatting changes

Pull requests that modify formatting elements (tables, code blocks, line end spaces) have the potential to render incorrectly. PR reviewers should pull these sorts of changes to the private repository for build validation and staging. Instructions are in the How to work with public pull requests article.

New articles and significant rewrites in a public repo

When someone submits a new article or a significant rewrite in a public repository, verify whether the contributor is internal or external.

- **Internal (Microsoft employee or vendor)**: Close the pull request and direct the internal contributor to work in the corresponding private repository. There are two resources they can use:
  - Public vs private repos
  - Fork and clone a repository
- **External (community)**: Contact the business approver for the service or product to let them know that a new article or significant rewrite from the community needs to be helped through the process. Point them to the process documentation in the How to work with public pull requests article.

Monitoring and unblocking the workflow

There are many problems in the public repos that can cause changes to get stuck in the process. Other problems can uneccesarily slowdown the process. Remember the job is to help authors and contributors complete their work and get approved PRs merged. The reviewers need to help unblock the PR workflow with a minimum of redundant communication and delays.

- Let the PR review program PM know if you see many PRs from a single user.

- If you see PRs in the public repo that update the ms.date attribute and nothing else, refer them to the business approver for the service or product area.
- Escalate anything unusual, unexpected, or concerning to the program PM.

Involving the FTE business approver

For the following kinds of issues in public pull requests, the PR reviewers must contact the business approver and the vendor program PM:

- You have questions about whether someone is a trusted source.
- Someone wants to change the listed author. These changes should be made in the private repo.
- New articles or significant rewrites. If the contributor is external to Microsoft, the business approver needs to pull in the appropriate writer to work with the external contributor to facilitate the contribution process. If the contributor is internal to Microsoft, the business approver needs to work with the contributor to help them work in the private repo.
- Any PR that contains more than five changed files. Customers usually submit changes article by article. Employees working on deeper content updates tend to modify many files. This level of change indicates that the work should be happening in the private repo.
- Any PR where the situation is getting convoluted or the customer is becoming agitated.
- Any PR where the comments indicate a trusted source approved the PR, but the conversation isn't driving the PR into the standard workflow.
- Any PR with conflicting guidance to hold or merge.
- Anything out of the ordinary.

How do the public repo changes get into the private repo?

The overall publishing process allows changes from the private and public repos to be merged to each other. The publishing system has a synchronization feature that is enabled for some repositories to help automate some of the syncing. The result of the publishing process is that changes in the public repo go live to docs.microsoft.com along with the publish of the changes in the private repo. The syncing and publishing work like this:

- When a public PR is merged, a new pull request is automatically created from the public repo to the private repo. This sync PR accumulates commits as PRs are merged in the public repo.
- As the first part of a publishing run, Pubdesk merges this sync PR in the private repo to the default branch in the private repo.
- Pubdesk then merges the default branch in the private repo to the live branch in the private repo.
- Within about 10 minutes of the merge to the live branch, an automatic commit is made from that live branch (in the private repo) to the default branch in the public repo. This is a **direct commit**, so there's no pull request created. This commit brings the two repos even with each other ("sync'd") until the next PR is merged in either repo.

This cycle restarts with the next merged commit in the public repo.

Aquent technical content authoring managed service

6/24/2021 • 7 minutes to read

Scope of service

Aquent Studios provides content authoring services to the Content & Learning team as a managed service. These services are a Content & Learning-funded service that helps extend the capabilities of Content & Learning content developers. It's scoped to articles and tasks that the content team owns and manages. This service isn't available to product teams for self-managed content. If you need more information about resourcing options for your content projects, contact your content publishing manager or Justin Chappell.

Content & Learning content teams can use the Aquent services for the following types of content updates and other knowledge-based tasks:

- Full freshness reviews and updates
- Testing procedures against the product and updating the corresponding content
- Adding new sections to existing articles based on bugs
- Converting content to new article templates/update to align to the content model
- Updating or editing existing screenshots, taking new screenshots
- Updating existing code samples
- Changing product or service names when more than a simple search and replace is needed
- SEO-driven updates to improve discoverability
- Documentation refactoring
- Content archival or deletion
- Metadata updates
- Updates to address geopol issues flagged by PoliCheck
- Localization fixes

The vendor authoring team focuses on:

- Normal and medium priority work
- Work that doesn't have a specific due date

The team schedules and completes work in the order in which it's received. The following work is out of scope:

- High priority, high-visibility work
- Work that is required by a specific deadline
- Content owned by program management
- Creating brand-new content

How to request the vendor team take on your project

Follow these guidelines when you create a user story in Azure Boards:

- Use this authoring request template to create your user story.
- In the title, specify "Content update" or "New content", followed by a description of the work.
- Provide the repository where the content resides, and list the files or URLs for any existing pages, as well as a page view report, as this will help us prioritize limited resources. For new articles, provide a filename list and

destination folder.

- Include scheduling constraints and the requested completion date.
- Determine a priority, and the business reason behind the priority.
- Scope your user story to list concrete deliverables, such as:
  - Create 10 articles for public preview for service X, due by date Y.
  - Make the fixes requested in content bugs X, Y, and Z filed against existing content.
  - Update product name X in entire content set for service Y, including when the context requires revisions due to this name change.
- Submit only work that requires core technical writing skills. Production-type work goes to the production vendor.
- List the FTE content SME who should review the final work to determine whether it meets the content team's expectations. This SME could be a manager or individual contributor, and doesn't have to be the person who created the user story.
- If you have a large work request, batch it into multiple requests, targeting a maximum of 10 URLs per request.

If you have an existing task or user story, Aquent can streamline this process by creating a new user story and linking to your tracking item. You're welcome to set up a call to talk through these details and Aquent will create the user story for you. In these cases, contact Tami Fosmark.

User story review and assignment

Once a user story is received, the Aquent project manager will assess the work request:

- Schedule a conference call to talk through the details of the user story. This step helps to uncover assumptions and agree on scope, and will be recorded as a reference for writers when work is ready to begin.
- Provide a rough estimate for when the work is likely to start. Due to queue volumes and recurring tasks (like high-priority validation fixes and build issues), and the first-in-first-out model, committing to specific dates is not feasible.
- If the request exceeds the capacity of the managed service team, work with the Microsoft program manager to determine the prioritization of the request.
- Once there are one or more resources available, Aquent starts work and monitors to ensure the work is completed as defined.

Microsoft and Aquent share responsibility of keeping team at capacity.

- Aquent provides weekly reporting on the queue, work status, and on-time completion.
- Microsoft ensures that adequate work is in the queue.
- Both share responsibility for negotiating competing project deadlines.

Rapid response

The Aquent team responds to issues in the daily builds for azure-docs-pr, such as merge conflicts, validation issues, and other warnings. When these issues block builds, the publishing team sends email to pr-fix@microsoft.com. The Aquent team acknowledges receipt and addresses the issues as quickly as possible, within 90 minutes of notification.

Ongoing quality tasks

The Aquent team handles some quality-driven tasks operationally, as ongoing work. The cadences for these ongoing quality tasks are specified per repo.

SLAs and scope

- One business day SLA to acknowledge a new user story.
- Aquent will notify listed authors when work is starting, and when content changes are ready for review.
- If the scope changes, the Aquent project manager and the requestor will determine whether to accept the changes or submit the work as a new user story.
- QA for Aquent ensures that the user story was completed within the scope outlined. Aquent is not providing deep technical review of the content.
- Listed authors and SMEs must review changes within five business days of notification; before the end of that period, authors and SMEs can provide an alternative date by which they will review the changes (within 10 business days). For work where there is no response from a listed author, or where the response is not within the agreed time window, the author's manager will be contacted for signoff within 4 business days. If there is no response from the listed author or the author's manager after the designated time, the Aquent author will sign off on the pull request to have the change merged.

Prioritization

Content teams should plan work early to get into the authoring queue. At times, prioritization may allow specific work to “leapfrog” the first-in-first-out queue. If in-progress work is affected by a prioritization change, or the impact to other work in the queue is large, Aquent and the Microsoft PM for the managed service will triage and prioritize work in the queue in consultation with the affected teams.

In general, Aquent will prioritize the queue in the following manner:

- P1 - Issues triage/response, merge conflict resolution for publishing are always committed work.
- P2 - Broad content changes caused by a major product update (ex: Azure portal UI), branding and terminology changes that affect large content sets.
- P3 - Freshness updates and all other content maintenance. Items that are P3 will be set aside for higher-priority work and could be set aside indefinitely, depending on the queue above.

Approval of completed work

If there is a technical SME or someone on the content team who needs to review the final work to ensure it meets the content team's expectations, the requestor must specify that FTE for technical review signoff in the user story description of work.

- For individual pull requests, the listed ms.author will indicate approval in a GitHub comment in the pull request.
- For user stories that require multiple pull requests, the vendor team may request a release branch in which to do the work. When the vendor team has completed the work in the release branch, they will create a pull request to master to highlight the changes for review. The FTE approver will indicate approval in a GitHub comment in the pull request. Then, the vendor team will follow the standard release process to get the branch merged and published.
- The FTE approver, SME, and ms.author can provide feedback on what needs to be fixed before approval. Teams should expect to provide some basic feedback. However, if there are significant or recurring issues or problems, the FTE approver should document the issues in the pull request and communicate the issue via email to Justin Chappell and Tami Fosmark.

Rules of engagement

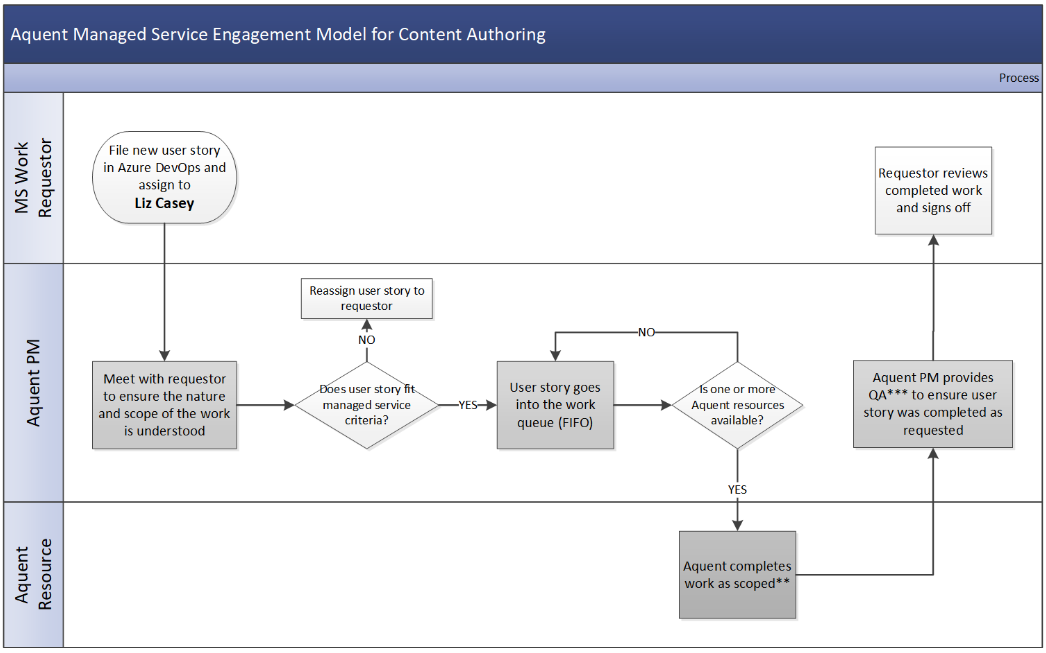
The following expectations provide compliance with Microsoft policies for external resourcing:

- Any new work using a product group budget will be completed using the staff augmentation model, rather than the managed service.

- If you need vendor writers to whom you can assign an area of work like you would to an FTE, use staff augmentation vendors. Staff augmentation positions require a job description and must be posted. In this case, Microsoft will interview the candidates, make the hiring decision, and manage the vendor writer.
- All work must be tracked using user stories in Azure Boards.
- Each user story's work will be scoped before it begins, to provide both an understanding of the volume of work and its position in the Aquent work queue/schedule.
- FTEs can't interview or select the staff within the managed service who will perform the work. Microsoft will let Aquent know the knowledge/skills needed by the managed service team, and Aquent will staff the service to meet those needs.
- Route feedback in a timely manner to Justin Chappell.

Work request process diagram

You can use the Aquent content authoring managed service for clearly defined and scoped content authoring tasks. Request work by creating user stories in Azure Boards:



*Acknowledge receipt of user story within one business day.

**If scope changes, Aquent will get sign-off from requestor on changes to scope and any impact to completion.

***QA for Aquent ensures that user story was completed within scope outlined. Aquent is not providing deep technical review of content.

How to obtain a staff augmentation vendor

If your content needs can't be met by this managed service model, hiring a staff augmentation vendor writer might be an option.

Questions/concerns/feedback

- Justin Chappell – Microsoft Program Manager for Aquent managed service
- Laura Belmont – Aquent service manager

Aquent content developer experience and skills

5/10/2021 • 2 minutes to read

The Aquent content authoring team provides a wide variety of skills and knowledge. This is a managed service available to Content & Learning content teams only. For more information on scope and engagement, see Aquent technical content authoring managed service.

If a proposed task requires expertise that is not listed below, feel free to contact Aquent's service manager Laura Belmont (v-laurab) to see whether the team can accommodate that task.

Coding skills

The Aquent team has expertise in reading, editing, and writing the following coding languages:

- .NET
- ASP.NET
- Azure CLI
- C#
- C++
- Java
- JavaScript
- JSON
- Node.js
- PHP
- PowerShell
- Python
- Ruby
- SQL
- Shell scripting (Bash)
- T-SQL
- VBA
- XML/XSLT

Technology/platform familiarity

The Aquent team has varying levels of familiarity with a wide range of technology areas, as shown in the following table, where:

- Familiar: know what it is and could get up to speed quickly.
- Conversant: good general knowledge but not an expert.
- Expert: deep understanding.

TECHNOLOGY	FAMILIAR	CONVERSANT	EXPERT
Azure Compute (Virtual Machines, Azure Batch, Azure Service Fabric)		X	

TECHNOLOGY	FAMILIAR	CONVERSANT	EXPERT
Azure Containers (Azure Batch, Azure Service Fabric)		X	
Azure Data + Analytics (HDInsight, Stream Analytics, Data Lake Analytics, Data Lake Store)		X	
Azure Databases (SQL, SQL Data Warehouse, Azure Cosmos DB, Azure Database for MySQL)		X	
Azure Developer Tools (Azure DevTest Labs, Azure DevOps, Xamarin, Application Insights)			X
Azure IOT (Hub, Edge, Steam Analytics, Machine Learning)			X
Azure Monitoring + Management (Azure Policy, Azure Monitor, Azure Backup)	X		
Azure Networking (Virtual Network, Load Balancer, Azure DNS)	X		
Azure Security + Identity (Active Directory, Multi-Factor Authentication)	X		
Azure Stack	X		
Azure Storage (Storage, Data Lake Store)		X	
Azure Web + Mobile (App Service, CDN, Media Services, Azure Search)	X		
Enterprise Mobility + Security (Intune, Identity Manager)		X	
Power Apps		X	
Power BI		X	
SQL Server			X
System Center		X	

TECHNOLOGY	FAMILIAR	CONVERSANT	EXPERT
Visual Studio			X

The Aquent team is also familiar with many platforms and operating systems, including:

PLATFORM	FAMILIAR	CONVERSANT	EXPERT
Windows			X
Linux		X	
Mac OS		X	
UNIX		X	

Content types

The Aquent content authoring team has extensive experience with writing conceptual technical content, how-to content, developer reference, and code samples, among others.

Schedule of ongoing quality tasks

5/10/2021 • 2 minutes to read

The Lionbridge content production team handles some ongoing tasks, in addition to ad-hoc work submitted by content teams. They handle localization fixes on demand, but other tasks follow regular cadences. Cadences are defined for each type of quality work, per repo.

The broken link workstream was migrated to Lionbridge in June 2020, and the team is phasing into the work, by adding a new repo each week. The Policheck workstream is a restart of a workstream that has not been in operation for a year or longer, and is restarting now as a pilot, with the team adding a new repo every other week as per the cadences below.

REPO	BROKEN LINKS	POLICHECK
architecture-center-pr	Active – Weekly	Active – Weekly
azure-csp-pr	Active – Monthly	Active – Monthly
azure-dev-docs-pr	Active – Weekly	Active – Weekly
azure-devops-docs-pr	Active – Weekly	Active – Weekly
azure-docs-pr	Active – Weekly	Active – Weekly
azure-reference-other-pr	Active – Monthly	Active – Monthly
azure-stack-docs-pr	Active – Monthly	Active – Monthly
cloud-adoption-framework-pr	Active – Weekly	Active – Weekly
cloudappsecuritydocs-pr	Active - Quarterly	Active - Quarterly
cpp-docs-pr	Active – Monthly	Active – Monthly
dataexplorer-docs-pr	Active – Weekly	Active – Weekly
emdocs-pr	Active - Quarterly	Active - Quarterly
intellicode-pr	Active – Monthly	Active – Monthly
IntuneEDU-pr	Active - Quarterly	Active - Quarterly
learn-pr	Active – Weekly	Active – Weekly
memdocs-pr	Active – Weekly	Active – Weekly
MIMDocs-pr	Active - Quarterly	Active - Quarterly
powerbi-docs-pr	Active – Monthly	Active – Monthly

REPO	BROKEN LINKS	POLICHECK
sql-docs-pr	Active – Weekly	Active – Weekly
SystemCenterDocs-pr	Active – Monthly	Active – Monthly
visualstudio-docs-pr	Active – Weekly	Active – Weekly

Check on in-progress work

The Lionbridge team tracks work via User Stories. These queries can help you find the current User Story for the repo(s) you're interested in:

- Query for in-progress broken link user tasks
- Query for localization fix user stories

Questions/concerns/feedback

- Justin Chappell – Microsoft project owner and content business SME
- Christopher McClister – Lionbridge project manager

Azure top 150 content freshness/maintenance

6/28/2021 • 2 minutes to read

To ensure that the most trafficked Azure articles remain “fresh”, the Aquent vendor content authoring team will perform a review each month and do a freshness pass on content that has not been updated in the last four months. This recurring task is now part of the ongoing quality work that’s part of the managed service.

Mission

On a monthly basis, the Aquent vendor content authoring team will identify Azure articles in the top 150 most-trafficked articles to identify any articles that have exceeded the 120 day freshness point (E:>120). An Aquent PM or writer will contact the author to let them know the content is past 120 days and that the Aquent team is available to update the article if the author agrees and can provide tech review at the end of the update. If the author agrees to the update, the Aquent team will conduct a full freshness pass.

Process

The Aquent vendor team will use this process to identify content for this work stream:

1. On or after the 10th of the month, visit https://aka.ms/contentperformancedashboard. Verify that data is posted for the prior month.
2. On the **PerformanceOverview** tab, click the link to download the page level detailed report.
3. From the OneDrive share, navigate to folder for the previous month and then download the Excel worksheet for high-traffic content. Choose the option to open the Desktop app for Excel.
4. In the Excel worksheet, enable content and turn on filtering. Then filter and sort as follows:
   - In the **LiveUrl** column, click **Text Filters** > **Contains** and enter `docs.microsoft.com/en-us/azure`.
   - Sort the **PageViews** column from largest to smallest.
   - Filter the **Freshness** column to `E:>120`.

   The list of articles that remains should be conceptual Azure content where the `ms.date` value has not been changed in over 120 days (4 months).
5. For each article, check value in the **LastCommitDateTime** column for the last commit. Frequently, a recent date may show for the last commit – check the **LinesChanged** column to determine if it was a major or minor update.
6. Contact the listed author by email using this template:

   > Hello *name*,
   >
   > I’m reaching out to you because you are listed as the author of this Azure technical article on docs.microsoft.com:
   >
   > - *Article name, hotlinked to the live page*
   >
   > This article is one of the top 150 most-visited Azure articles on docs.microsoft.com this month, and it does not appear to have been updated recently (based on the ms.date value). My team is tasked with ensuring the top 150 articles are kept up to date. Could you let me know if you are okay with us updating this article to ensure it is still complete, accurate, and current? When I’ve finished the

> update, I will ask you for review and sign off – I'll need your sign off within 5 days of my completion of the update (see the SLA details for more info).
>
> If you would rather update the article yourself, please let me know.
>
> Thanks, *Aquent staffer*

7. If you don't hear back from the author within 5 days, try to contact the author by IM. If the author is on vacation, please note their return date from their OOF and log this in the work item; follow up with them when they return by IM and email. At that point, if there is no response, let the author know via email that you are proceeding with the update.

8. Complete the article update per standard freshness processes.

Questions/concerns/feedback

- Justin Chappell – Microsoft project owner and content business SME
- Tami Fosmark – Aquent project manager

Repository coverage for PRMerger automation, vendor pull request review, and Acrolinx

7/9/2021 • 2 minutes to read

Private repos

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
Azure/azure-docs-powershell-aip	No	No	Yes
Azure/azure-docs-rest-apis	Yes	Yes	Yes
dotnet/docs-internal	No	No	Yes
dotnet/docs-maui-internal	Yes	No	Yes
MicrosoftDocs/ai-docs-pr	No	No	Yes
MicrosoftDocs/altspace-vr-pr	No	No	Yes
MicrosoftDocs/appcenter-docs-pr	Yes	No	Yes
MicrosoftDocs/architecture-center-pr	Yes	Yes	Yes
MicrosoftDocs/atadocs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-csp-pr	Yes	Yes	Yes
MicrosoftDocs/azure-dev-docs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-devops-docs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-docs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-reference-other-pr	Yes	Yes	Yes, VSCode only
MicrosoftDocs/Azure-RMSDocs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-stack-docs-pr	Yes	Yes	Yes

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/azure-stack-tzl-docs-pr	Yes		No
MicrosoftDocs/bi-shared-docs-pr	Yes	Yes	Yes, GitHub only
MicrosoftDocs/bot-docs-pr			Yes, GitHub only
MicrosoftDocs/cloud-adoption-framework-pr	Yes	Yes	Yes
MicrosoftDocs/cloudappsecuritydocs-pr	Yes	Yes	Yes
MicrosoftDocs/composer-docs-pr			Yes, GitHub only
MicrosoftDocs/cpp-docs-pr	Yes	Yes	Yes
MicrosoftDocs/cspp-pr	No	No	Yes
MicrosoftDocs/dataexplorer-docs-pr	Yes	Yes	Yes
MicrosoftDocs/emdocs-pr	Yes	Yes	Yes
MicrosoftDocs/enterprise-graph-docs-pr	No	No	Yes
MicrosoftDocs/fslogix-docs-pr	No	No	Yes
MicrosoftDocs/hybrid-pr	Yes	Yes	Yes
MicrosoftDocs/intellicode-pr	Yes	Yes	Yes
MicrosoftDocs/IntuneEDU-pr	Yes	Yes	Yes
MicrosoftDocs/learn-bizapps-pr	Yes	Yes	Yes
MicrosoftDocs/learn-dynamics-pr	Yes	Yes	Yes
MicrosoftDocs/learn-m365-pr	Yes	Yes	Yes
MicrosoftDocs/learn-mec-pr	No	Yes	Yes
MicrosoftDocs/learn-pr	Yes	Yes	Yes

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/LearnShared	Yes	No	Yes
MicrosoftDocs/machine-learning-server-docs-pr	No	No	Yes
MicrosoftDocs/media-services-pr		No	Yes
MicrosoftDocs/MEMDocs-pr	Yes	Yes	Yes
MicrosoftDocs/MIMDocs-pr	Yes	Yes	Yes
MicrosoftDocs/mixed-reality-pr	No	No	Yes
MicrosoftDocs/partner-center-pr	Yes	Yes	Yes
MicrosoftDocs/partner-center-sdk-pr	Yes	Yes	Yes
MicrosoftDocs/power-automate-docs-pr	Yes	No	Yes
MicrosoftDocs/powerbi-docs-javascript		No	Yes, VSCode only
MicrosoftDocs/powerbi-docs-pr	Yes	Yes	Yes
MicrosoftDocs/quantum-docs-private	No	No	Yes
MicrosoftDocs/salvador-sdk-pr	No	No	Yes
MicrosoftDocs/sccm-docs-powershell	Yes	No	Yes
MicrosoftDocs/security-benchmark-docs-pr	No	No	Yes
MicrosoftDocs/security-pr	Yes	No	Yes
MicrosoftDocs/socrates-pr	No	No	Yes
MicrosoftDocs/sql-docs-pr	Yes	Yes	Yes
MicrosoftDocs/SystemCenterDocs-pr	Yes	Yes	Yes

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/visualstudio-docs-pr	Yes	Yes	Yes, VSCode only
MicrosoftDocs/windows-ai-docs-pr	No	No	Yes, VSCode only
MicrosoftDocs/windows-uwp-pr	Yes	Yes	No
MicrosoftDocs/windowsserverdocs-pr	Yes	Yes	Yes
MicrosoftDocs/winrt-related-pr	Yes	Yes	No
MicrosoftDocs/workload-repository-pr	No	No	No
MicrosoftDocs/Xamarin-docs-pr	Yes	No	Yes

Public repos

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
aspnet/Docs	No	No	Yes, VSCode only
aspnet/EntityFramework.Docs	No	No	Yes, VSCode only
Azure/azure-docs-powershell	n/a	n/a	No
Azure/azure-docs-powershell-azuread	n/a	n/a	No
Azure/azure-docs-sdk-dotnet	No	No	No
Azure/azure-docs-sdk-java	No	No	No
Azure/azure-docs-sdk-node	No	No	No
Azure/azure-docs-sdk-python	No	No	No
dotnet/docs	No	No	Yes, VSCode only
dotnet/dotnet-api-docs	No	No	Yes, VSCode only
dotnet/docs-maui	Yes	No	Yes, VSCode only

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
Microsoft/vscode-docs	No	No	Yes, VSCode only
MicrosoftDocs/architecture-center	Yes	Yes	Yes
MicrosoftDocs/atadocs	Yes	Yes	Yes
MicrosoftDocs/azure-dev-docs	Yes	Yes	Yes, VSCode only
MicrosoftDocs/azure-devops-docs	Yes	Yes	Yes
MicrosoftDocs/azure-docs	Yes	Yes	Yes
MicrosoftDocs/azure-reference-other	Yes	Yes	No
MicrosoftDocs/Azure-RMSDocs	Yes	Yes	Yes
MicrosoftDocs/azure-stack-docs	Yes	Yes	Yes
MicrosoftDocs/biztalk-docs	No	No	No
MicrosoftDocs/cloud-adoption-framework	Yes	Yes	Yes
MicrosoftDocs/cloudappsecuritydocs	Yes	Yes	Yes
MicrosoftDocs/microsoft-community-training	Yes	No	No
MicrosoftDocs/cpp-docs	Yes	Yes	Yes
MicrosoftDocs/dataexplorer-docs	Yes	Yes	Yes
MicrosoftDocs/docs-powershell (This repo has only one article and only 35 PRs over time)	n/a	n/a	n/a
MicrosoftDocs/emdocs	Yes	Yes	Yes
MicrosoftDocs/hybrid	Yes	Yes	Yes
MicrosoftDocs/IntuneEDU	Yes	Yes	Yes
MicrosoftDocs/machine-learning-server-docs	No	No	Yes

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/MEMDocs	Yes	Yes	Yes
MicrosoftDocs/MIMDocs	Yes	Yes	Yes
MicrosoftDocs/mixed-reality	No	No	Yes
MicrosoftDocs/msix-docs	No	No	Yes, VSCode only
MicrosoftDocs/partner-center-sdk	Yes	Yes	Yes
MicrosoftDocs/power-automate-docs	Yes	No	Yes
MicrosoftDocs/powerbi-docs	Yes	Yes	Yes
MicrosoftDocs/PowerShell-Docs	n/a	n/a	Yes, VSCode only
MicrosoftDocs/project-rome	No	No	Yes, VSCode only
MicrosoftDocs/sql-docs	Yes	Yes	Yes
MicrosoftDocs/visualstudio-docs	Yes	Yes	Yes, VSCode only
MicrosoftDocs/windows-iotcore-docs	No	No	Yes, VSCode only
MicrosoftDocs/windows-uwp	Yes	Yes	No
MicrosoftDocs/winrt-api	Yes	Yes	No
MicrosoftDocs/winrt-related	Yes	Yes	No
MicrosoftDocs/winui-api	Yes	Yes	No

DOCS Common Tasks

3/5/2021 • 2 minutes to read

TASK	ARTICLE	LINK
Onboarding		
How can I get Private Repo Access?	Contribution Quickstart	https://aka.ms/msdocsonboard
Where is my content?	List of business approvers and repositories	https://aka.ms/msdocsrepos
Contribute Content		
I want to make a small doc change	GitHub contribution workflow for minor or infrequent changes	https://aka.ms/msdocsminor
I want to make a large doc change	GitHub contribution workflow for major or long-running changes	https://aka.ms/msdocsmajor
What's all this Metadata?	Metadata Reference	https://aka.ms/msdocsmetadata
Table of Contents updates		https://aka.ms/msdocsupdatetoc
Taking Screenshots		https://aka.ms/msdocimages
Documentation Performance		
How do I get metrics on my Docs?		https://aka.ms/get-mdm
How do I improve the SEO of my content?		https://aka.ms/msdocsseo
Repository Guidance		
Move content between Public and Private Repos	Article authors working in the public repo	https://aka.ms/msdocsreposwap
Can I use CodeFlow for content reviews?	Use CodeFlow to review PRs	https://aka.ms/msdocscodeflow
Can I get an Editing pass?	Repository coverage for PRMerger automation, vendor pull request review, and Acrolinx	https://aka.ms/msdocsrepoautomation
Reference Documentation		
What repo is my reference content published from?	List of reference repos by language and product	https://aka.ms/refrepolist

TASK	ARTICLE	LINK
How do I create or update REST API reference docs?	Update and onboarding instructions for REST API content	REST API Documentation
Community Engagement		
How do I respond to questions/comments on a Doc		https://aka.ms/msdocscommunity

Best practices for contributors

6/18/2021 • 2 minutes to read

Set up and maintain your work environment

Don't put your local repo in a OneDrive-synced folder

When you clone a repo, don't install to a folder that syncs with OneDrive (for example, your **Documents** folder). OneDrive can cause unintended merge conflicts when you merge your changes back to the main branch. For simplicity, it's best to create a **GitHub** folder on your C:\ drive (for example, ***C:\GitHub***).

Make sure you can access help channels

Many content teams have a Microsoft Teams channel you can join to get answers to your questions. Use these channels if you can't find help in the contributor guides or if you get stuck. Join the help channel that corresponds to your content group.

CONTENT GROUP	HELP CHANNEL
Content & Learning	Docs Support
MARVEL	Ask an admin

Work in branches to isolate changes

Never work directly in a main branch. You should always work in your own working branch and then merge those changes to the main branch.

If you'll be working on different projects, or even just updating topics with different timelines, you should work in multiple working branches. Using multiple branches let you work and save your changes in isolated environments.

Keep your branches (and forks) in sync

Refresh your branches and forks often and *always* before you start editing content. Keeping your branches in sync is the best way to minimize annoying merge conflicts. For more information, see Keep branches up to date in GitHub.

Author content

Always check which branch you're in before you start work

It's easy to open a file in Visual Studio Code and start editing, only to discover you're in the wrong branch. You can move your changes to your correct branch, but it's prudent to get in the habit of checking which branch you're in before you start editing.

Stash your work

If you're working in one of your branches on a longer-term project and need to make some quick updates to some other files, you can stash your changes before switching branches. Git will preserve the state of your in-progress files while you work in another branch. When you return to the first branch, you restore your stashed files and pick up where you left off.

You can stash files using Git Bash or GitHub Desktop.

Visual Studio Code and Markdown

- When you create a new article, copy the metadata from another topic and update as needed.
- Yellow squiggly lines indicate some kind of syntax warning. Hover over the squiggle to see an explanation and possible fix.
- Check the preview window for errors that are hard to spot in the code window (for example, text formatting, links, and images).

Publish content

Pull requests

Before you submit your pull request, check your content against the pull release criteria to minimize your risk of blocking issues. Each content organization has its own criteria. Be sure to check your work against the correct list.

- C&L criteria
- MARVEL criteria

Review content

Set up read-only or editable review links to make it easier for your reviewers. For more information, see Set up a link for tech reviewers.

Work with partners

Help internal partners who aren't familiar with the PubOps system and GitHub get started by creating stub files and providing them with branch and repo names. For more information, see Working with internal contributors.

Visual Studio Code keyboard shortcuts

6/18/2021 • 4 minutes to read

These are the most common keyboard shortcuts that writers use in Visual Studio Code. The complete set of Vshortcuts can found in Visual Studio Code under **File** > **Preferences** > **Keyboard Shortcuts**. For keyboard shortcuts for other operating systems and additional unassigned shortcuts, see Key bindings for Visual Studio Code.

General

COMMAND	ACTION
Ctrl + Shift+ P, F1	Show **Command Pallet**
Ctrl + P	Quick open, go to ...
Ctrl + Shift + N	New window/instance
Ctrl + Shift + W	Close window/instance
Ctrl + ,	User settings
Ctrl + K + S	Keyboard shortcuts

Basic editing

COMMAN	ACTION
Ctrl + X	Cut line (empty selection)
Ctrl + C	Copy line (empty selection)
Alt + ↑/↓	Move line up/down
Shift + Alt + ↑/↓	Copy line up/down
Ctrl + Shift + K	Delete line
Ctrl + Enter	Insert line below
Ctrl + Shift + Enter	Insert line above
Ctrl + Shift + \	Insert line above
Ctrl +]/[	Indent/outdent line
Home/End	Go to beginning/end of line

COMMAN	ACTION
Ctrl + Home	Go to beginning of file
Ctrl + End	Go to end of file
Ctrl + ↑/↓	Scroll line up/down
Alt + PgUp/PgDn	Scroll page up/down
Ctrl + Shift + [	Fold (collapse) region
Ctrl + Shift +]	Unfold (expand) region
Ctrl + K + [	Fold (collapse) all subregions
Ctrl + K +]	Unfold (expand) all subregions
Ctrl + K + 0	Fold (collapse) all regions
Ctrl + K + J	Unfold (expand) all regions
Ctrl + K + C	Add line comment
Ctrl + K + U	Remove line comment
Ctrl + /	Toggle line comment
Shift + Alt + A	Toggle block comment
Alt + Z	Toggle word wrap

Navigation

COMMAND	ACTION
Ctrl + T	Show all symbols
Ctrl + G	Go to line...
Ctrl + P	Go to file...
Ctrl + Shift + O	Go to symbol...
Ctrl + Shift + M	Show **Problems** panel
F8	Go to next error or warning
Shift + F8	Go to previous error or warning
Ctrl + Shift + Tab	Navigate editor group history

COMMAND	ACTION
Alt + ←/→	Go back/forward
Ctrl + M	Toggle **Tab Moves Focus** setting

Search and replace

COMMAND	ACTION
Ctrl + F	Find
Ctrl + H	Replace
F3/Shift + F3	Find next/previous
Alt + Enter	Select all occurrences of **Find** match
Ctrl + D	Add selection to next **Find** match
Ctrl + K + D	Move last selection to next **Find** match
Alt + C/R/W	Toggle **Case-sensitive/Regex/Whole word**

Multicursor and selection

COMMAND	ACTION
Alt + Click	Insert cursor
Ctrl + Alt + ↑/↓	Insert cursor above/below
Ctrl + U	Undo last cursor operation
Shift + Alt + I	Insert cursor at end of each selected line
Ctrl + I	Select current line
Ctrl + Shift + L	Select all occurrences of current selection
Ctrl + F2	Select all occurrences of current word
Shift + Alt + →	Expand selection
Shift + Alt + ←	Shrink selection
Shift + Alt + (drag mouse)	Column selection
Ctrl + Shift + Alt + (arrow keys)	Column selection
Ctrl + Shift + Alt + PgUp/PgDn	Column selection page up/page down

Rich languages editing

COMMAND	ACTION
Ctrl + Space	Trigger suggestion
Ctrl + Shift + Space	Trigger parameter hints
Shift + Alt + F	Format document
Ctrl + K + F	Format selection
F12	Go to definition
Alt + F12	Peak definition
Ctrl + K F12	Open definition to the side
Ctrl + .	Quick fix
Shift + F12	Show references
F2	Rename symbol
Ctrl + K + X	Trim trailing whitespace
Ctrl + K M	Change file language

Editor management

COMMAND	ACTION
Ctrl + F4 + W	Close editor
Ctrl + K F	Close folder
Ctrl + \	Split editor
Ctrl + 1/2/3	Focus into first/second/third editor group
Ctrl + K + ←/→	Focus into previous/next editor group
Ctrl + Shift + PgUp/PgDn	Move editor left/right
Ctrl + K ←/→	Move active editor group

File management

COMMAND	ACTION
Ctrl + N	New file

COMMAND	ACTION
Ctrl + O	Open file
Ctrl + S	Save
Ctrl + Shift + S	Save as...
Ctrl + K S	Save all
Ctrl + F4	Close
Ctrl + K + W	Close all
Ctrl + Shift + T	Reopen closed editor
Ctrl + K Enter	Keep preview mode editor open
Ctrl + Tab	Open next
Ctrl + Shift + Tab	Open previous
Ctrl + K P	Copy path of active file
Ctrl + K R	Reveal active file in Explorer
Ctrl + K O	Show active file in new window/instance

Display

COMMAND	ACTION
F11	Toggle full screen
Shift + Alt + 0	Toggle editor layout (horizontal or vertical)
Ctrl + =/-	Zoom in/out
Ctrl + B	Toggle sidebar visibility
Ctrl + Shift + E	Show Explorer/toggle focus
Ctrl + Shift + F	Show Search
Ctrl + Shift + G	Show Source Control
Ctrl + Shift + D	Show Debug
Ctrl + Shift + X	Show Extensions
Ctrl + Shift + H	Replace in files

COMMAND	ACTION
Ctrl + Shift + J	Toggle Search details
Ctrl + Shift + U	Show Output panel
Ctrl + Shift + V	Open Markdown preview
Ctrl + K V	Open Markdown preview to the side
Ctrl + K Z	Zen mode (Esc + Esc to exit)

Debug

COMMAND	ACTION
F9	Toggle breakpoint
F5	Start/continue
Shift + F5	Stop
F11/Shift + F11	Step into/out
F10	Step over
Ctrl + K + I	Show hover

Integrated terminal

COMMAND	ACTION
Ctrl + '	Show integrated terminal
Ctrl + Shift + '	Create new terminal
Ctrl + C	Copy selection
Ctrl + V	Paste into active terminal
Ctrl + ↑/↓	Scroll up/down
Shift + PgUp/PgDn	Scroll page up/down
Ctrl + Home/End	Scroll to top/bottom

Repository roles and responsibilities

5/10/2021 • 4 minutes to read

In Content & Learning documentation repositories, as a writer and contributor, you'll interact with different people in different roles within the publishing and quality review process. This article describes the roles and the responsibilities that correspond to those roles. The roles are:

- Repository administrator
- Business approver
- Quality program manager
- Pull request reviewer
- GitHub program manager (DocOps PM)

Different repositories handle the roles of repo admin and business approver differently.

- In some repos, the repo admin may also be the business approver.
- In the azure-docs-pr repository, there are different business approvers for different things, and the repo admin is a separate person.

Repository admin

The repository admin is responsible for the integrity of the repository. The admin is also responsible for the day-to-day operation of the repository.

- Makes sure that publishing is occurring on a documented, planned, and recurring schedule. Manages OOB and hotfix publishing.
- Creates release branches.
- Responsible for ensuring that the repository is not blocked by warnings, errors, or other publishing pipeline problems.
- Maintains the integrity of the repository by reviewing and merging pull requests that are out of scope for the pull request review team.
- Makes sure that configuration file changes are correct per documentation and that new features are turned on only after approval by the contributor guide team, by leadership, or based on current status of testing or experimentation.
- Makes sure that permissions are current and appropriate for the repository.
- Makes sure to engage the support team when necessary to keep authors unblocked.
- Until automation is provided, the admin closes abandoned pull requests that are more than three months old on a monthly basis and cleans up old or unused branches.

Business approver

The business approver is responsible for ensuring that all content that is merged meets the guidelines described in the contributor guide and other relevant standards. As an accountability in Content & Learning, the business approver role maps to content team managers.

For most routine content, the accountability for this work is delegated to the vendor pull request team managed by the vendor supported quality PM.

Business approvers must review and approve or decline certain types of change:

- Hub and landing page updates
- Escalations: pull requests that don't meet the pull request criteria and where the author indicates they want an exception.
- Merging of stub articles to a release branch. Content managers must approve these PRs to ensure that follow up occurs so that the stub articles are complete before publication. Even with approval, the stub articles need to follow the PR criteria for stub articles.

Business approvers must be documented in the contributor guide and should be content team managers. Business requirements may mean others are assigned as business approvers based on domain expertise – for example, Elizabeth Ross is a business approver for GDPR exceptions.

Most business approvers are tracked in the docs metadata tables in the ContentManager and ContentApprover columns. If ownership changes, follow the Change metadata process.

Azure business approvers are listed in the Service content owners article. Other business approvers are listed in the Business approvers and repos article.

Quality PM

The Quality PM leverages outsourced vendor and software resources to extend the capabilities of Content & Learning.

- Drives alignment with content guidelines across the organization through vendor PR review and editorial resources.
- Works with business approvers and repository admins on escalations and cross-repo processes to ensure consistency where multiple roles intersect in the PR review and merging process.
- Drives alignment with content guidelines across the organization through Acrolinx editorial and content strategy automation software.
- Collaborates with Contribution Guide owners to incorporate decisions into Acrolinx and the human PR review process.
- Drives vendor-based programs to improve content freshness and accuracy at time of publish.
- Manages Aquent and Acrolinx contracts and relationships. Single point of contact for these engagements.

Pull request reviewer

The pull request reviewer is accountable for ensuring that content meets the defined minimum quality bar for publication. Pull request reviewers review all pull requests that contain new articles and significant changes. They provide feedback to authors when the new articles or updated content fail to meet the published criteria. In some repos, the administrator may also perform this role, but in most Content & Learning repositories, a vendor team does this work. This vendor team reports to the quality PM, whose role is described in the previous section.

Pull request reviewers evaluate content against the content channel guidance and against the pull request review criteria:

- Content channel guidance
- Pull request review criteria

GitHub/DocOps PM

The GitHub and workflow PM is responsible for maintaining and improving Content & Learning workflow patterns and automation across all repositories in support of content quality and improved contributor workflow.

- Ensures that feedback from authors, partners, and Content & Learning and non-Content & Learning stakeholders is collected and integrated into GitHub authoring workflow improvements and automation.
- Works to streamline and automate GitHub authoring processes where possible and in a way that supports the quality and governance imperatives of other stakeholders in the process.
- Works with Content & Learning engineering PM/dev to ensure Content & Learning requirements are part of engineering planning.
- Drives Content & Learning strategies all up for increasing the depth of engagement with community in GitHub (issues, public repos, etc.).
- Planning and managing the eventual break up of the azure-docs-pr repository.
- Large event release management (Build, Ignite).
- Works with other workflow stakeholders to incorporate requirements. .

List of business approvers and repositories

7/16/2021 • 3 minutes to read

Repository and approver

REPOSITORY	APPROVERS	GITHUB ALIAS
dotnet/AspNetDocs	Mark LeBlanc	mcleblanc
dotnet/AspNetCore.Docs	Mark LeBlanc	mcleblanc
dotnet/EntityFramework.Docs	Mark LeBlanc	mcleblanc
dotnet/docs	Mark LeBlanc	mcleblanc
MicrosoftDocs/architecture-center-pr	Ed Price	EdPrice-MSFT
MicrosoftDocs/atadocs-pr	Rachel Karlin	rkarlin
MicrosoftDocs/azure-china-pr	Yue Shi (CELA); Martin Ekuan	-
MicrosoftDocs/azure-devops-docs-pr	Mike Jacobs	mijacobs KathrynEE Steved0x
MicrosoftDocs/azure-devops-extensions	Mike Jacobs	mijacobs
MicrosoftDocs/azure-dev-docs-pr	Barbara Kess Karl Erickson	barbkess KarlErickson
MicrosoftDocs/azure-docs-pr	Martin Ekuan and the content publishing manager for the service	martinekuan. For this repo, business approver and repo admin are separate. Cory Fowler is the admin.
MicrosoftDocs/azure-reference-other	Jason Howell Carolyn Gronlund	JasonWHowell (stream-analytics-docs, u-sql) CJGronlund (studio-module-reference)
MicrosoftDocs/Azure-RMSDocs-pr	Rachel Karlin	rkarlin
MicrosoftDocs/azure-stack-docs-pr	Femila Anilkumar	femila
MicrosoftDocs/bot-docs-pr	Kamran Iqbal, Erik Hopf, Nitin Mehrotra	Kaiqb, erhopf, nitinme
MicrosoftDocs/bi-shared-docs-pr	Owen Duncan	owend
MicrosoftDocs/bi-shared-docs	Owen Duncan	owend

REPOSITORY	APPROVERS	GITHUB ALIAS
MicrosoftDocs/cloud-adoption-framework-pr	Alex Buck	abuck
MicrosoftDocs/cloudappsecuritydocs-pr	Rachel Karlin	rkarlin
MicrosoftDocs/composer-docs-pr	Erik Hopf, Nitin Mehrotra	erhopf, nitinme
MicrosoftDocs/cognitive-toolkit-docs	Thiago Crepaldi Minna Xiao	thiagocrepaldi mx-iao liqunfu
MicrosoftDocs/commercialization-docs-pr	Eliot Graff Ted Maroutsos	EliotSeattle themar
MicrosoftDocs/cpp-docs-pr	Colin Robertson	corob-msft
MicrosoftDocs/dataexplorer-docs-pr	Rachel Karlin	rkarlin
MicrosoftDocs/devops-learn-pr	Mike Jacobs	mijacobs
MicrosoftDocs/DocsLegal-pr	Duncan Mackenzie	DuncanmaMSFT
MicrosoftDocs/emdocs-pr	Doug Eby	dougeby
MicrosoftDocs/flow-docs-pr	Kumar Vivek	KumarVivek
MicrosoftDocs/fslogix-docs-pr	Mike Tillman	mtillman
MicrosoftDocs/intellicode-pr	Mark Wilson-Thomas Josée Martens	markw-t j-martens
MicrosoftDocs/IntuneDocs-pr	Doug Eby	dougeby
MicrosoftDocs/IntuneEDU-pr	Doug Eby	dougeby
MicrosoftDocs/iis-docs	Mark LeBlanc	mcleblanc
MicrosoftDocs/IIS.Administration-docs	Mark LeBlanc	mcleblanc
MicrosoftDocs/kusto	Rachel Karlin	rkarlin
MicrosoftDocs/live-share	Jonathan Carter Josée Martens	lostintangent j-martens
MicrosoftDocs/machine-learning-server-docs-pr	David Peter Hansen	dphansen
MicrosoftDocs/MachineLearning-Python-pr	Carolyn Gronlund	cjgronlund
MicrosoftDocs/memdocs-pr	Doug Eby	dougeby

REPOSITORY	APPROVERS	GITHUB ALIAS
MicrosoftDocs/MIMDocs-pr	Karen Horan Bill Mathers	karenhoran billmath
MicrosoftDocs/mlnet-pr	Yiwen Zhu (NimbusML)	zyw400
MicrosoftDocs/msix-pr	John Kennedy	GrantMeStrength
MicrosoftDocs/partner-center-docs-powershell	Jason Howell	JasonWHowell
MicrosoftDocs/partner-center-java-docs	Jason Howell	JasonWHowell
MicrosoftDocs/partner-center-pr	Jason Howell	JasonWHowell
MicrosoftDocs/partner-center-sdk-pr	Jason Howell	JasonWHowell
MicrosoftDocs/partner-china-redirect	Jason Howell	JasonWHowell
MicrosoftDocs/partner-rest-pr	Jason Howell	JasonWHowell
MicrosoftDocs/powerbi-docs-pr	Kate Follis	kfollis
MicrosoftDocs/project-rome	John Kennedy	GrantMeStrength
MicrosoftDocs/query-docs-pr	Owen Duncan Vivek Kumar	owend kvivek
MicrosoftDocs/SCCMDocs-pr	Doug Eby	dougeby
MicrosoftDocs/stream-docs-pr	Archana Nair	archnair
MicrosoftDocs/sql-docs-pr	Craig Guyer Mike Ray Jason Roth	craigg-msft MikeRayMSFT rothja
MicrosoftDocs/visualstudio-docs-pr	Josée Martens Gordon Hogenson	j-martens ghogen
Microsoft/vscode-docs	Greg van Liew	gregvanl
MicrosoftDocs/win32-pr	John Kennedy	GrantMeStrength
MicrosoftDocs/windows-ai-docs-pr	John Kennedy	GrantMeStrength
MicrosoftDocs/windows-driver-docs-pr	Eliot Graff Ted Hudek	EliotSeattle tedhudek
MicrosoftDocs/windows-iotcore-docs	Martin Ekuan	martinek
MicrosoftDocs/windows-uwp-pr	John Kennedy Mike Jacobs	GrantMeStrength mijacobs

REPOSITORY	APPROVERS	GITHUB ALIAS
MicrosoftDocs/windowslegaldocs	Maggie Evans	MaggiePucciEvans
MicrosoftDocs/windowsserverdocs-pr	Liz Ross	eross-msft
microsoftgraph/microsoft-graph-docs/	Celeste De Guzman	celestedg
NuGet/docs.microsoft.com-nuget	Karan Nandwani	karann-msft
Samples		
Azure-Samples/azure-cli-samples	Sean Wheeler Delora Bradish	sdwheeler dbradish-microsoft
Azure/azure-docs-powershell-samples	Sean Wheeler Mike Robbins	sdwheeler mikefrobbins
dotnet/samples	Mark LeBlanc	mcleblanc
Microsoft/Windows-appsample-annotated-audio	Karl Erickson	KarlErickson
Microsoft/windows-appsample-coloringbook	Karl Erickson	KarlErickson
Microsoft/windows-appsample-customers-orders-database	Karl Erickson	KarlErickson
Microsoft/windows-appsample-familynotes	Joshua Partlow	JoshuaPartlow
Microsoft/windows-appsample-get-started-js2d	John Kennedy	GrantMeStrength
Microsoft/windows-appsample-get-started-js3d	John Kennedy	GrantMeStrength
Microsoft/windows-appsample-get-started-mg2d	John Kennedy	GrantMeStrength
Microsoft/windows-appsample-huelightcontroller	Joshua Partlow	JoshuaPartlow
Microsoft/windows-appsample-lunch-scheduler	Brendan Mitchell	bmitchell287
Microsoft/windows-appsample-marble-maze	Karl Erickson	KarlErickson
Microsoft/windows-appsample-networkhelper	Karl Erickson	KarlErickson
Microsoft/windows-appsample-photo-editor	Joshua Partlow	JoshuaPartlow

REPOSITORY	APPROVERS	GITHUB ALIAS
Microsoft/windows-appsample-photo-lab	Karl Erickson	KarlErickson
Microsoft/windows-appsample-remote-system-sessions	Brendan Mitchell	bmitchell287
Microsoft/windows-appsample-rssreader	Karl Erickson	KarlErickson
Microsoft/windows-appsample-trafficapp	Karl Erickson	KarlErickson
Microsoft/windows-appsample-xaml-hosting	Joshua Partlow	JoshuaPartlow
Microsoft/windows-task-snippets	Karl Erickson	KarlErickson
Microsoft/windows-tutorials-customer-database	Quinn Radich	QuinnRadich
Microsoft/windows-tutorials-inputs-and-devices	Karl Bridge	Karl-Bridge-Microsoft
API Reference		
aspnet/apidocs	Mark LeBlanc	mcleblanc
Azure/azure-docs-sdk-dotnet	Mark LeBlanc	mcleblanc
Azure/azure-docs-sdk-java	Robert Outlaw	rloutlaw
Azure/azure-docs-sdk-python	Robert Outlaw	rloutlaw
Azure/azure-docs-rest-apis	Robert Outlaw	rloutlaw
Azure/azure-docs-cli-python	Robert Outlaw	rloutlaw
dotnet/automl-api-docs	Natalie Kershaw	natke
dotnet/dotnet-api-docs	Mark LeBlanc	mcleblanc
dotnet/fsharp-api-docs	Mark LeBlanc	mcleblanc
dotnet/ml-api-docs	John Alexander	JRAlexander
dotnet/roslyn-api-docs	Mark LeBlanc	mcleblanc
MicrosoftDocs/azure-docs-sdk-node	Robert Outlaw	rloutlaw
MicrosoftDocs/azure-iot-c-sdk-reference	Wesley McSwain	wesmc7777

REPOSITORY	APPROVERS	GITHUB ALIAS
MicrosoftDocs/PowerShell-Docs	Sean Wheeler Joey Aiello	sdwheeler joeyaiello
MicrosoftDocs/powershell-docs-sdk-dotnet	Sean Wheeler Joey Aiello	sdwheeler joeyaiello
MicrosoftDocs/project-rome-java-api-docs	John Kennedy	GrantMeStrength
MicrosoftDocs/spark-dotnet-pr	Luis Quintanilla	luisquintanilla
MicrosoftDocs/visualstudio-docs-dotnet-api-pr	Anthony Cangialosi Josée Martens	acangialosi j-martens
MicrosoftDocs/vsts-docs-rest-apis	Will Smythe Jessie Huang	
MicrosoftDocs/vsts-docs-sdk-dotnet	Will Smythe Ted Chambers	
MicrosoftDocs/vsts-docs-cli-python	Will Smythe Ted Chambers	
MicrosoftDocs/windows-vision-skills-api-pr	John Kennedy	GrantMeStrength
MicrosoftDocs/windows-partner-center-dotnet	Jason Howell	JasonWHowell
MicrosoftDocs/winrt-related-pr	John Kennedy	jken

Azure DevOps repos:

REPOSITORY	APPROVERS	MSFT ALIAS
cpubwin/partner	Jason Howell	JasonWHowell
TechnicalContent/keros-docs-pr	Wendy Chisholm Mike Jacobs	wendc mijacobs
API reference		
cpubwin/advertising-sdk	John Kennedy	jken
cpubwin/sdk-api	John Kennedy	jken
cpubwin/store-api	John Kennedy	jken
cpubwin/wdk-ddi	Eliot Graff	EliotSeattle
cpubwin/winrt-api	John Kennedy	jken

REPOSITORY	APPROVERS	MSFT ALIAS
cpubwin/winui-api	John Kennedy	jken

Business approver list for Azure services

7/16/2021 • 7 minutes to read

This page lists the business approvers that oversee and ensure high quality for Azure services. The list helps PR Reviewers and other contributors know who to contact when approvals are needed or questions come up for a certain Azure technology. Business approver responsibilities are documented in the Roles and responsibilities article.

> **NOTE**
>
> There is a second similar list of managers that is used for reporting and automation. Follow the link to the **Detail** view from the `ms.service` list at Metadata taxonomies to find the corresponding Manager value for each row there.

How to update this list

By default, the content publishing manager accountable for a service is considered the business approver for that service. Content team managers are responsible for keeping this business approver list up to date. If an individual contributor is identified as a content lead with business approver responsibility, that person's manager needs to update this list.

There are two places to update:

1. Update this page (click Edit) to change or add new rows into the markdown table below. Keep the table sorted alphabetically, ignoring the prefix word Azure in the sort.
2. Also, follow the metadata changes steps to submit a request to update the ContentManager column in the metadata tracking database.

Business approver list

Refer to the table below for the name and GitHub alias of the business approver for each Azure Service:

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
A		
Active Directory	Karen Horan	karenhoran
Active Directory Domain Services	Karen Horan	karenhoran
Active Directory Federation Services	Karen Horan	karenhoran
Azure AD B2B	Celeste de Guzman	celestedg
Azure AD B2C	Celeste de Guzman	celestedg
Azure AD Application Management	Celeste de Guzman	celestedg
Azure AD Application & HR Provisioning	Celeste de Guzman	celestedg

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Azure AD for developers / Microsoft identity platform	Celeste de Guzman	celestedg
Azure Active Directory	Karen Horan, Celeste de Guzman	karenhoran, celestedg
Advisor	David Berry	DavidCBerry13
Analysis Services	Kate Follis	kfollis
API Management	George Wallace	georgewallace
Azure API for FHIR	Carolyn Gronlund	cjgronlund
App Center	George Wallace	georgewallace
App Configuration	John Kennedy	jken
App Service	George Wallace	georgewallace
Application Gateway	Kumud Dwivedi	KumudD
Application Insights	David Berry	DavidCBerry13
Arc for Servers	Maggie Evans	MaggiePucciEvans
Arcadia A365	Jason Roth, Craig Guyer	rothja, craigg-msft
Azure Artifacts	Mike Jacobs	mijacobs
Azure Communication Services	John Kennedy	jken
Automation	Maggie Evans	MaggiePucciEvans
Availability Zones	Mary McCready	mamccrea
Avere vFXT for Azure	Femila Anilkumar	femila
B		
Backup	Femila Anilkumar	femila
Bastion	Kumud Dwivedi	KumudD
Batch	Maggie Evans	MaggiePucciEvans
Batch AI	Carolyn Gronlund	cjgronlund
Big Compute scenarios	Maggie Evans	MaggiePucciEvans
Billing	Jason Howell	jasonwhowell

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Bing News Search	Nitin Mehrotra	nitinme
BizTalk Services	Doug Eby	dougeby
Blob	Tessa Wooley	twooley
Blockchain	Femila Anilkumar	femila
Azure Blueprints	Maggie Evans	MaggiePucciEvans
Azure Boards	Mike Jacobs	mijacobs
Azure Bot Service Composer	Erik Hopf, Nitin Mehrotra	erhopf, nitinme
Azure Bot Service SDK (Framework)	Erik Hopf, Nitin Mehrotra	erhopf, nitinme
C		
Azure Cache for Redis	Barbara Kess	barbkess
Azure CLI	Barbara Kess	barbkess
Content Delivery Network (CDN)	Kumud Dwivedi	KumudD
Cloud Services	Mary McCready	mamccrea
Cognitive Services	Nitin Mehrotra	nitinme
ConfigMgr	Doug Eby	dougeby
Container Instances	George Wallace	georgewallace
Container Registry	George Wallace	georgewallace
Azure Cosmos DB	Jason Howell	jasonwhowell
Cost Management	Jason Howell	jasonwhowell
Cost Management by Cloudyn	Jason Howell	jasonwhowell
D		
Data Box	Femila Anilkumar	femila
Data Catalog	Jason Howell	jasonwhowell
Data Factory	Jason Howell	jasonwhowell
Data Lake Analytics	Jason Howell	jasonwhowell

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Data Lake Storage Gen2	Tessa Wooley	twooley
Data Lake Storage Gen1 (was Data Lake Store)	Tessa Wooley	twooley
Data Share	Jason Howell	jasonwhowell
Azure Data Science Virtual Machines	Carolyn Gronlund	cjgronlund
Azure Databricks	Jason Howell	jasonwhowell
Database Migration Service	Jason Roth, Craig Guyer	rothja, craigg-msft
DDos Protection	Kumud Dwivedi	KumudD
Dedicated HSM	Rachel Karlin	rkarlin, mbaldwin
Deployment Manager	Carmon Mills	carmonm
Azure Device Certification	Phil Meadows	philmea
Device Update for IoT Hub	Phil Meadows	philmea
Azure Dev Spaces	George Wallace	georgewallace
Azure DevOps (all)	Mike Jacobs, Kathryn Elliott, Steve Danielson	mijacobs, KathrynEE, Steved0x
Azure DevOps Project	Mike Jacobs	mijacobs
Azure DevOps Starter	Mike Jacobs	mijacobs
Azure DevOps - Agile/Work	Mike Jacobs	mijacobs
Azure DevOps - CI/CD	Mike Jacobs	mijacobs
Azure DevOps - Collaboration	Mike Jacobs	mijacobs
Azure DevOps - Ecosystem & Integration	Mike Jacobs	mijacobs
Azure DevOps - Git	Mike Jacobs	mijacobs
Azure DevOps - Package	Mike Jacobs	mijacobs
Azure DevOps - SCS	Mike Jacobs	mijacobs
Azure DevOps - Test	Mike Jacobs	mijacobs
Digital Twins	Elizabeth Ross	lizross

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Disk Storage	Tessa Wooley	twooley
Azure DNS	Kumud Dwivedi	KumudD
E		
Azure Edge Zones	Kumud Dwivedi	KumudD
Event Grid	Jason Howell	jasonwhowell
Event Hubs	Jason Howell	jasonwhowell
ExpressRoute	Kumud Dwivedi	KumudD
F		
Azure Firewall	Kumud Dwivedi	KumudD
Azure Firewall Manager	Kumud Dwivedi	KumudD
Azure Front Door Service	Kumud Dwivedi	KumudD
Azure Functions	George Wallace	georgewallace
Azure FXT Edge Filer	Femila Anilkumar	femila
FHIR Server for Azure	dseven	-
Files	Tessa Wooley	twooley
G		
Genomics	Carolyn Gronlund	cjgronlund
Azure GO	Barbara Kess	barbkess
Azure Government	Femila Anilkumar	femila
Azure Governance	Maggie Evans	MaggiePucciEvans
H		
Azure HDInsight	Jason Howell	jasonwhowell
HPC Cache	Femila Anilkumar	femila
Azure Hub Page (articles/index.md)	Martin Ekuan, Craig Guyer	martinekuan, craigg-msft
I		

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Internet Analyzer	Kumud Dwivedi	KumudD
Intune	Doug Eby	dougeby
Intune for Education	Doug Eby	dougeby
IoT Central	Elizabeth Ross	lizross
IoT Device Developer	Elizabeth Ross	lizross
IoT Device Provisioning	Elizabeth Ross	lizross
IoT Edge	Elizabeth Ross	lizross
IoT Fundamentals	Elizabeth Ross	lizross
IoT Hub	Elizabeth Ross	lizross
IoT PnP	Elizabeth Ross	lizross
J		
Azure Java	Barbara Kess	barbkess
Azure JavaScript	Barbara Kess	barbkess
K		
Key Vault	Rachel Karlin, Matthew Baldwin	rkarlin, mbaldwin
Kinect DK	Martin Ekuan	martinekuan
Azure Kubernetes Service (formerly Azure Container Service)	George Wallace	georgewallace
Kusto/Data Explorer	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
L		
Azure Lab Services	Femila Anilkumar	femila
Azure Lighthouse	Maggie Evans	MaggiePucciEvans
Linux Virtual Machines	Mary McCready	mamccrea
Load Balancer	Kumud Dwivedi	KumudD
Log Analytics	David Berry	DavidCBerry13

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Logic Apps	Mary McCready	mamccrea
M		
M365 Business	Doug Eby	dougeby
M365 Enterprise	Doug Eby	dougeby
Azure Machine Learning	Carolyn Gronlund, Larry Franks	cjgronlund, blackmist
Managed Disks	Tessa Wooley	twooley
Managed Applications	Maggie Evans	MaggiePucciEvans
Management Groups	Maggie Evans	MaggiePucciEvans
Azure Maps	Phil Meadows	philmea
Azure Database for MariaDB	Jason Roth	rothja
Marketplace	Jason Howell	jasonwhowell
Media Services	Femila Anilkumar	femila
Microsoft Identity Manager	Karen Horan	karenhoran
Microsoft Information Protection	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Azure Migrate	Maggie Evans	MaggiePucciEvans
Mixed Reality	Martin Ekuan	martinekuan
Azure Mobile App	Maggie Evans	evansma
Mobile Engagement	Matt Soucoup	codemillmatt
Mobile Services	Matt Soucoup	codemillmatt
Azure Monitor	David Berry	DavidCBerry13
Mooncake / Azure in China	Femila Anilkumar	femila
Azure Database for MySQL	Jason Roth	rothja
N		
Azure NetApp Files	Tessa Wooley	twooley
Network Watcher	Kumud Dwivedi	KumudD

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Azure Notebooks	Joshua Partlow	JoshuaPartlow
Notification Hubs	Femila Anilkumar	femila
O		
Azure Orbital	Kumud Dwivedi	KumudD
Open Datasets	Carolyn Gronlund	cjgronlund
OMS	David Berry	DavidCBerry13
P		
Peering Service	Kumud Dwivedi	KumudD
Azure Percept	Phil Meadows	philmea
Azure Pipelines	Mike Jacobs	mijacobs
Azure portal	Maggie Evans	evansma
Azure Policy	Maggie Evans	MaggiePucciEvans
Azure Database for PostgreSQL	Jason Roth	rothja
Azure Powershell	Barbara Kess	barbkess
Power BI	Rachel Karlin, Kate Follis	rkarlin, kfollis
Power BI Embedded	Rachel Karlin, Kate Follis	rkarlin, kfollis
Private Link	Kumud Dwivedi	KumudD
Azure Purview	Jason Howell	jasonwhowell
Azure Python	Barbara Kess	barbkess
Q		
Azure Quantum	Carolyn Gronlund, Kitty Yeung	cjgronlund, KittyYeungQ
Queue	Tessa Wooley	twooley
R		
Azure Red Hat Open Shift	George Wallace	georgewallace
RemoteApp	Femila Anilkumar	femila

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Azure Remote Rendering	Florian Born	florianborn71
Azure Repos	Mike Jacobs	mijacobs
Reserved Instances	Mary McCready	mamccrea
Azure Resource Graph	Maggie Evans	MaggiePucciEvans
Resource Health	Carmon Mills	carmonmills
Resource Manager	Maggie Evans	MaggiePucciEvans
Rights Management	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Role-based access control (RBAC)	Karen Horan	karenhoran
Azure Route Server	Kumud Dwivedi	KumudD
S		
Scheduler	Mary McCready	mamccrea
Search	Heidi Steen, Nitin Mehrotra	HeidiSteen, nitinme
Security	Rachel Karlin	rkarlin
Security Center	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Security Center for IoT	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Azure Sentinel	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Server Management	Carmon Mills	carmonmills
Service Bus Messaging	Jason Howell	jasonwhowell
Service Bus Relay	Jason Howell	jasonwhowell
Service Fabric	George Wallace	georgewallace
Service Fabric Mesh	George Wallace	georgewallace
Service Health	David Berry	DavidCBerry13
Service Map	Carmon Mills	carmonmills
SignalR	Eliot Graff	EliotSeattle

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Site Recovery	Maggie Evans	MaggiePucciEvans
Azure Spring Cloud	Barbara Kess	barbkess
SQL Data Warehouse	Jason Roth	rothja
SQL Database	Jason Roth, steve stein	rothja, stevestein
SQL Server Stretch Database	Jason Roth	rothja
SQL Virtual Machines	Jason Roth	rothja
Azure Stack	Femila Anilkumar	femila
Static Web Apps	George Wallace	georgewallace
Storage	Tessa Wooley	twooley
Storage - Blob	Tessa Wooley	twooley
Storage - Disk Storage	Tessa Wooley	twooley
Storage - Files	Tessa Wooley	twooley
Storage - Queue	Tessa Wooley	twooley
StorSimple	Femila Anilkumar	femila
Stream Analytics	Jason Howell	jasonwhowell
Azure Supportability	Michael Blythe	mgblythe
Synapse Analytics	Jason Roth	rothja
System Center	Carmon Mills	carmonmills
T		
Azure Test Plans	Mike Jacobs	mijacobs
Time Series Insights	Ornat Spodek	orspodek
Traffic Manager	Kumud Dwivedi	KumudD
U		
V		
Virtual Machine Scale Sets	Mary McCready	mamccrea

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Virtual Machines	Mary McCready	mamccrea
Virtual Network	Kumud Dwivedi	KumudD
Virtual WAN	Kumud Dwivedi	KumudD
Azure VMware Solution by Virtustream	George Wallace	georgewallace
VPN Gateway	Kumud Dwivedi	KumudD
W		
Web Application Firewall	Kumud Dwivedi	KumudD
Web Apps	George Wallace	georgewallace
Web Apps on Linux (Containers)	George Wallace	georgewallace
Windows Virtual Machines	Mary McCready	mamccrea
Windows Virtual Desktop	Femila Anilkumar	femila
X		
Y		
Z		

Cheat sheet: basic Git command-line steps

7/7/2021 • 3 minutes to read

This cheat sheet is an abridged version of the steps outlined in Tutorial: Make changes to an article published on docs.microsoft.com. As a prerequisite, you must have already forked and cloned a repository (repo).

Create a new article, or update an article

These steps use MicrosoftDocs/azure-docs-pr as an example, but you can substitute any org/repo combination you've cloned. These steps also assume your fork is at the remote named `origin` and the upstream published repo is at the remote named `upstream`.

1. Launch your Git command-line tool (for example, Git Bash).

2. Change to your local repository:

   ```
   cd azure-docs-pr
   ```

3. Check out the default branch:

   ```
   git checkout master
   ```

4. Create a new local working branch derived from the master branch:

   ```
   git fetch upstream master
   git checkout -b <working branch> master
   ```

5. Let your fork know you created the local working branch:

   ```
   git push origin <working branch>
   ```

6. Create your new article or make changes to an existing article.

7. After you create or modify your article and images, add and commit the changes you made:

   ```
   git add .
   git commit -m "<comment>"
   ```

   Or, if you wish to only add the specific files you modified:

   ```
   git add <file path>
   git commit -m "<comment>"
   ```

8. Push the changes to your fork on GitHub:

   ```
   git push origin <working branch>
   ```

9. When you're ready to submit your content to the upstream master branch for staging, validation, and/or

publishing, go to the repo in GitHub (for example, https://github.com/MicrosoftDocs/azure-docs-pr) and create a pull request (PR) from your fork to the master branch.

10. If you're an employee working in the private repository, the changes you submit are automatically staged and a staging link is written to the PR. Review your staged content and sign off in a PR comment by adding the `#sign-off` comment. This indicates the changes are ready for review and to be published live.

11. If your update is small in scope, the PR may qualify for automatic merge. If not, the PR review team reviews your PR, provides feedback, and merges it if the PR meets the minimum quality criteria.

Work in multiple working branches

When you create more than one working branch, and you want to switch back-and-forth between them to iterate on different sets of work, do the following to prevent moving commits from one working branch to another working branch.

> **IMPORTANT**
> If you unintentionally move commits from one working branch to another, you will co-mingle files and edits from the first working branch into the second working branch.

1. When you're in a working branch and you want to get out of it, check if you have untracked files (changes that aren't staged for a commit):

```
git status
```

   a. If you have untracked files, in the Git Bash output they're shown in red font: go to Step 2.

   b. If you don't have untracked files, in the Git Bash output no files are shown in red font: go to Step 4.

2. Add and commit the changes in the untracked files:

```
git add .
git commit -m "<comment>"
```

3. Push the changes to your fork on GitHub:

```
git push origin <working branch>
```

4. Check out the other pre-existing working branch:

```
git checkout <other working branch>
```

5. Create a new article or make changes to an existing article. Add, commit, and push the changes to your fork on GitHub.

> **NOTE**
> If you do the procedure above and more than about an hour goes by, the prompt in the GitHub UI to create a PR disappears. To create a PR, see Create a pull request.

Next steps

- See these steps in full detail: Tutorial: Make changes to an article published on docs.microsoft.com
- Partner quick reference
- Learn more about Git

Create and publish sample code

6/24/2021 • 6 minutes to read

Learn how to create a sample code repository in the `Azure-Samples` GitHub organization for backing your articles on docs.microsoft.com.

> **IMPORTANT**
> These steps are for Azure sample repos for programming languages like Java, Node.js, or .NET, not scripting environments such as Azure CLI and Azure PowerShell. For samples guidance for those scripting languages, see Getting started with samples.

Prerequisites

- A GitHub account.
- A development environment to develop your sample.
- Review the Governance for Code samples at Microsoft before creating the content.

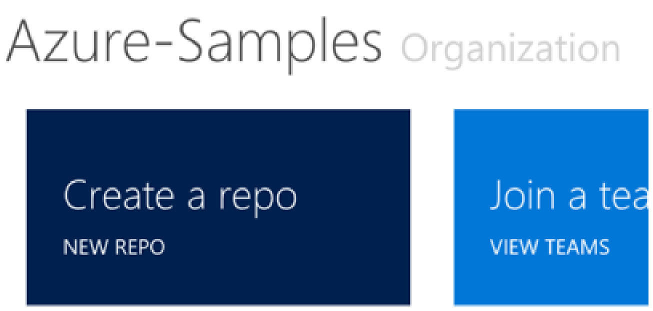
Associate your GitHub identity with your Microsoft identity

To request a new repo in the Azure-Samples organization and manage its permissions, you must first associate your GitHub identity with your Microsoft domain identity. Follow the steps the tutorial to associate your GitHub identity with your Microsoft one and use the Open Source portal to join the `Azure-Samples` organization.

Request a new repo

To create a new sample repo in the `Azure-Samples` organization:

1. Sign in to the Microsoft Open Source Hub
2. Click "Join Azure-Samples".
3. Select **GitHub** in the top menu.
4. Select the **Azure-Samples** link, then select the **Create a repo** box.



You're presented with a wizard-like experience for creating the new repo. Use the following values when prompted:

- For **Release registration**, choose **New release registration**.
- For **New release registration**, choose **Yes** to confirm it's a public open-source project.

> **NOTE**
>
> You can create private repos and make them public later when you're ready to release the sample.

Select **Sample code** for project type, **MIT** license, and fill in the appropriate answers for:

- Other source/binary code usage
- Whether the sample collects any personal data, including data sent in HTTP headers to REST API calls to Azure services.
- Confirm that the sample is less than 5,000 lines of code.

Select **Next** to confirm your choices.

Name your repo

On the **Create a new repository** page, enter a name for your new repo. Use the following rules to accurately describe your sample:

- Avoid using terms like "sample" or "getting started" in your repo name.
- All lowercase, words separated with dashes.
- Name should start with the service slug, as it is represented in docs.microsoft.com/azure. The easiest way to know your service slug is to go to the service marketing on azure.microsoft.com (example: https://azure.microsoft.com/services/search/ so the slug is `search`).
- After the slug, there should be a language or platform: dotnet, java, node, python
- Next, describe in two or three words what the sample does when run. It's often easiest to write out a sentence describing the repo, then extract the key terms in the sentence to create the rest of the name.

Examples:

- `mysql-java-create-todo-app`
- `cosmosdb-nodejs-create-mongodb-rest-api`

Add a repo description

The description text is visible on the repo page on GitHub. Limit the description to ~120 characters. Include what the sample is (web app, API, application template), the Azure services the sample uses, and the languages and/or frameworks used in the sample.

Examples:

- `Java bookshelf app using SQL Database and Azure Storage`
- `.NET REST API with Cosmos DB's SQL API`

Assign permissions

Assign at least one team to manage the repo settings for the sample repo. If you already belong to a team, that team name will be presented here. If you're not part of a team, create a new one in the Azure-Samples organization, then refresh this page.

Select the name of the team with Admin permissions for the repo, then select **Next**.

Select additional teams with write permissions to your repo. These teams can accept PRs and push commits to your repo. It's recommended that you add an Azure content development team to the write permissions. If problems with the repo are reported in the associated docs, the content team can correct and merge the changes.

Assign any Read Permissions in the same way.

In the **Contributor License Agreement**, leave **Microsoft** as the Legal Entity and select **Next**. Select an appropriate prepopulated `.gitignore` based on the language used in the sample, then select **Next**.

Review and submit the application for your repo.

Complete the README

Once the repo request is approved and the repo is created, the following files are added:

- README.md - Template for your sample's README. **Complete all sections of the README**. Do not alter the *Contributing* section.
- CODE_OF_CONDUCT.md - Contains links to Microsoft's open-source code of conduct documentation.
- LICENSE.md - Azure samples standardize around the MIT license.
- SECURITY.md - Information for reporting security issues found in the sample.

README title & description

Title - The main title (H1) of the README file (the line of text after the first `#` in markdown) is used as the title of the sample on both the details page and the docs.microsoft.com samples browser. Limit it to 120 characters or less.

Description - Under the title (H1), add a short explanation describing your sample. Show a screenshot of your sample running if it's web page or other interactive application. If it's a console application, include a formatted code block tagged with `console` that shows its usage and output.

- Link back to the appropriate quickstart or tutorial content on docs.microsoft.com if the sample is used in product documentation. If the sample is not used directly in content, link back to the Azure docs landing page for the service or developer center for that developer language.
- If using REST APIs or Azure client libraries in your sample, make sure to link both to the reference on docs.microsoft.com for the library and the GitHub where the client library is developed.

README content

The rest of the Markdown describes your sample. Complete each section of the README template:

- **Contents** - Outline the file contents of the repository.
- **Prerequisites** - Required components and tools to build and/or run the sample.
- **Setup** - How to prepare to run the sample.
- **Running the sample** - Step-by-step instructions to execute the sample and see its output. Include instructions on modifying the sample code for use in another project.
- **Key concepts** - Context on the tools and services used in the sample.
  - Add learn more links that go out to deeper content tutorials on docs.microsoft.com or other relevant samples on GitHub.
  - Point out which lines of code in the sample contain interesting Azure code, and make a short table with descriptions linking directly to those lines of code.

README metadata

The metadata at the top of the readme file helps the docs.microsoft.com platform associate the sample with Azure services and languages/platforms and improves their discoverability. Having the metadata section is required for a sample to show up on the docs.microsoft.com samples browser.

For detailed information about completing the YAML frontmatter metadata, see:

Onboarding Samples to docs.microsoft.com/samples

Best practices for sample code

- Sample code that creates billable Azure service instances and resources should clean up those resources after it finishes execution and the user terminates the sample. Don't remove the resources without confirming with the user. If possible, clean up resources even if an unhandled exception occurs at runtime.
- If you need access keys or connection strings for a service, point out to use the Azure CLI, PowerShell, or Cloud Shell to do so. If Azure CLI or PowerShell can't be used, point to content on docs.microsoft.com that shows how to find the values in the Azure portal.
- When mentioning class names in Azure client libraries in your README, link straight to the class in the API reference on docs.microsoft.com.

Accessibility tools and references

7/7/2021 • 4 minutes to read

Accessibility is an important part of contributing towards Microsoft's goal of inclusivity, and our mission to empower every person and every organization on the planet to do more.

> **IMPORTANT**
>
> Every team at Microsoft must design its products with every user in mind
>
> -- Satya Nadella

> **TIP**
>
> Official guidance is available on the intranet site: Microsoft Accessibility.
>
> We recommend reviewing the Accessibility 101 training to get a baseline understanding of the accessibility needs of our customers.

Accessibility tools

There are many tools available to assist you in your evaluation of content to accessibility standards. The following sections list a series of tools that can assist you with making your content accessible.

Discovery accessibility of content

> **IMPORTANT**
>
> Microsoft uses Accessibility Insights to test accessibility requirements. We need to implement the accessibility features so that both Microsoft Narrator and NonVisual Desktop Access (NVDA) produce the same user experience across both the Microsoft Edge browser and the Chrome browser.

Accessibility Insights for Web

Accessibility Insights is a browser plugin that you can install for Chrome or Microsoft Edge. It helps developers find and fix accessibility issues in web apps and sites.

WINDOWS	MACOS	COMMAND
`Ctrl + Shift + K`	`Shift + ⌘ + K`	Activate the extension
`Ctrl + Shift + 1`	`Shift + ⌘ + 1`	Toggle Automated checks
`Ctrl + Shift + 2`	`Shift + ⌘ + 2`	Toggle Landmarks
`Ctrl + Shift + 3`	`Shift + ⌘ + 3`	Toggle Headings

Test accessible content

Windows Narrator

Narrator lets you use your PC without a mouse to finish common tasks if you're a person who is blind or has low vision. It reads and interacts with things on the screen, like text and buttons. Use Narrator to read and write email, browse the internet, and work with documents.

Shortcut keys

Shortcut keys will help you go to a page when using Narrator. If you would like a full list of keyboard shortcuts, review Appendix B: Narrator keyboard commands and touch gestures.

KEYBOARD SHORTCUT	NOTES
`Windows + Ctrl + Enter`	Start/Stop Narrator
`Esc`	Exit Narrator
Narrator[1] + ←	Change Views in Narrator (previous)
Narrator[1] + →	Change Views in Narrator (next)

[1] By default the Narrator key is set up as both `Caps Lock` and `Insert`

NVDA

NVDA is a free and open-source screen reader for Windows. You can download it from the NVDA Download page. If you would like to learn more about how to use NVDA, review their User Guide.

NAME	DESKTOP KEY	DESCRIPTION
Stop speech	control	Instantly stops speaking.
Pause Speech	shift	Instantly pauses speech. Pressing shift again will continue speech from where it left off (if the current synthesizer supports pausing).
NVDA Menu	NVDA+n	Pops up the NVDA menu to allow you to access preferences, tools, help, and so on.
Toggle Speech Mode	NVDA+s	Toggles speech mode between speech, beeps, and off.
Toggle Input Help Mode	NVDA+1	Pressing any key in this mode will report the key, and the description of any NVDA command associated with it.
Quit NVDA	NVDA+q	Exits NVDA.

Create accessible experiences

ColorBox

Created by Lyft, ColorBox ensures that hue, saturation, and luminosity applied on a color is distinguishable to people with vision disabilities.

Accessibility guidelines

Microsoft put Accessibility Guidelines in place so there's a consistent experience for people who have disabilities. Microsoft is a global company and as such has to meet government compliance with accessibility standards in different nations.

VERSION	STANDARD	GOVERNING ENTITY
WCAG 2.1	Web Content Accessibility Guidelines	W3C
MAS 2.1	Microsoft Accessibility Standards	Microsoft

The Microsoft Accessibility Standards are a superset of the Web Content Accessibility Guidelines. Encompassing the W3C guidelines ensures our standards meet compliance needs in all regions around the world. The following table, assists you in determining the priority of an accessibility bug in your content.

Accessibility Severity Level

Storefront Priority Definitions

Severity 1: Not MAS compliant.

This issue is a showstopper and constitutes a block-ship bug. There are no known workarounds. Must be fixed immediately.

Priority 0: Must fix immediately - blocking with no workaround

Severity 2: Not MAS compliant.

This issue is either non-blocking for core user tasks or blocking for non-core user tasks. Remediation action needed ASAP or within three months post release with an approved exception!

Priority 1: Must fix as soon as possible - blocking some areas;

OR, depending on actual experience,

Priority 2: Should fix soon - before product release.

Severity 3: Not MAS compliant, but low user impact.

Remediation action required in the next major release or the next time the site is updated, whichever occurs first.

Priority 3: Fix if time

Severity 4: Technically a MAS violation, but negligible user impact.

For bugs with this lowest impact, the division accessibility driver can approve a "won't fix" designation.

Priority 3: Fix if time

Disability etiquette

Disability etiquette is a set of voice principles to reduce the risk of offending people with disabilities. Microsoft specifically adheres to the United States standards of people first language (PFL).

> **IMPORTANT**
> You should consider Disability Etiquette when describing how the customer is interacting with web technologies. To ensure that you are using input-neutral terms, review Describing interactions with UI.

*Disability terminology reference guide

Green (preferred/recommended language)	Yellow (acceptable language)	Red (insensitive/offensive language)
Person with a cognitive disability	Developmentally disabled	Retarded/Mental Retardation/Brain damaged
Person with a learning disability	← Trending green Dyslexic person	Reading problems/SPED/Special Ed
Person who is Deaf or Hard of Hearing	← Trending green Hearing disability/ Deaf person	Mute/Hearing Impaired
Person with a physical disability or person with motor disability	Physically disabled	Wheelchair Bound/Crippled
Person with Autism	← Trending green Autistic person	Slow/Dumb/Weird
Person who is blind	← Trending green Blind person	Sight challenged
Person with a mental health disability	Suffers from mental illness	Crazy/Insane/Nuts/Psycho
Person with a disability	Disabled person	Handicapped
Everybody, including people with disabilities	People of all abilities	Differently abled
Person without a disability	Non-disabled person	Normal person/Healthy person

Accessibility guidelines for images and videos

6/25/2021 • 8 minutes to read

This page summarizes the accessibility guidelines for images, videos, and animated GIFs.

Alt text requirements for images

Alternative text (alt text) must follow these guidelines for images that convey information like diagrams and screenshots.

- A minimum length of 40 characters and a maximum length of 150.
- For complex images, supplement alt text with adjacent text in the content or a long description.
- Begin with the type of graphic (for example, "Screenshot of..." or "Diagram showing...").
- Don't start with "Image..." or "Graphic...". Screen readers say this automatically.
- End with a period so the screen reader can identify the end of the alt text.
- Omit extraneous elements like figure numbers or bold/italic formatting.
- Add spaces to acronyms where each character should be read (e.g., "H T M L") but not for words (e.g., "NASA").
- Decorative images (images that do not convey information) and icons do **not** include alt text. Use the `:::image:::` Markdown extension with `type="icon"`, which will output an empty alt attribute.

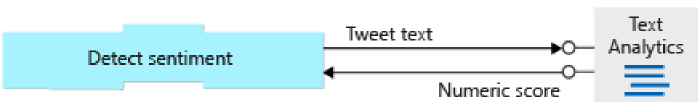
> **TIP**
>
> Use the `:::image:::` Markdown extension for all images. It supports simple images with alt text, complex images with alt text and a long description, and decorative images and icons having no alt text.

Alt text best practices for images

The following recommendations show example alt text for several technical images. For more information, see Microsoft's non-text content training.

Simple images

Consider the following image:



GUIDELINE	EXAMPLE
Express the core idea of the image	Appropriate alt text: *"Diagram showing a Logic App using the detect-sentiment action to invoke the Text Analytics service."*
Don't describe the image literally	Inappropriate alt text: *"Diagram of Text Analytics service usage. Lines with arrows connect the two elements and show text flowing to the service and a numeric score being returned."*
Don't write in the style of a title or caption	Inappropriate alt text: *"Use of the Text Analytics service."*

GUIDELINE	EXAMPLE
Don't duplicate content in the alt text and elsewhere on the page	Lead-in sentence preceding the image: *"The following image shows a Logic App invoking the Text Analytics service:"* Appropriate alt text that is distinct from lead-in: *"Diagram showing a Logic App passing a tweet to the Text Analytics service and receiving a sentiment score in return."*

Complex images

Consider the following image:

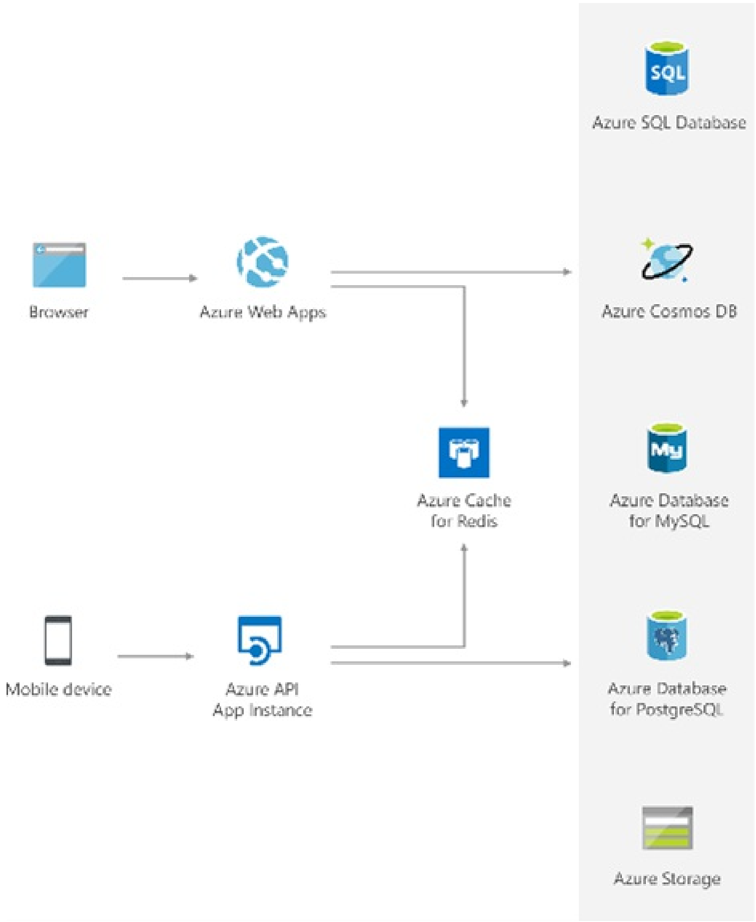


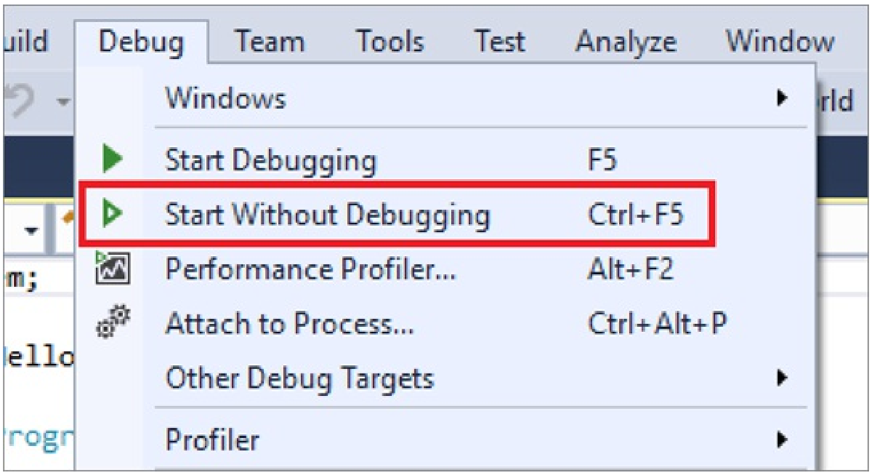
Diagram showing a typical caching architecture using an Azure Cache for Redis in a web application. The architecture has three layers: the client-access layer and the data tier with an Azure Cache for Redis between them. The client layer shows the access pattern for two types of clients: customers using a web browser interact with Azure Web Apps while customers on a mobile device interact with an Azure API App Instance. Both clients pull data from the Azure Cache for Redis when the needed data is in the cache. If the data isn't in the cache, the clients query the data tier directly. The data tier contains five storage locations: Azure SQL Database, Azure Cosmos DB, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Storage.

GUIDELINE	EXAMPLE

GUIDELINE	EXAMPLE
Provide both alternative text and a long description	Appropriate alt text: *"Diagram of a caching architecture with Azure Cache for Redis positioned between the client-access layer and the data tier."* Appropriate long description: *"Diagram showing a typical caching architecture using an Azure Cache for Redis in a web application. The architecture has three layers: the client-access layer and the data tier with an Azure Cache for Redis between them. The client layer shows the access pattern for two types of clients: customers using a web browser interact with Azure Web Apps while customers on a mobile device interact with an Azure API App Instance. Both clients pull data from the Azure Cache for Redis when the needed data is in the cache. If the data isn't in the cache, the clients query the data tier directly. The data tier contains five storage locations: Azure SQL Database, Azure Cosmos DB, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Storage."*

Screenshots of user interface

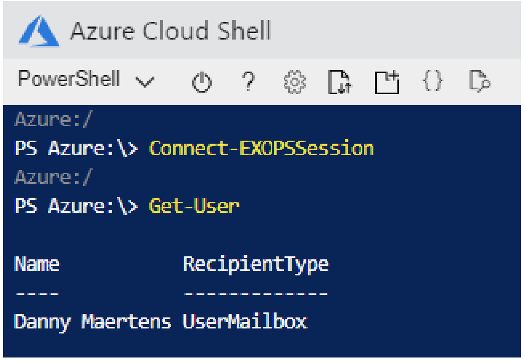
Consider the following image:



GUIDELINE	EXAMPLE
Begin with *"Screenshot of"*	*"**Screenshot of** the Visual Studio Debug menu. The menu entry titled Start Without Debugging and its keyboard shortcut Ctrl+F5 are highlighted."*
Identify the product	*"Screenshot of the **Visual Studio** Debug menu. The menu entry titled Start Without Debugging and its keyboard shortcut Ctrl+F5 are highlighted."*
Identify highlighted areas	*"Screenshot of the Visual Studio Debug menu. The menu entry titled **Start Without Debugging** and its keyboard shortcut Ctrl+F5 are highlighted."*
Identify keyboard shortcuts	*"Screenshot of the Visual Studio Debug menu. The menu entry titled Start Without Debugging **and its keyboard shortcut Ctrl+F5** are highlighted."*
Include the location of UI elements, state of UI controls, and any relevant values in data-entry fields	*"Screenshot of the Visual Studio **Debug menu**. The menu entry titled Start Without Debugging and its keyboard shortcut Ctrl+F5 are highlighted."*

Screenshots of command-line interface

Consider the following image:



Screenshot of the Azure Cloud Shell. The screenshot shows the PowerShell environment being used to execute two commands. The first command is Connect-EXOPSSession, which launches a session. This command has no output. The second command is Get-User, which yields basic information about the current user. The output shown for this command is formatted as a table: the first row lists the column headers of Name and RecipientType. The second row consists of dashes used as a separator between the column headers and the data. The third row gives the user information, the value in the Name column is Danny Maertens, and the RecipientType is UserMailbox.

GUIDELINE	EXAMPLE
Provide both alternative text and a long description	Appropriate alt text: *"Screenshot of the Azure Cloud Shell using the PowerShell environment to launch a session and execute the Get-User command."* Appropriate long description: *"Screenshot of the Azure Cloud Shell. The screenshot shows the PowerShell environment being used to execute two commands. The first command is Connect-EXOPSSession, which launches a session. This command has no output. The second command is Get-User, which yields basic information about the current user. The output shown for this command is formatted as a table: the first row lists the column headers of Name and RecipientType. The second row consists of dashes used as a separator between the column headers and the data. The third row gives the user information, the value in the Name column is Danny Maertens, and the RecipientType is UserMailbox."*
The long description should include all relevant text shown in the screenshot	Appropriate long description: *"Screenshot of the Azure Cloud Shell. The screenshot shows the PowerShell environment being used to execute two commands. The first command is **Connect-EXOPSSession**, which launches a session. This command has no output. The second command is **Get-User**, which yields basic information about the current user. The output shown for this command is formatted as a table: the first row lists the column headers of **Name** and **RecipientType**. The second row consists of **dashes** used as a separator between the column headers and the data. The third row gives the user information, **the value in the Name column is Danny Maertens, and the RecipientType is UserMailbox**."*

GUIDELINE	EXAMPLE
The long description should differentiate between each command and the output of those commands	Appropriate long description: *"Screenshot of the Azure Cloud Shell. The screenshot shows the PowerShell environment being used to execute two commands. The **first command is Connect-EXOPSSession**, which launches a session. This command **has no output**. The **second command is Get-User**, which yields basic information about the current user. The **output shown for this command is** formatted as a table: the first row lists the column headers of Name and RecipientType. The second row consists of dashes used as a separator between the column headers and the data. The third row gives the user information, the value in the Name column is Danny Maertens, and the RecipientType is UserMailbox."*

Cleanup script to add periods to alt text

Users with visual impairments rely on meaningful alt text to get the information provided by an image. Alt text must end in a period so screen readers can pause at the end, making it easier to understand within the flow of an article.

Because this guideline wasn't well known or included in our guidance until recently, much of the older content on `docs.microsoft.com` doesn't include periods at the end of alt text. The Docs Authoring Pack now includes a cleanup script that you can run on a repo to add periods to alt text in bulk.

1. Consult with your repo owner and localization representative to agree on a time to make the bulk update.
2. Open the repo in VS Code with the Docs Authoring Pack installed.
3. Open any file, and then select Alt + M to open the command menu.
4. Select **Cleanup**, then **Add periods to alt text**.
5. The script will run on the repo and add a period at the end of any alt text that doesn't have it. If the alt text is in quotes, the period will go inside the quotes.
6. Review changes, and then submit a PR through the usual process.

Close caption requirements for video

Videos require closed captions:

- Writers must make sure that videos they add to content include closed captions. Most Channel 9 and Red Tiger videos already include closed captions. (Red Tiger is a Microsoft publishing system for videos. It's the preferred system because it supports both accessibility and localization.)
- Writers should contact Channel 9 if closed captions are missing. Channel 9 can usually supply them in a few days.

For more information, see Create and publish a video.

Playback control requirements for animated GIFs

Animated GIFs require a pause before autoplay. This pause capability is in place on docs.microsoft.com. Writers should make sure the pause is working after their article is published.

GDPR references in new or existing technical documentation articles

6/28/2021 • 9 minutes to read

Technical documentation cannot use the terms "General Data Protection Regulation" or "GDPR" outside the context of the CELA-approved includes listed in this article.

General Data Protection Regulation (GDPR) is about protecting and enabling the privacy rights of individuals in the European Union (EU). GDPR establishes strict global privacy requirements governing how organizations manage and protect personal data while respecting individual choice - no matter where data is sent, processed, or stored.

Enforcement of this regulation begins on May 25, 2018. Organizations that fail to comply, whether intentionally or unintentionally, can be fined up to €20 million or 4% of the company's global annual turnover of the previous financial year, whichever is higher based on the type of infraction incurred.

All new GDPR-related content requires frontline CELA review and approval before publication. To find your appropriate CELA representative, see: https://fyc-prod.azurewebsites.net/. We also recommend asking your product group to check with their frontline CELA for review and sign-off, as needed.

Mandatory GDPR guidance for all articles

The following are mandatory guidelines that apply to any feature or functionality impacted by the GDPR requirements.

You must NEVER:

- Interpret the GDPR regulations (by saying things like, "Article 20 of the GDPR says...")
- Imply or promise any action will lead to compliance (by saying things like, "Turning off this feature will keep you compliant..."
- Imply or promise that Microsoft will be compliant. Microsoft will be, but those commitments need to come from contracts and formal statements, not the documentation.
- Use the term GDPR in file names, titles, introduction, or procedural text. All general discussions about GDPR, data deletion through the portal, and breach notification information should be handled through the Service Trust Portal and Compliance sites. Not using this term helps to avoid potential legal implications and future maintenance of content.
- Duplicate content. If content already exists about this feature/functionality, you should link to the existing information.
- Use Microsoft internal data classifications, such as EUII, EUPI, PII, customer content, or telemetry. Instead, you should use the terms: personal data, customer data, customer-authored data, and system-generated logs.

Where does the GDPR content appear?

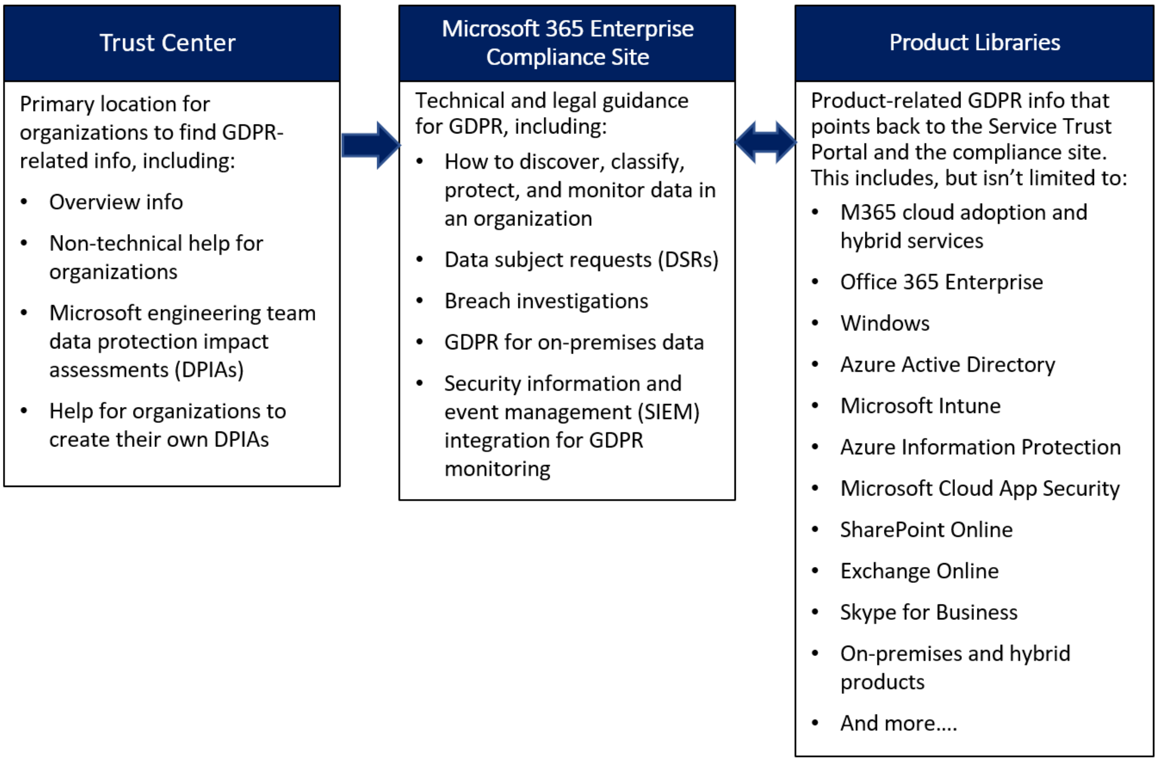
At a high-level, this diagram points to where various GDPR content is located:

- Microsoft Trust Center. Includes general GDPR information, along with information about GDPR compliance best practices, and documentation helpful to your GDPR accountability, such as Data Protection Impact Assessments (DPIAs), Data Subject Requests (DSRs), and data breach notification. This site is targeted towards business decision makers and marketing, to determine the compliance capabilities of our products.

- [Microsoft 365 Enterprise GDPR Compliance site](). Includes legal and technical guidance, based on product groups, pointing to the individual product library articles for more information.

- [Service Trust portal](). Includes all the global GDPR and all-up Microsoft information, pointing into the Microsoft 365 Enterprise Compliance site and the individual product library articles for more information. This site requires authentication to view, and is targeted to technical analyst, auditors, and compliance people.

- **Product libraries (various).** Provides the specific details about features and functionalities, pointing into both the Service Trust portal and the Microsoft 365 Enterprise Compliance site.



Add GDPR info and Include references based on scenario

Typically, the product group or a PM will contact you to let you know that a feature or functionality is impacted by GDPR. This notification means that you'll need to add content about how to export and delete a user's personal data. Currently there are three scenarios:

Scenario 1: Delete and/or export customer data and system-generated logs through the Azure portal

Any GDPR-impacted feature or functionality, where the delete and export of system-generated logs happens through the Azure portal. This scenario includes any feature or functionality that uses the primary Azure portal delete or export functionality (https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-azure#step-5-delete and https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-azure#step-6-export). Anything that is deleted or exported in a different way should be considered "in-product" for both instructions and the Include reference.

You must:

- Document the feature or functionality in place, where you would talk about the feature or functionality normally. However, since the feature or functionality uses the global Azure portal delete and export functionality, you should point to the Microsoft Trust Center and the Service Trust Portal for instructions instead of rewriting the steps in your article.

- Add the following Include file (azure-docs-pr\includes\gdpr-dsr-and-stp-note.md) to the end of your introduction paragraph.

Example of Include in markdown:
`[!INCLUDE [GDPR-related guidance](../../includes/gdpr-dsr-and-stp-note.md)]` with the `../` references matching the hierarchy of your article.

Include file text (CELA-approved):

> **NOTE**
>
> For information about viewing or deleting personal data, see Azure Data Subject Requests for the GDPR. For more information about GDPR, see the GDPR section of the Microsoft Trust Center and the GDPR section of the Service Trust portal.

Scenario 2: Delete and/or export customer data and system-generated logs through an in-product solution

Any GDPR-impacted feature or functionality, where the delete and export of system-generated logs and customer data happens in-product, using the feature or functionality itself. This scenario includes any feature or functionality that uses anything outside of the global Azure portal delete or export functionality (https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-azure#step-5-delete and https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-azure#step-6-export), including any feature or functionality in the Azure portal that uses a different delete or export process.

You must:

- Document the feature or functionality in place, where you would talk about the feature or functionality normally.
- Add the following Include file (azure-docs-pr\includes\gdpr-intro-sentence.md) to the end of your introduction paragraph.

  Example of Include in markdown:
  `[!INCLUDE [GDPR-related guidance](../../includes/gdpr-intro-sentence.md)]` with the `../` references matching the hierarchy of your article.

  Include file text (CELA-approved):

> **NOTE**
>
> This article provides steps about how to delete personal data from the device or service and can be used to support your obligations under the GDPR. For general information about GDPR, see the GDPR section of the Microsoft Trust Center and the GDPR section of the Service Trust portal.

Scenario 3: Deleting customer data requires deleting an underlying Windows component

Any GDPR-impacted feature or functionality, where the user is asked to delete the Windows event log or any other underlying Windows component, to delete customer data.

You must:

- Update an existing feature or functionality article to point users to the Microsoft corporate policy about this type of data removal. The official statement will appear in the Compliance Manager, with secondary content appearing in the M365 Enterprise compliance site (both statements are going to appear on 5/25). This scenario appears to be hybrid-only at this point.
- Add the following Include file (azure-docs-pr\includes\gdpr-hybrid-note.md) to the end of your introduction paragraph.

  Example of Include in markdown:
  `[!INCLUDE [GDPR-related guidance](../../includes/gdpr-hybrid-note.md)]` with the `../` references

matching the hierarchy of your article

Include file text (CELA-approved):

> **NOTE**
>
> For information about viewing or deleting personal data, please review Microsoft's guidance on the Windows data subject requests for the GDPR site. For general information about GDPR, see the GDPR section of the Microsoft Trust Center and the GDPR section of the Service Trust portal.

Use multiple Include references together

If your feature or functionality uses both the global Azure portal and an in-product delete or export functionality, you should use both Include references:

- **Global Azure portal reference (gdpr-dsr-and-stp-note.md).** Use this reference first, after your introduction to the entire article.
- **In-product reference (gdpr-intro-sentence.md).** Use this reference second, after the introduction to the procedural section of the article.

Add includes to other repos

For repos other than azure-docs-pr, create an identical set of include files in the repo and use them as needed in that repo's content. In the gdpr-dsr-and-stp-note.md file, change the Azure Data Subject Requests for the GDPR link to point to the DSR page for your product area. Contact Liz Ross to let her know you're using the include files in your repo (in case something needs to change) or if you have any questions about their use.

Recommendations for writing about personal data

Within individual articles, you should think about covering features and functionality, based on the following types of information.

- **Inventory.** Include information about what personal data exists and where, along with any steps about how to use the feature or functionality to search for and identify personal data.
- **Protect.** Include detailed instructions or links to existing articles about how customers can protect their personal data. This content can also cover how the feature or functionality helps to facilitate data classification to help satisfy customer data subject requests, along with how to use the feature or functionality to display a custom privacy statement.
- **Respond.** Include information about any resources the customer can use to respond to data subject requests. This content should include step-by-step information about how to use the in-product feature or functionality to update personal data, to delete personal data, or how to export personal data. See the Delete and/or export customer data and system-generated logs through the Azure portal and the Delete and/or export customer data and system-generated logs through an in-product solution sections of this article.
- **Audit and reporting.** Include information and relevant links about how a customer can verify they're providing the proper information for data subject requests. This content can include any auditing or reporting features or functionality, which the customer can use to show compliance with data subject request fulfillment.

Exceptions

If you believe you need an exception to this guidance for Content & Learning-sponsored content, you must obtain written approval. Contact lizross for an initial review of your exception. They will determine if additional Azure Operations or CELA review is required.

Pull request review and GDPR mentions

In Content & Learning repositories covered by PRmerger and the PR review process, all pull requests that contain content that mentions GDPR are flagged by PRmerger for human review. This includes articles where the updates in the pull request do not affect the sections of the article that mention GDPR.

The PR review team will review the new or updated parts of the article for compliance with the GDPR content guidelines described in this article. If the new or updated parts of the article do not appear to align to the guidance, the PR review team will send mail to:

- The pull request author
- The content manager for the Azure service or the business approver for the repository (outside Azure)

Subject: GDPR review needed for PR

In the body, the reviewer needs to clearly list the file that has an issue and clearly describe or show how the article does not align to the guidance.

The review must include this text in the message - some authors just need a pointer to bring their content into alignment:

> The CELA-approved guidance for our tech docs is published here - ./contribute-add-gdpr-references.md

The reviewer closes with text that allows the author to explain possible reasons that might justify an exception. If no justification exists, the author needs to fix the article to align to the guidance:

> If you have reasons or legal approvals for not aligning with the guidance, let the folks on this thread know. Otherwise, the article should be brought into alignment with the GDPR content guidance.

Terminology and inclusive language

7/7/2021 • 6 minutes to read

These guidelines address inclusive language in content. They are part of a long-term effort to make our content more inclusive and will be added to over time. The branch name change in GitHub is a separate effort but part of the same commitment.

The guidelines were reviewed across Microsoft: Global D+I, Global Readiness (CELA), Cloud Style Guide, C+L term leads, and members of marketing, engineering, and localization. We also try to align with partners in the industry.

Inclusive language guidelines

The guidelines recommend specific term updates while allowing for some flexibility, depending on context. The goal is consistency throughout the user experience. The note included in the guidelines for references to *slave* in content or code we can't change should be used in technical documentation. The guidelines should otherwise be applied across Microsoft.

TERM	STATUS	OPS VALIDATION STATUS	GUIDELINES
master (git default branch)	Ready to deploy	Enabled	GitHub (which includes the Azure DevOps team) is changing the name of the default repo branch name from master to main.
master/detail	In review	Not enabled	Master/detail refers to a control or control pattern that combines a list with a details view, such as the mail list and mail view in Outlook. Replacements under consideration: list/detail, collection/detail, parent/detail.

TERM	STATUS	OPS VALIDATION STATUS	GUIDELINES
master	Ready to deploy	Not enabled	Microsoft recognizes that the term *master* has numerous meanings and is used in various contexts. Don't use *master* in a master/slave context: **Examples that must be changed**: - *master server* (which implies a slave client). Change to *primary server* or other appropriate term. - *master in, slave out* (for hardware signal names). Change to *serial data in/out* or other appropriate term. **Examples that are OK, where master describes proficiency or an original**: - *master list*, *master slide*, *master plan*, *master detail*, etc. Teams can consider alternatives to *master* in these examples, such as *primary*, *main*, or *template*, but make the change consistently throughout the experience.
slave	Ready to deploy	Enabled (Suggestion)	Use *replica* or alternatives such as *secondary*, *subordinate*, *worker*, *target*, etc., depending on context. - Don't use *slave* in new code or content. - For legacy or third-party content or code that we can't change, add this note before the reference in documentation: **Note**: This article contains references to the term *slave*, a term that Microsoft no longer uses. When the term is removed from the software, we'll remove it from this article.

TERM	STATUS	OPS VALIDATION STATUS	GUIDELINES
master/slave	Ready to deploy	Not enabled	Use *primary/replica* or alternatives such as *primary/secondary*, *primary/subordinate*, *principal/agent*, *parent/child*, *controller/worker*, etc., depending on context. - Don't use *master/slave* in new code or content. - For legacy or third-party content or code that we can't change, add this note before the reference in documentation: **Note**: This article contains references to the term *slave*, a term that Microsoft no longer uses. When the term is removed from the software, we'll remove it from this article.
whitelist	Ready to deploy	Enabled (Suggestion)	Use *allowlist* as the general reference. More specific usage like *safelist* can be used in the appropriate context. - *allowlist* is one word.
blacklist	Ready to deploy	Enabled (Suggestion)	Use *blocklist* as the general reference. More specific usage like *blocked senders list* can be used in the appropriate context. - *blocklist* is one word.

Here is example markdown for the note you can use if needed:

```
>[!NOTE]
> This article contains references to the term *slave*, a term that Microsoft no longer uses. When the term
is removed from the software, we'll remove it from this article.
```

Specific uses of master that we're checking

To minimize false positives, the docs build system checks for very specific uses of the word *master*:

```
master branch
*master* branch
**master** branch
checkout master
checkout *master*
checkout **master**
checkout@master
pull master
pull *master*
pull **master**
push master
push *master*
push **master**
fetch master
fetch *master*
fetch **master**
master server
*master* server
**master** server
master target server
*master* target server
**master** target server
branch: [master]
git pull azure master
git pull azure *master*
git pull azure **master**
git push azure master
git push azure *master*
git push azure **master**
git fetch azure master
git fetch azure *master*
git fetch azure **master**
git pull origin master
git pull origin *master*
git pull origin **master**
git push origin master
git push origin *master*
git push origin **master**
git fetch origin master
git fetch origin *master*
git fetch origin **master**
```

Request an exemption

To request an exemption, send mail to terminologytaskforce@microsoft.com. Let them know the term, the context, the repo, and the paths of the files affected.

Terminology tools

We're developing new tools and updating existing tools to help with terminology updates.

TOOL	STATUS	DESCRIPTION
Sensitive term checking via OPS builds	Deployed	
Multi-search-and-replace	Available	Uses a configuration file to replace multiple terms. To use the tool on your repo, please contact Alex Buck (abuck).

TOOL	STATUS	DESCRIPTION
TermFixer for links	Available	A command-line tool Mike Jacobs developed, looks for GitHub links in your articles. For each one it finds, it queries GitHub to check whether the default branch has been renamed yet. If it has, it updates the link to point to the new default branch name. TermFixer usage and installation instructions
Acrolinx	Update pending	The Acrolinx tool is being updated with the latest term guidelines.
Policheck	Update pending	The Policheck terms list is being updated with the latest term guidelines.

Frequent asked questions (FAQ)

Q: Can I go ahead and replace sensitive terms in my content right now?

A: Yes! Just make sure there's an approved replacement. If the term's status is "Ready to deploy," you can make the replacement. If the status is "In review," hold off for now.

Q: What's our deadline?

A: We'd like to get these terms fixed as soon as possible. That said, there are a lot of them in a lot of repos. Please make fixing these terms a part of your Q3 planning. As we get closer to fixing all the terms, we'll set a deadline for turning the sensitive-language build validations from suggestions into warnings.

Q: I found a term that's not on the list... what should I do?

A: Send mail to terminologytaskforce@microsoft.com and tell us about the term and where it appears.

Q: My Doc repo's default branch is named "master"--when I can change it?

A: The Docs build system now supports "main" as a default branch. Khairun Jamal's team has developed tooling to automate changing the default branch from "master" to "main" in our publishing repos. Please wait until you hear from Khairun before changing the default branch name in your Docs repos.

If you need to change a default branch name before that, reach out to terminologytaskforce@microsoft.com.

Q: I found a sensitive term in my content that is actually in the product, how do I fix it?

A: If you are the content owner for the product, reach out to the usual engineering or PM contact on the product team. Provide a link to the page https://aka.ms/terminologytaskforce, that explains the goal and scope of the project, as well as guidance for changing the terms. Also include the link to the content the term was found in, for reference, and a description of where in the product the term is located, if possible. For any further questions, feel free to direct them to terminologytaskforce@microsoft.com.

If you don't know the product owner, email terminologytaskforce@microsoft.com for help.

See also

Here are other sources for additional information on the change to *master*:

- GitHub guidance for renaming branches
- Open Source Hardware Association
- Wired

- ZDNet re: MySQL
- ZDNet re: GitHub
- ZDNet re: Linux
- Drupal.org
- Washington Post

How to create a term list

7/1/2021 • 3 minutes to read

This article provides guidance for making a term list as part of the Content & Learning terminology management process. You create the term list for your doc set as an Excel spreadsheet and publish it as an article in the contributor guide. There it functions as a reference for writers and partners, helping to ensure that they use the terminology correctly and can provide clear and concise explanations. Here's a sample term list spreadsheet:

Term	Definition	Usage
Blob Storage	An Azure service that stores unstructured data in the cloud as objects/blobs. It is part of Azure Storage.	First mention: Azure Blob Storage Subsequent mentions: Blob Storage Always spell out the name. No acronym or abbreviation. Example: Blob Storage is optimized for storing massive amounts of unstructured data.
point-in-time restore	A feature that provides protection against accidental deletion or corruption by enabling you to restore block blob data to an earlier state.	Lowercase. Do not use the acronym PITR. Example: You can use point-in-time restore to restore one or more sets of block blobs to a previous state.

Although the following steps are presented in a sequential order, feel free to do them in whatever order works best for you.

Select terms to manage

Find and list terms that could be misused or misunderstood. "Terms" may be individual words or phrases. Here are some ways to identify terms to manage:

- Scan existing docs for terms that might be problematic:
  - Published docs for your product or service.
  - Docs written by the product unit and published only in the product repositories.
  - Source code comments in the product repositories.
  - Blogs written by the product unit.
  - The UI of the product or service.
  - Competitors' doc sets for products or services that are similar to yours.
- Check the style guides for entries related to your product or service.
- Use SEO tools such as BrightEdge & Google Search Console to find search terms that people use to find your docs.
- Ask Localization to provide terms for your product or service that already exist in Term Studio.
- Use tools such as DocuTune and Python NLTK to extract keywords and key phrases from existing content.

After you've created a list of terms, review the list and remove any terms that you're fairly confident:

- Have a low risk of being misused by writers and partners.
- Will be recognized and understood correctly by your intended audience.

Create a term list spreadsheet

- Create an Excel spreadsheet with a heading row that names the first three columns "Term," "Definition," and "Usage."
- Enter your terms in the first column.

When entering text into spreadsheet cells:

- Be aware that when you import the spreadsheet to Markdown, text styling such as bold and italics is lost.
- Use `Alt+Enter` to insert line breaks in a cell.

Provide definitions

Fill out the definition column, as much as possible following this pattern for each term:

- Explain what it is and what it does.
- Provide another sentence or two for further clarification, if needed.
- Provide information about alternate meanings depending on context, if needed.
- Provide a link to an article that explains the term, if available.

Here's an example spreadsheet row with a clear and concise definition:

Term	Definition	Usage
device twin	A JSON document that stores device state information such as metadata and configuration data. Azure IoT Hub maintains a device twin for each device that connects to a hub and keeps the device twin synchronized with the real device to monitor and manage it. https://docs.microsoft.com/azure/iot-hub/iot-hub-devguide-device-twins	Lowercase.

In this example, notice that the definition:

- Doesn't start by repeating the term:
  - This: "A JSON document..."
  - Not this: "A device twin is a JSON document..."
- Starts by explaining what it is ("A JSON document") and what it does ("stores device state information").

Provide usage guidance

Fill out the Usage column, adding the following information for each term. All of these elements are optional:

- Capitalization.
- First use vs. subsequent uses.
- Approved and unapproved acronyms and abbreviations.
- An example sentence that uses the term, if needed.

Use `Alt+Enter` to insert line breaks in a cell.

Here's an example:

Term	Definition	Usage
Blob Storage	An Azure service that stores unstructured data in the cloud as objects/blobs. It is part of Azure Storage.	First mention: Azure Blob Storage Subsequent mentions: Blob Storage Always spell out the name. No acronym or abbreviation. Example: Blob Storage is optimized for storing massive amounts of unstructured data.
point-in-time restore	A feature that provides protection against accidental deletion or corruption by enabling you to restore block blob data to an earlier state.	Lowercase. Do not use the acronym PITR. Example: You can use point-in-time restore to restore one or more sets of block blobs to a previous state.

Request reviews

- Send the spreadsheet to the Term council first, and address feedback.
- Send the spreadsheet to writers and partners, including:
  - The doc team
  - Engineering
  - Marketing
  - Localization
  - Cloud advocates
  - Customer Experience & Success (CE&S)
- Address feedback from writers and partners.

Convert to Markdown

- Create a Markdown file named *<doc set name>-terms.md* in the docs-help-pr repo, *help-content/contribute* folder.

- Add a metadata block and H1:

  ```
  ---
  title: Term list for <doc set name>
  description: Definitions and usage instructions for terms used in the <doc set name> documentation.
  author: <your GitHub ID>
  ms.author: <your MS alias>
  ms.date: mm/dd/yyyy
  ms.prod: <your product>
  ms.topic: contributor-guide
  ms.custom: internal-contributor-guide
  ---

  # Term list for <doc set name>
  ```

- Install the Visual Studio Code add-in Excel to markdown table.

- Copy the data from Excel, and paste into the *.md* file

- Select the text and press `Shift+Alt+v` to convert.

- Add an entry to the TOC (*toc.yml*) in the Resources/Terminology node.
- Create a pull request, get it reviewed and merged.

Next steps

Send the term list link to the doc team and partners.

Legal content

7/9/2021 • 2 minutes to read

Documentation in large repos like 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. These mechanisms 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 Corporate, External, and Legal Affairs (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. Funding those translations is also the product team's responsibility. 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 isn't product or service specific, use `ms.prod: legal`
  - All files should use the generic `ms.topic: article`

We recommend that all files use another metadata value: `layout: ContentPage`. It creates a flat-page experience with no TOC. The exception to the rule is groups who want to create a contextual TOC to make legal articles appear within the TOC of a product or service guide.

Don't create a regular TOC within the DocsLegal-pr repo. The root TOC and breadcrumbs are out of context and will confuse readers.

Publication process

Follow these steps to contribute source content to DocsLegal-pr, which publishes to the en-us docs.microsoft.com site.

1. Draft your Markdown article 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're ready for the content to go live.

   - If you require special (timed) or rushed publication, don't #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 isn't ready to be merged. It's your responsibility to 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's important that the English content is published **before** you move forward with updates to translated content.

Use PoliCheck to identify sensitive language

6/28/2021 • 2 minutes to read

Policheck is a content-scanning tool designed to check for sensitive geopolitical terms, profanity, and trademark terms in Microsoft products.

Learn more: https://microsoft.sharepoint.com/sites/globalreadiness/SitePages/PoliCheck.aspx

When to use PoliCheck

- During localization
- As an IC, if you want immediate scan results on a repo

> **NOTE**
>
> As of July 2020, the Content Production Services team is phasing into a pilot where they run Policheck on this scope and frequency.

Prerequisites

1. Install PoliCheck from http://toolbox/policheck.
2. Ensure that your local git repository is up to date.
3. Verify that you are in the repository that you want to test in GitBash.

Run PoliCheck

You can run PoliCheck against content in a GitHub repository. You can scan an entire repository, a folder, or a single markdown file. Or, repository administrators can periodically run PoliCheck on the local copy of a repo.

1. Launch PoliCheck.
2. Click **Options** > **File Type Settings**.
3. In the **File Type Extensions** box, make sure **Pure Text Files** is selected. The markdown extension (.md) is part of this group of file extensions.
4. Close the File Type Settings dialog box - if you are prompted to save, click **Yes**.
5. Click **Task** > **New**, and then select **File Scan** in the **Task Type** dialog box.
6. On the **Step 1** tab, name the task. Then, if you want to test a folder in your Git repository, select Folder in the Target Area. If you want to test a specific file, select File in the Target area. Specify the location in which you want to save the results. Click **Next**.
7. On the **Step 2** tab, select **English**, all severities, and all term classes. Click **Next**.
8. On the **Step 3** tab, select **Default File**, and click **OK**.
9. In the Policheck console, right-click the scan task you just created, and then click **Start**.

Policheck displays the results in the right pane. To export the results to an XML file that you can then open in Excel, click **Actions** > **Export Scan Results** > **File**.

Rerun PoliCheck

To run the same task again, right-click the existing task and select `Start`.

Contributor guide operating principles

5/10/2021 • 2 minutes to read

We believe our contributor guides are content that deserve the same care and attention we give to all our work. Our decisions on what to include and how it's organized are informed by the principles of **Find, Trust, Transfer**. These principles are intentionally oriented from the guide consumer's point-of-view. We want contributors to quickly find the right article, trust that it's up to date, and apply what they learn to their own writing.

Find

We have the right content and it's easy to navigate.

- We reuse and link content, we don't duplicate.
- No more than three layers deep in the TOC where possible.
- Emphasize consistency in organization and format.
- When a user finds a gap in the guidance it's easy to fill that gap, and get approval for the change.
- It's clear what each set of guidance does and doesn't apply to. By default, guidance applies globally unless otherwise indicated.

Trust

Our content is kept up to date and shows approved procedures. Errors are fixed quickly.

- Anyone can suggest a change to an article, but page owners approve or get approval for changes as needed.
- Ensure appropriate notice to users before making substantive changes. Make clear to those contributors who propose changes what the response timeline (SLA) is likely to be.
- Clearly distinguish between recommended guidelines and required rules.
- Clearly distinguish which organizations or types of content a set of guidelines applies to.
- We have clear, current contacts for each article.

Transfer

We include a mix of procedures, concepts, and examples to enable our consumers to apply the ideas to their own work.

- An article is for one user doing one thing.
- Focus on the task and support it with necessary conceptual information and examples.
- Document the preferred/supported way to do something, not all the possible options.

Contributor guides content standards

6/24/2021 • 2 minutes to read

The content on docs.microsoft.com should speak with one voice, but different situations call for different conventions. Striking a balance between speaking with one voice and speaking to different communities requires clear decision-making processes. By establishing content standards that scope responsibility to specific organizational levels, we maintain consistency and avoid revisiting the same decisions eternally.

The goal of the contributor guide content standards is the consistent appearance and function of internal-facing contributor guidance across content sets and modalities. This consistency standardizes the experience from one content set to another.

Federated content standards

The approach uses a government analogy to distinguish three hierarchical levels for implementing content standards across the docs.microsoft.com platform: federal, state, and local.

Federal level

Standards at the federal level apply to all users of the docs.microsoft.com platform.

Federal: platform

State level

Standards at the state level support the ways that individual business organizations use the platform. The guidance for one organization wouldn't necessarily be the same as for another since the audiences may differ in their needs.

State: business org

Local level

Standards at the local level apply to variations for a service/product. For example, this level of guidance aligns with the "repo-specific" section of the Docs contributor guide TOC.

Local: product/service

Templates and style guides

Style guides help keep a content set consistent when there are different products, contributors, and editors. They provide distributed governance for things like terminology, grammar, and usage (example: footnotes).

Templates help keep a content set consistent by providing a pre-patterned starting point for a specific content format.

Both tools support consistency and reduce the decision-making load on contributors when creating content.

List of available contributor guides

6/28/2021 • 2 minutes to read

C&L

GUIDE	CONTACT	GITHUB ALIAS
Docs contributor guide (Internal)	Jim Parker	Jim-Parker
Docs contributor guide (External)	Jim Parker	Jim-Parker
Learn contributor guide	Jim Parker	Jim-Parker
Repo admin guide	Jim Parker	Jim-Parker
Localization guide	Jason Card	JasonCard
Tools & process guide	Jim Parker	Jim-Parker
Content product service (CPS) guide		

Other

GUIDE	CONTACT	GITHUB ALIAS
Apex Publishing Desk Documentation		
Applied AI Docs	Erik Hopf	erhopf
Azure Databricks		
Business Applications Content Experience Team (BACX) guide		
Cloud Studio		
Contributor guide for Azure Data Explorer (Kusto)		
Dev Div		
Developer relations team guide		
Docs style guide		
Engineering		
Engineering: new-hope		

GUIDE	CONTACT	GITHUB ALIAS
Experimentation platform		
DevRel UX	Adam Kinney	
Event Learning Experience		
MARVEL Authoring Guide for Docs		
Microsoft open source policies & strategy guide		
Microsoft open source program tools guide		
PowerShell Contributor Guide	Sean Wheeler	sdwheeler
Product Launch Readiness (PLR) guide		
Windows authoring guide		
World Wide Learning (WWL) guide		
PLR Launch Criteria		