

Contents

[Learn Contributor Guide](#)

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

[Frequently asked questions](#)

[Get started with GitHub at Microsoft](#)

[Set up your GitHub account](#)

[Install authoring tools](#)

[Fork and clone a repository](#)

[Add/update Learn content](#)

[Create a pull request](#)

[Use our tools](#)

[GitHub commands](#)

[Microsoft Learn repos](#)

[Create code repos](#)

[Troubleshooting build issues](#)

[Track content in Azure DevOps](#)

[Single-source PowerPoint transformation \(Particle\)](#)

[Module transformation rules](#)

[Transformation tooling](#)

[Authoring best-practices](#)

[Template creation](#)

[Previewer for VS Code \(Quark\)](#)

[Author starter kit](#)

[Authoring guidelines](#)

[Content requirements](#)

[Module planning](#)

[Module design](#)

[Module structure](#)

[Standard modules](#)

[Introduction modules](#)

[Choose modules](#)

[Unit structure](#)

[Structure learning content](#)

[Structure exercise content](#)

[Detailed guidelines](#)

[Quick reference](#)

[Titles](#)

[Introductory summaries](#)

[Learning objectives](#)

[Prerequisites](#)

[Scenarios](#)

[Introductions](#)

[Knowledge checks](#)

[Summaries](#)

[Style guides](#)

[File name and folder path guidelines](#)

[Add or edit Learn content](#)

[Create content](#)

[Track content development in Azure DevOps](#)

[Create compliant content](#)

[Create a module](#)

[Scaffold manually](#)

[Scaffold with templates](#)

[Create a unit](#)

[Create a unit](#)

[Create learning content](#)

[Add a knowledge check](#)

[Add a lab](#)

[Lab funding model](#)

[Preventing fraud](#)

[Add an Azure sandbox-based lab](#)

[Understand the Sandbox limitations](#)

- [Add task validation \(optional\)](#)
- [Add a VM-based lab](#)
 - [Get help creating a LOD-based VM lab](#)
 - [Create your own LODS-based VM lab](#)
- [Add a Dynamic 365 Lab](#)
- [Add a Try.NET C# lab](#)
- [Guidance for creating technical exercise content](#)
- [Add a Notebook sandbox lab](#)
- [Add zone pivots](#)
 - [Create or modify a zone pivot](#)
- [Add a video](#)
 - [Choose a video type for your Learn module](#)
 - [Create and publish a video](#)
- [Create a screenshot](#)
- [Create conceptual art](#)
- [Create an animated GIF](#)
- [Create a lightbox](#)
- [Add code samples](#)
- [Add links](#)
- [Add alert boxes](#)
- [Add lists and bullets](#)
- [Include reusable content](#)
- [Create a learning path](#)
- [Create a topic landing page](#)
- [Request achievement images](#)
- [Refine content](#)
 - [Hide content](#)
 - [SEO Basics](#)
- [Review and publish content](#)
 - [Optimize for PR review process](#)
 - [Pre-publish checklist](#)
 - [Formal editorial review](#)

- [How to build Learn URL](#)
- [PR review checklist](#)
- [Launch new groups](#)
- [Support content](#)
 - [Add lab disabled notification to modules](#)
 - [Add new Azure DevOps area path](#)
- [Learner support](#)
 - [Learner support options](#)
 - [Triage and support workflow](#)
 - [Internal support escalation](#)
 - [Vendor triage process](#)
- [Event support](#)
 - [How to run an event](#)
- [Customer Feedback Dashboard](#)
- [Propose a change to content](#)
- [Retire content](#)
 - [Implement redirections](#)
- [View portfolio and roadmap](#)
 - [Content portfolio and roadmap](#)
- [View data](#)
 - [Power BI Learn reports](#)
 - [Microsoft Docs Metrics \(MDM\) tool](#)
 - [Kusto queries](#)
 - [Report on individual user \(employees and partner\) data](#)
- [Tools and reports](#)
 - [Microsoft Learn PowerBI report](#)
- [Customer Research](#)
 - [How to do customer research](#)
 - [How to select a research method](#)
 - [How to get a UserTesting license](#)
 - [How to do customer interviews](#)
 - [How to do surveys](#)

- [Customer feedback - Azure Pulse program](#)
- [Customer feedback - PSAT program](#)
- [Receive GitHub issues and public PR reports](#)
- [Host a hackadoc](#)
- [Run experiments](#)
- [Localization](#)
 - [Global Experiences Content Localization team](#)
 - [Localization FAQ](#)
 - [Checklist for localization](#)
 - [Avoid force push](#)
 - [Guidance for legal articles](#)
 - [How the loc team selects content to localize](#)
 - [Source escalation](#)
 - [Writing for translation](#)
 - [Provide Feedback to DevRel Localization Team](#)
 - [How to Avoid Over-Localization](#)
 - [How to specify the localization scope for an image](#)
 - [Video Caption Localization
 - \[Request Video Caption Localization\]\(#\)
 - \[Guidelines for efficient English video caption\]\(#\)](#)
- [Resources](#)
 - [Learn platform and projects](#)
 - [Learn project vision document](#)
 - [Creating new Learn repos](#)
 - [Microsoft Learn interactivities](#)
 - [Reference guide for performing Learn pull request \(PR\) reviews](#)
 - [Customer feedback AzDO field uses \(triage and fixing/resolving\)](#)
 - [Create Learn Contributor Guide PDF for external sharing](#)
- [Docs platform and technical writing](#)
 - [Accessibility guidelines \(CG\)](#)
 - [Docs writing principles \(CG\)](#)
 - [Quick data references and job aids \(CG\)](#)

[Supported languages \(CG\)](#)

[Art & multimedia gallery guidance](#)

[GDPR references in content \(CG\)](#)

[Content process](#)

[Content process overview](#)

[Planning content](#)

[Designing content](#)

[Writing content](#)

[Reviewing and publishing content](#)

[Maintaining content](#)

[How-to write for part-time content authors](#)

[Learn GitHub commands](#)

[Module design cheat sheet](#)

[Docs quick reference](#)

[Instructional design and learning](#)

[Definitions of learning terms](#)

[Learning principles](#)

[Overview of UID](#)

[DevRel organization](#)

[Who owns what in Docs design & engineering](#)

[Unified content model \(CG\)](#)

[Unified content model FAQ \(CG\)](#)

[Tools](#)

[Sandbox Mapping Tool](#)



The mission of Microsoft Learn is to provide best-in-industry, free, interactive training that teaches customers, partners and employees how to use Microsoft products and platforms, demonstrate best practices, and prepare learners for Microsoft certifications.

What is Microsoft Learn?

Microsoft Learn combines short step-by-step tutorials, browser-based interactive coding/scripting environments, and task-based achievements to provide technical audiences who want to learn Microsoft technologies a free interactive way of learning to help them advance their technical skills.

<https://msit.microsoftstream.com/embed/video/9a00e9bd-f2a9-4a14-8f67-9dbadbe36e69?autoplay=false&showinfo=true>

Notable characteristics:

- **Learning that respects your time:** Step-by-step bite-sized tutorials and modules aligned to role-based certification make it easier to dedicate time for learning.
- **Learning by doing:** Interactive coding environments right in the browser provide a friction-free hands-on experience.
- **Easy to navigate:** Follow guided learning by product, skill level, or job role that is easy to navigate and help you solve your business challenges or prepare for Microsoft certification.
- **Get recognized with achievements:** Complete modules, test your knowledge and earn achievements to recognize your

skills. Share achievements through social media.

- **Learn in ways best for you:** Select from free self-paced quick tutorials and hands-on learning in Microsoft Learn, and get connected to structured online courses (free from Pluralsight for Azure roles) and instructor led classes from Microsoft Learning Partners. Learning across modalities will align to role-based certification.
- **Deep integration into documentation quickstarts and tutorials:** Learn is an extension of the Microsoft Docs portal, with a unified user profile between Docs, Learn, and Q&A forums.

Production sites

Microsoft Learn's primary URL is <https://docs.microsoft.com/learn>. There are two other URLs, <https://microsoft.com/learn> and <https://docs.com/learn>, that also redirect to it.

Frequently Asked Questions (FAQs)

View the [Microsoft Learn FAQs](#) article for a list of frequently asked questions about Microsoft Learn.

Learn Contributor Guide

The Learn Contributor Guide provides detailed guidance on how to make Learn a consistent and smooth learning experience for our learners. The following sections on this site document the guidelines, requirements, and procedures:

SECTION	DESCRIPTION
What's New	The Learn Contributor Guide is a living site and is continuously updated to add new information as well as enhance existing content What's new section gives a list of recent changes to the Learn Guide
Learn Walkthrough	Explore Learn from a learner's perspective , see all the features of Learn including interactivities , learner validation , ratings/feedback , and support infrastructure . This section also gives a behind the scene overview of Learn. See how Learn repo is structured and about content structure within a learn repo. All Learn content is designed with specific roles and job task, see what roles and job tasks we currently cater to.
Get started	Prepare for writing your Learn content by getting acquainted with our tools and content management system. We use GitHub as a content management system and all Learn content is stored in GitHub repositories . See the Get started section to learn all about setting up your GitHub account and installing required tools.
Authoring guidelines	Learn content is governed by strict guidelines to ensure that we provide consistent learning experience. Our detailed content quality and standards guidelines clearly define the basic requirements that must be met before publishing the content. <ul style="list-style-type: none">- The authoring guidelines section documents the detailed instructional design standards that help you to create quality learning content.- A quick reference checklist gives you a list of all ID guidelines at a glance.- Detailed guidelines on how to write title, introductions, scenarios, knowledge checks, prerequisites, and summaries are also explained.

SECTION	DESCRIPTION
Create content	<p>This section gives you detailed guidance on tactical steps on how to scaffold the files for your Learn content. All Learn content follows a specific structure that includes:</p> <ul style="list-style-type: none"> - Unit: a single page of content. This can be a conceptual page that covers a task, an interactive exercise to get the learner to do a task, or a quiz to test comprehension of a task. See how to scaffold different types of unit: textual, knowledge check, video, labs, zone pivots. - Module: a set of related units (pages) that cover one or two job tasks, providing all the knowledge necessary to perform the task(s). See the guidance on how scaffold module files. - Learning paths: an organized way to go through a set of related modules. See process for creating a learning path.
Review and publish	<p>All learn content must go through a formal editorial. This section explains the requirements for these reviews. It also explains the activities involved when launching a new group on Learn.</p>
Support	<p>Learn content has a well-defined support framework. Learners have several support options available, from online troubleshooting guides to the option of submitting support queries via email. A well-defined triage process is followed to track and manage all customer reported issues. The Support section of the Learn Guide also provides information on Microsoft Learn events.</p>
Retire content	<p>All Learn content must be continually maintained and updated for freshness. Any obsolete content, or content that does not have proper maintenance plan in place is retired and taken down from the live site. However, retiring Learn content is not as simple as pulling it down from the live site, all published content reward achievements for completion and removing it from the live site without following proper redirection steps might lead to learners losing their achievements and progress. The Retire content section lays down the procedures to remove published content and implementing proper redirections.</p>
View data	<p>The View data section gives an overview of data and metrics used to track Learn content. Explore different ways to view metrics for your published content.</p>
Resources	<p>Any additional information related to Learn platform, Docs platform, Instructional design, DevRel organization, and tools are housed under the Resources section.</p>

What's new and updated in Learn Guide

1/14/2022 • 22 minutes to read

July 2021

ARTICLE	CHANGE	DESCRIPTION
Single-source content (Particle)	New	Overview of the single-source content strategy.
Single-source: module-to-PowerPoint transformation rules	New	Rules for transformation of a Learn module into PowerPoint slides.
Single-source: PowerPoint transformation usage	New	Instructions on how to run the transformation tooling to generate PowerPoint slides from Learn content.
Single-source content: PowerPoint authoring best practices	New	Recommended authoring practices for Learn content that transform into effective PowerPoint slides.
Single-source: PowerPoint template creation	New	Rules for creating a PowerPoint template that is consumable by the Particle transformation tooling.
Single-source: PowerPoint previewer for Visual Studio Code	New	Instructions on how to install and use the PowerPoint previewer for Visual Studio Code.

April 2021

ARTICLE	CHANGE	DESCRIPTION
The Learn Notebooks sandbox	New	Instructions on how to add a Jupyter Notebooks sandbox to a Learn module.

March 2021

ARTICLE	CHANGE	DESCRIPTION
How to structure "Introduction to 'product'" modules	Updated	Clarified that the module pattern is for a single product. Reduced verbiage to shorten the page.
The Learn Azure sandbox	Updated	Revised article to give specific guidance on using the Azure sandbox in a Learn module.

ARTICLE	CHANGE	DESCRIPTION
The Try.NET experience	New	Instructions on how to add Try.NET to a Learn module.
The Learn Dynamics 365 sandbox	New	Instructions on how to add the Dynamics 365 sandbox to a Learn module.

February 2021

ARTICLE	CHANGE	DESCRIPTION
Optimize for the pull request review process	New	Includes best practices for PR submissions and guidance for how to efficiently work with the PR review team.

December 2020

ARTICLE	CHANGE	DESCRIPTION
Guidance for creating new Learn code repos	Updated	Revised for new OSS portal process.

October 2020

ARTICLE	CHANGE	DESCRIPTION
Add a new Azure DevOps area path	New	Instructions for creating new Azure DevOps area paths and mapping incoming customer feedback from modules to AzDO area path values.
Guidance for creating technical exercise content	New	Created a doc that has a bunch of tactical guidance around writing exercises
Add an interactive lab to a MS Learn module	Updated	Updated guidance on adding labs to MS Learn content.

September 2020

ARTICLE	CHANGE	DESCRIPTION
Partner Engagements	Deleted	Removed old resource article listing partner engagements.
Engagement Overview	Deleted	Removed article about Microsoft Learn engagement process (being replaced by new articles in Tools and Processes Guide).

ARTICLE	CHANGE	DESCRIPTION
Phase 1 - Engage and assess	Deleted	Removed article about Microsoft Learn engagement process (being replaced by new articles in Tools and Processes Guide).
Phase 2 - Onboarding	Deleted	Removed article about Microsoft Learn engagement process (being replaced by new articles in Tools and Processes Guide).
Phase 3 - Content Development	Deleted	Removed article about Microsoft Learn engagement process (being replaced by new articles in Tools and Processes Guide).
Phase 3 - Ongoing Support	Deleted	Removed article about Microsoft Learn engagement process (being replaced by new articles in Tools and Processes Guide).
Onboarding checklist	Deleted	Removed resource article with checklist of onboarding activities.
Create and publish a video	Updated	Video guidance replaced by this article
Author Starter Kit	Updated	Replaced link to old module proposal form with new universal contribution request intake form for all of docs.microsoft.com (inclusive of Learn, replaces all other/old forms). Also added verbiage that people should not continue through the setup process until they've been approved, as they should do this work during those phases.
Microsoft Learn GitHub Repositories	Updated	Changed Microsoft 365 primary content contact

July 2020

ARTICLE	CHANGE	DESCRIPTION
[Create a topic landing page] (/help/learn/create-a-topic-landing-page)	New	Article about creating topic landing pages.
Create and publish a video	Updated	Video guidance replaced by this article.

June 2020

ARTICLE	CHANGE	DESCRIPTION
Additional reporting options for Learn data	Updated	Replaced DevRel-specific "CGA dataout" cluster with new public "CGA Kusto Public" cluster.

May 2020

ARTICLE	CHANGE	DESCRIPTION
ID Guidelines	Updated	Added explanatory video to each guideline.

January 2020

ARTICLE	CHANGE	DESCRIPTION
AzDO root-cause customer feedback field instructions	Updated	Added instructions for using the new <i>Cause of issue</i> field added to Customer Feedback work items in Azure DevOps.
AzDO root-cause field value definitions]	Updated	Added field value definitions for the <i>Cause of issue</i> field.
Added new module update instructions]	New	Added page on updating a module.

December 2019

ARTICLE	CHANGE	DESCRIPTION
Power BI reports and dashboards	New	New article describing the Power BI reports and dashboards for Learn data
Additional reporting options	New	New article based off the Contributor Guide article on additional reporting options for Learn data
PR review checklist	New	New article based off the Contributor Guide article on PR review checklist.
Create compliant content	Updated	Added the GDPR references in content guideline and included the link to the Contributor guide article
SEO basics for Learn	New	New article to describe SEO basics for Learn content. Includes links to Contributor Guide links for details.
AzDO Customer Feedback work item field definitions and uses	New	Added intentions and team uses for the various work item fields.

ARTICLE	CHANGE	DESCRIPTION
Create Learn Contributor Guide PDF for external sharing	New	New article explaining the process to create PDF for external sharing

November 2019

ARTICLE	CHANGE	DESCRIPTION
ID Planning: Module Design	Updated	Re-targeted documentation to vendor-provided review service from ID-team-provided review.
Microsoft Learn FAQ	Updated	Updated the article to add answers to few questions.

October 2019

ARTICLE	CHANGE	DESCRIPTION
Hide content	New	New article with guidance on how to use the hidden feature to hide/unlist module and learning path content from browse and search engines.
Author starter kit	New	New article for helping authors get started writing and publishing content for Learn.
Pre-publish checklist	New	New article that contains all of the things authors should check for before publishing content - full list of authoring requirements and PR Review checklist.
Module design template	Updated	Updated the template to fix reference links, changed the knowledge check to not be a standalone unit and added a Notes section.
ID Service: Module Proposal Review	Updated	Re-organized for ease of use and listing current services.
ID Service: Module Design Review	Updated	Re-organized for ease of use and listing current services.
Implement redirection for retired Microsoft Learn content	Updated	Added more metadata descriptions for clarity and referenced other retirement documents in article.
Create a learning path	Updated	Added information about using the generic badge.

ARTICLE	CHANGE	DESCRIPTION
Add an Azure Sandbox Lab	Updated	<ul style="list-style-type: none"> * Updated policy information for triplecrown4 policy to include new list of services supported by policy. * Updated instructions to avoid the <rgn> tags from being converted incorrectly. * Added debugging note for possible testing issue.
Multiple articles	Updated	Removed references to using ROBOTS:NOINDEX, as this is no longer needed since the system automatically implements.
Multiple articles	Updated	Fixed inconsistent capitalization of 'labid:' metadata property.

September 2019

ARTICLE	CHANGE	DESCRIPTION
Add links	New	New article, based off Docs Contributor Guide article, explaining how to add links to Microsoft Learn content.
Add alert boxes	New	New article, based off Docs Contributor Guide article, explaining how to add alert texts to Microsoft Learn content.
Add lists and bullets	New	New article, based off Docs Contributor Guide article, explaining how to add numbered and bulleted lists to Microsoft Learn content.
Add reusable content	New	New article, based off Docs Contributor Guide article, explaining how to reuse text blocks in Microsoft Learn units.
Create compliant content	New	New article that explains making content compliant for accessibility, instructional design, style guides, etc.
Add or edit Learn content	New	New article, based off Docs Contributor Guide article, specific for adding or editing Learn content.
Fork and clone the Learn repos	New	New article, based off Docs Contributor Guide article, specific for forking and cloning the Learn repos.

ARTICLE	CHANGE	DESCRIPTION
Improve your Learn pull request	New	New article, based off Docs Contributor Guide article, specific to improving pull requests for Learn and avoiding issues.
Install Learn authoring tools	New	New article, based off Docs Contributor Guide article, to install authoring tools.
Setup your GitHub account	New	New article, based off Docs Contributor Guide article, about setting up your GitHub account.
Interactivity in Microsoft Learn	New	New resource article about interactivity in Microsoft Learn - how we define it, how we use it, etc.
Work with Microsoft Learn	New	New article about working with Microsoft Learn and the high-level engagement lifecycle.
Work with Microsoft Learn: Phase 1 - Engage and Assess	New	New article that provides a high-level overview of the 'engage and assess' process when new teams initially express interest in onboarding to Microsoft Learn.
Work with Microsoft Learn: Phase 2 - Onboard	New	New article that provides a high-level overview of the Microsoft Learn onboarding process.
Work with Microsoft Learn: Phase 4 - Ongoing Support	New	New article that provides a high-level overview of the Microsoft Learn ongoing support process.
Add a Lab to a unit using a hosted VM lab experience	Updated	Added basic troubleshooting for common LODS lab issues. Article retitled for clarity.
Add a Lab to a unit using the Azure Sandbox	Updated	Article retitled for clarity.
Working with LODS to create a hosted VM Lab	Updated	Article retitled for clarity.
Creating your own hosted VM Labs with LODS' Lab on Demand	Updated	Article retitled for clarity.
Microsoft Learn Repos	Updated	Removed learn-wwl-pr repo mention, since it is now deprecated.
Microsoft Learn Support overview	Updated	Replaced the Learner support option to 'Report issue via Email' with the new 'Report issue via Report Feedback form'

ARTICLE	CHANGE	DESCRIPTION
Triage and support 'report an issue' feedback	Updated	The article is now renamed as Azure DevOps Customer Feedback workflow. Updated the triage workflow for customer reported feedback and issues.
Troubleshoot common issues	Updated	Updated the method to report a content-related issue in Microsoft Learn via the 'Report Feedback' form.

August 2019

ARTICLE	CHANGE	DESCRIPTION
How to structure exercise units	New	New article explaining how to structure Learn exercise units.
Reference guide for performing Learn pull request (PR) reviews	New	New article to give a quick reference of Learn-specific items checked during PR review process.
How to build Learn URL	New	A new article explaining how Learn production and review URLs are built.
Learn Guide Home page	Updated	Updated the Home page to include move Content requirements and role-specific info to the relevant articles and added a section to include quick info on different sections in the Learn Guide.
Implement redirection for retired Microsoft Learn content	Updated	Updated the article to add more details about when redirections are required and when not required. Also added info about how to add master redirection file to repo, if one doesn't already exist.
Microsoft Learn FAQ	Updated	Updated the FAQ page to add more questions and answers.
Troubleshooting known issues	Updated	Added the details on how to troubleshoot the daily limit of 4 Sandboxes in Learn modules to the online learner support page for troubleshooting known issues.

July 2019

ARTICLE	CHANGE	DESCRIPTION
Microsoft Learn repo structure	New	New article explaining the structure of a standard Learn repo.

ARTICLE	CHANGE	DESCRIPTION
Creating your own LODS VM Learn Labs	New	New article outlining process for content authors & their preferred vendors to create their own VM labs for Learn content.
Working with LODS to create a VM Learn Lab	New	New article gives an overview of the typical process content authors can expect when working with LODS to create VM lab experiences for their Learn content.
Add a Learn Lab to a unit using a VM Lab experience	Updated	Article updated for new DIY VM lab creation process
Add a Learn Lab to a unit using a VM lab experience	Updated	Added a note to the ' Who can create a VM Lab for use on Learn ' section announcing that the Learn Guide will soon provide instructions for DIY LODS VM lab creation
Scaffold a module using template	Updated	Added a demo video of the scaffolding process
Triage and support workflow	Updated	Added triage information for localizations issues for ASP.NET, Windows and PowerPlatform.
All content	Updated	Learn docs is now referred to as Learn Guide. Updated all the references to use the new term

June 2019

ARTICLE	CHANGE	DESCRIPTION
Learn walkthrough overview	New	New landing page for Learn walkthrough articles.
Microsoft Learn learner experience	New	New article, part of Learn walkthrough, describing the Learn experience from a learner's perspective.
Microsoft Learn content structure	New	New article, part of Learn walkthrough, to give an overview of Learn content workflow and process from an author's perspective.
Microsoft Learn content development process	New	New article, part of the Learn walkthrough, explaining a high-level overview of the content development process for Learn.

ARTICLE	CHANGE	DESCRIPTION
Getting started with GitHub	New	New article, part of the Learn walkthrough, giving a high-level overview of how to get started with GitHub for Learn.
New author onboarding session	New	How to request a meeting for a new-author onboarding session
Updated video contacts for adding video	Updated	Add video in Microsoft Learn content
TOC	Updated	Added new section for Microsoft Learn Walkthrough.

May 2019

ARTICLE	CHANGE	DESCRIPTION
Module proposal review	New	How to request a meeting for a review of your module proposal
Module design review	New	How to request a meeting for a review of your module design document
Quick reference guidelines	New	A new quick reference guide to all the detailed guidelines for the Learn modules.
Add a Learn Lab to a unit using the Azure Sandbox	Updated	<ul style="list-style-type: none"> - Added Need Help section to provide the contact email address for help and enquiries. - Added Sandbox supported Azure services
Create Microsoft Learn repo	Updated	Added link to the Xamarin repo to the recurring scheduled publishing requests list.
Content requirements for Microsoft Learn	Updated	Updated to link to the pull request quality criteria article
Microsoft Learn partner engagements	Updated	Updated the format and information.

April 2019

ARTICLE	CHANGE	DESCRIPTION
Add a Lab to an MS Learn unit	New	New Learn Labs landing page article. Provides information about adding a lab experience to an MS Learn unit

ARTICLE	CHANGE	DESCRIPTION
Add a Learn Lab to a unit using a VM Lab experience	New	New article provides information about adding a VM lab experience to an MS Learn unit
Add a Learn Lab to a Unit Using the Azure Sandbox	New	New article provides information about using the Azure Sandbox to add a lab experience to an MS Learn unit
VM Learn Lab Intake Form	New	Added intake form to request & scope the creation of a new VM lab from Learn on Demand Systems (LODS)
How to structure a standard module	New	New ID Guidance for organizing a standard Learn module.
How to write a module summary unit	New	New ID Guidance for module summary.
Upload video on Red Tiger	New	New article describing detailed process for uploading and publishing MS Learn video on Red Tiger.
Add video to Learn content	New	New article explaining the process to add videos to the MS Learn content.
Guidelines for adding video in Microsoft Learn	New	New article explaining the guidelines to follow for adding a video to the MS Learn content.
Add notifications to modules	New	New article that provides information about how to add status notifications to modules when labs are unavailable.
Implement redirection	New	New article describing the process to implement redirections when retiring MS Learn content.
Track Learn content development in Azure DevOps	New	New article with documentation about how to track Learn content and content development in Azure DevOps.
View Learn content portfolio and roadmap	New	New article providing information and Azure DevOps queries for Learn content portfolio and roadmap.
Add task validation to units	Updated	Updated and retitled article to "Add task validation to units with an Azure Sandbox lab experience." Article provides information about adding task validation to Microsoft Learn units.

ARTICLE	CHANGE	DESCRIPTION
Add embedded interactivity to a unit	Deleted	Removed from Learn Guide, relevant information has been captured in multiple new Learn Lab-specific articles.
Enable Azure sandbox for a unit	Deleted	Removed from Learn Guide, relevant information has been captured in new Add a Learn Lab to a Unit Using the Azure Sandbox article.
Review info - PR checklist	Deleted	Removed from Learn Guide, since it is now included in the Docs Contributor Guide (Quality criteria for pull request edit review)
Guidelines for adding video in Microsoft Learn	Updated	Updated TOC to correct the link to the article.
Create a new sample code repository	Updated	Added a section on how to search existing code repos in GitHub.
Microsoft Learn GitHub repositories	Updated	Removed hard-coded list of code repos and replaced with how to find all. Re-organized sections for easier navigation.
Add video in Microsoft Learn content	Updated	Split the article into four smaller articles. Overview Upload video on Red Tiger Add video to Learn content Guidelines for adding video in Microsoft Learn

March 2019

ARTICLE	CHANGE	DESCRIPTION
Partner Engagements	New	New article listing the details for all the current engagement for Microsoft Learn.
Scaffold a module using template	New	New article describing the procedure to scaffold a module using template.
Learn code sample repos	New	New article for creating code repos, split from Creating Microsoft Learn repo article.
Job task analysis and roles description	New	New article to reference links to existing JTA documents
Overview of uid	New	New article giving a detailed overview of uids

ARTICLE	CHANGE	DESCRIPTION
Create a unit	Updated	Updated the requirements for the description metadata.
Propose new content	Updated	Updated the link to module proposal.
Create a learning path	Updated	Removed the reference to brand metadata from sample yaml code.
Enable Azure sandbox for a unit	Updated	Updated details with the new tool for module onboarding.
Propose new content	Updated	Updated for new processes.
Authoring Guidance	Updated	Guidance for "How to write an <i>Introduction to module</i> " added.
Engineering platform enhancements	Updated	Added details on priorities, states, and tags used to track requests, as well as some misc. minor changes.
Create a unit Scaffold a module manually	Updated	Added links to the new Overview of uid article.
Create new Microsoft Learn repo	Updated	Updated the steps to create new learn repo in OPS.
Authoring Guidance	Updated	Reference Design Doc and Proposal for "Introduction to Azure Logic Apps" added.
Request Learn achievement images	Updated	New PM contact info.
Triage and support	Updated	New contact information for "Assigned to" and categories for "Feedback Feedback Type".
Module Design Doc template	Updated	Changed this to a full code block making it super easy to copy/paste.
TOC	Updated	<ul style="list-style-type: none"> * Added Tools section, with applicable links. * Added content roadmap queries. * Renamed 'Report on content' section to 'View data'.

February 2019

ARTICLE	CHANGE	DESCRIPTION
Documentation integration	New	Information about how we're thinking about Docs vs. Learn.

ARTICLE	CHANGE	DESCRIPTION
Intake and onboarding - Overview	New	Split up the larger onboarding article - information how to engage with the Learn team.
Intake and onboarding - Learn roadmap	New	Split up the larger onboarding article - information about the Learn onboarding, content, and engineering roadmap
Intake and onboarding - Governance	New	Split up the larger onboarding article - information how we're thinking about content governance.
Intake and onboarding - Roles and responsibilities	New	-
Onboarding Checklist	New	Checklist for PMs to use when onboarding a new group to Microsoft Learn.
Launch new groups	New	Checklist for PMs to use when launching a new group on the Microsoft Learn platform.
Star ratings pilot	New	Overview of how we're piloting the new 5-star rating system for modules.
PR Review checklist	New	First draft of PR Review checklist.
How to run an event with Microsoft Learn	New	Resources to assist when you're running an event that is using Microsoft Learn.
Module Design Doc template	Updated	Format has been extensively changed and simplified.
Microsoft Learn GitHub repositories	Updated	<ul style="list-style-type: none"> * Updated information about how we're organizing repos (by product, not organization). * Updated 'what's live' repo list. * Removed information about loc snapshot branches, since no longer applicable. * Added info about whether Acrolinx is enabled for each repo.
Tracking content development in Azure DevOps	Updated	Outlines the way we use Azure DevOps to track learning paths and modules through the content development process.
Engineering platform enhancements	Updated	Details the newly created process to capture the ideas for platform enhancements.

ARTICLE	CHANGE	DESCRIPTION
Content requirements for Microsoft Learn	Updated	Added a "top rules" section to highlight the most important items.
Support triage	Updated	Updated support triage contact list with new groups. Freshness check for new AzDo workflow.
Enable Azure sandbox for a unit	Updated	Added information to inform content developers to not instruct learners to open Cloud Shell via the Azure Portal - because it doesn't work with sandbox.
Create new Learn repos	Updated	Added additional information about creating and onboarding new Learn repos for launch processes.
Star Ratings Workflow (removed)	Updated	Added info on enabling ratings and dashboard status
Content requirements for Microsoft Learn	Updated	Added a "top rules" section to highlight the most important items.
TOC	Updated	Included links to applicable Docs Contributor Guide guidance.
-	Updated	Included Learn graphics in the Learn Guide content repo.
Multiple	Updated	* Updated DevRel localization contact in all places to new contact. * Removed locales (en-us) from URLs. * Removed branch references (?branch=master) from URLs.

January 2019

ARTICLE	CHANGE	DESCRIPTION
How to write "Choose..." modules	New	Guidance for writing Microsoft Learn modules that help the learner choose the best technology for their projects.
Add zone pivots	New	Instructions on how to leverage the zone pivot feature in Learn units, with information about known issues and link to Engineering spec.
Enable Azure sandbox	New	Instructions how to enable the Azure sandbox in units, and information about policies, configuration, and limitations.

ARTICLE	CHANGE	DESCRIPTION
How to write introductory summaries	New	Guidance for writing summaries at the beginning of Microsoft Learn modules and learning paths.
ID Guidance page	Updated	Added link to draft reference content.
Create a module/Scaffold a module	Updated	Removed 'scaffold a module' article and moved related information into the 'implementation' section of the create a module article.
Add a knowledge check to a unit	Updated	Updated implementation information to reflect new 'explanation' property schema (replaces incorrectExplanation and correctExplanation).
Request Learn achievement images	Updated	Updated with AzDO template links to request badge and trophy images.
Create a module, Create a learning path, Request achievement images	Updated	Updated with information achievement property schema (badges and trophies are created in the module and learning path index.yml files, instead of in achievement.yml).
Microsoft Learn GitHub repositories	Updated	Updated status about using LearnShared now instead of DocsRoot for landing pages and shared assets.
Create a new Microsoft Learn GitHub repository	Updated	Added instructions on how to make code sample repos to support module content.
Content requirements for Microsoft Learn	Updated	Updated MVP rules to link to ID Guidance pages instead of repeating rule on both the MVP page and the ID Guidance pages.
Design-stage instructional design review	Updated	Updated to link to ID Guidance pages to remove duplication.
Content-handoff instructional design review	Updated	Updated to link to ID Guidance pages to remove duplication.
Multiple	Updated	Updated ms.topic, ms.custom metadata values for all articles to be Learn Guide specific
Multiple	Updated	Combined old 'Microsoft Learn experience' article content into the Learn Guide home page
Multiple	Updated	Combined 'Policies for Azure Learn subscriptions' article into the new 'Enable Azure sandbox' article

December 2018

ARTICLE	CHANGE	DESCRIPTION
How to write introductions	New	Guidance for writing the introduction for modules and units.
How to write scenarios	New	Guidance for writing scenarios for modules and units in Microsoft Learn.
Report on MSFT Employee Learn data	New	Contains information about how to report on MSFT Employee consumption of Learn content.
Customer feedback queue triage process	New	Contains information about how the Learn customer feedback queue is managed and triaged.
Support troubleshooting tips (removed)	New	Tips for troubleshooting customer support issues.
Video streaming administration	New	This article provides information to Microsoft Learn administrators about setting up external contributors to upload and publish videos on Red Tiger.
How to write prerequisites	New	Guidance on how to write prerequisites for Microsoft Learn modules.
Request achievement artwork	Updated	Expanded information about 'request an achievement' process to include info about process, naming of artwork files, usage of files, etc. Renamed article in TOC.
Create a module, Create a learning path	Updated	Updated achievements section to remove information about requesting achievements (since that has its own article) and added information about including achievement YAML values in content files.
Microsoft Learn repos	Updated	Re-organized repo information into 'live' and 'pre-release' groups, updated repo list to include DocsRoot and new LearnShared repo, added information about which organizations each content team is in.
Support model	Updated	Removed outdated information about using GitHub issues.
Learn Guide Home	Updated	Added information about how all Microsoft Learn content rolls up to the same URL, and linked to source repository list.

ARTICLE	CHANGE	DESCRIPTION
Create new Learn repos	Updated	Added information about configuring GitHub settings, added go-live tasks, added links to continuous publishing requests, re-organized content for easier use, added note of checks that should occur before creating new repos.
Learn Repos	Updated	Updated scheduled releases.
TOC	Updated	Added links to the Learn PowerBI report and the Customer Feedback PowerBI report (Note: These two reports are in progress of being combined in the future.)
TOC	Updated	Added link to Azure icon images to 'Resources' section.
Learn Guide Home	Updated	Removed outdated information and resources.
TOC	Updated	Re-organized TOC for easier use.
Creating achievements	Updated	Streamlined guidance around requesting achievements.
Add a video unit	Updated	Clarifications about how the Learn team uses video.
Create a module	Updated	Metadata has been updated.
Create a learning path	Updated	Metadata has been updated.
Scaffold module files	Updated	YML guidance has been updated.
Instructional design guidance landing page	Updated	Links updated to include new Prerequisites article.
Multiple	Updated	Removed old terms 'APEX' (now 'Developer Relations') and 'TripleCrown' (now 'Microsoft Learn') from documentation.

November 2018

ARTICLE	CHANGE	DESCRIPTION
Creating Learn achievements	New	This article provides information about the process of creating achievement badges and trophies.

ARTICLE	CHANGE	DESCRIPTION
How to write a knowledge check	New	Guidelines for writing knowledge checks for Microsoft Learn.
How to write learning objectives	New	This article provides guidance for writing MSLearn learning objectives.
Instructional design homepage	New	This article is the landing page for all ID guidance in Learn Guide.
What's new in Learn Guide	New	This article.
Content requirements for Microsoft Learn	New	This article provides content structure requirements for Microsoft Learn content.
Policies for Learn Azure sandbox subscriptions	New	This article discusses the policies in use in Learn Sandbox.
Working on Learn content in GitHub	New	-
Creating new Microsoft Learn GitHub repositories	New	-
Retiring modules	New	-
Add embedded interactivity to a Learn module	Updated	New guidance around chromeless, a new example of a live Lab On Demand lab, and implementing Lab On Demand Labs in review.
How to write titles	Updated	Guidelines for authoring titles for Learning Paths, Modules, and Units in Microsoft Learn.
Scaffold module files	Updated	-
Microsoft Learn GitHub repositories	Updated	-

Microsoft Learn frequently asked questions (FAQs)

1/14/2022 • 19 minutes to read

High-level project information

Q: What is Microsoft Learn?

Microsoft Learn combines short step-by-step tutorials, browser-based interactive coding/scripting environments, and task-based achievements to provide technical audiences who want to learn Microsoft technologies a free interactive way of learning to help them advance their technical skills. Notable characteristics:

- **Learning that respects your time:** Step-by-step bite-sized tutorials and modules aligned to role-based certification make it easier to dedicate time for learning.
- **Learning by doing:** Interactive coding environments right in the browser provide a friction-free hands-on experience.
- **Easy to navigate:** Follow guided learning by product, skill level, or job role that is easy to navigate and help you solve your business challenges or prepare for Microsoft certification.
- **Get recognized with achievements:** Complete modules, test your knowledge and earn achievements to recognize your skills. Share achievements through social media.
- **Learn in ways best for you:** Select from free self-paced quick tutorials and hands-on learning in Microsoft Learn, and get connected to structured online courses (free from Pluralsight for Azure roles) and instructor led classes from Microsoft Learning Partners. Learning across modalities will align to role-based certification.
- **Deep integration into documentation quickstarts and tutorials:** Learn is an extension of the Microsoft Docs portal, with a unified user profile between Docs, Learn, and Q&A forums.

Q: What's the relationship between Microsoft Learn and Docs.microsoft.com?

Microsoft Learn is an extension of the existing Microsoft Docs platform. It provides free, role-based, micro, hands-on, scenario-based learning content – so that learners can easily gain new skills with our products in less than an hour. Once the student knows how to use our products, they can then shift to other resources as needed (docs, StackOverflow, etc.) to answer specific questions about what they are trying to build. Our goal is to make the user experience with Docs seamless in both directions so that the unified platform can help students get the information they need, right when they need it.

Q: What content is available on Microsoft Learn?

Microsoft Learn currently hosts a wide range of content from different business groups and product teams across Microsoft. The existing Learn content portfolio includes learning content from the following groups:

- .NET
- Azure
- Dynamics
- Power Platform
- Cloud Advocates
- Databricks
- Microsoft 365
- SQL Server
- Windows Developer

- Worldwide Learning

For more details, see the [Learn portfolio and roadmap](#) article.

Q: Will all of Docs content be converted to the Microsoft Learn format?

No, although some content will. Microsoft Learn enables new functionality that isn't applicable for all of the content types that are currently on Docs. For example, quickstarts, articles, API reference guides, etc. would continue to use their current functionality so that we can provide information to customers as quickly as possible. Microsoft Learn allows us to host highly curated, interactive training content on the Docs platform so that users don't have to go to two different places to find helpful content. Post-Ignite, we'll want to review all of the tutorials that already exist on Docs, since they are training and identify which ones make sense to move into the Microsoft Learn feature set.

Q: What languages will Microsoft Learn be localized in?

Our goal is that Learn content for each product is available for learners in the same list of languages (sometimes referred to as the Extent of Localization (EOL)) as the product itself.

For example, if a product is available in 12 different languages, the Learn content for that product area is also localized in those languages. Here is the EOL for the products with content on Learn:

- **.NET (17 languages):** Japanese, Korean, T.Chinese, S. Chinese, Russian, German, Spanish, French, Italian, Brazilian Portuguese, Hungarian, Swedish, Czech, Dutch, Iberian Portuguese, Polish, Turkish
- **Azure (17 languages):** Japanese, Korean, T.Chinese, S. Chinese, Russian, German, Spanish, French, Italian, Brazilian Portuguese, Hungarian, Swedish, Czech, Dutch, Iberian Portuguese, Polish, Turkish Dynamics (11 languages): Japanese, German, Spanish, French, Italian, Brazilian Portuguese, Swedish, Dutch, Danish, Finnish, Norwegian
- **Microsoft 365:** TBD
- **PowerApps (23 languages):** Japanese, Korean, T.Chinese, S. Chinese, Russian, German, Spanish, French, Italian, Brazilian Portuguese, Hungarian, Swedish, Czech, Dutch, Iberian Portuguese, Polish, Turkish, Danish, Greek, Slovak, Finnish, Thai, Norwegian
- **PowerBI (23 languages):** Japanese, Korean, T.Chinese, S. Chinese, Russian, German, Spanish, French, Italian, Brazilian Portuguese, Hungarian, Swedish, Czech, Dutch, Iberian Portuguese, Polish, Turkish, Danish, Greek, Slovak, Finnish, Thai, Norwegian
- **Flow (23 languages):** Japanese, Korean, T.Chinese, S. Chinese, Russian, German, Spanish, French, Italian, Brazilian Portuguese, Hungarian, Swedish, Czech, Dutch, Iberian Portuguese, Polish, Turkish, Danish, Greek, Slovak, Finnish, Thai, Norwegian
- **SQL Server (10 languages):** T.Chinese, S. Chinese, German, Spanish, French, Italian, Japanese, Korean, Portuguese, Russian

Sometimes, business exceptions are made to this list for a subset of the content, such as Worldwide Learning's Technical Seller Initiative (TSI) Azure content is only available in 4 languages. To see the most up-to-date list of languages for each body of content, you can look at the "localization_scopes" section in the docfx (configuration) files in the [content repos](#) (example: `learn-pr`).

Q: Is registration required for Microsoft Learn?

No. Learners can access Microsoft Learn and the learning content anonymously. However, if a user doesn't create an account or sign in, they will not be able to access some functionality.

Notable features that become available when you create a profile and sign-in are:

- The ability to accrue points and achievements
- Tracking progress on learning activities
- Usage of the free labs
- User Profile with badges and completion history

Q: Is Microsoft Learn the place we send people to learn about our products and technologies?

Microsoft is simplifying our customer experience by offering a single destination for technical learning via Microsoft Learn, although the system is not ready to host everything at this time – we're working our way towards that.

Customers who want to learn how and why to use our products will go to docs.microsoft.com/learn/ to find training for all our products.

Because of this, we will be retiring many products across Microsoft to reduce the footprint and increase customer satisfaction with our learning offerings. MVA users pursuing training on topics covered by Microsoft Learn will be directed to equivalent material on Microsoft Learn when it goes live in September; all MVA users will be directed to the new site for future training.

Q: Is this for all customer types (partners, customers, employees)?

Yes, the intention is for this to be a free, open training platform available to anyone who has an interest to learn our products.

Q: How does Microsoft Learn relate to Channel9?

Channel9 is predominantly a video hosting platform. It has video-based training, and lots of interesting content, but isn't intended to be a structured learning platform that guides a student from where they are to where they want to be. Microsoft Learn will incorporate video where it makes sense, but will also utilize text, images, interactive assessments, and other modalities to engage the learner and assess the knowledge transfer. It will favor shorter, task-based learning that allows students to quickly master a skill they need to know in a just-in-time fashion. In addition, Microsoft Learn will be a living portal in that we will constantly keep the content refreshed and in sync with the products.

Q: Will Microsoft Learn allow for external pull requests to training content like Docs does today?

Not at the moment Microsoft Learn uses the same platform infrastructure as the Microsoft Docs platform, which does allow this functionality in certain content areas. Because Microsoft Learn content is more complex than traditional Docs articles though, we do not support external editing at the moment, but will be investigating opening it up as a part of the product strategy.

Q: How do I contribute content to the platform?

The Microsoft Learn platform is currently gated to ensure that all content meets specific standards and is strategic for our content roadmap. If you are interested in creating content for the platform, fill out the [Learn intake form](#) and Learn PM will get in touch.

Onboarding

Q: When can I get onboarded to the platform?

Before you can get onboarded to Microsoft Learn, you need to go through our intake process, that involves you submitting a request to host your content through an online [intake form](#). This form gets you in a queue and you will be reached out to by a Learn PM. Final onboarding to the platform depends on multiple factors, such as our current workload, pre-planned roadmap, and urgency of the content need. Once you submit the intake form, the Learn PM should be able to help you with getting these sorted out.

Q: How long does it take to produce content for Microsoft Learn?

An average one hour Learn module takes a minimum of 9 weeks for end-to-end completion.

Q: How much does it cost to host our learning content?

We do not charge a fee to host content on the platform; however, enabling interactivity via labs, and video production costs, localization costs, etc... should be taken into account.

Q: What is the difference between content types on Microsoft Learn and Docs. Isn't a learning module the same thing as a walkthrough or a tutorial?

Learn modules use gamification and short-form, task-based exercises to teach users through experience how to do a task. Walkthroughs typically don't provide much conceptual guidance and simply provide instructions for a user to accomplish a task quickly. Tutorials are closer to Learn modules but don't provide users with interactivity and are written typically on only a few sets of commonly-used topics rather than across the entire library of docs for a product, and often reference other docs as prerequisites or for further contextualizing within a docset.

Q: How do I provide feedback on features or ask for new features?

For external customers, they have the following options: Email: The 'Report and issue' link at the bottom of each Learn unit allows you to submit your feedback or support question via email Star rating: The Start rating features allows you to rate a Learn module from 1 to 5 stars.

Microsoft employees can submit feature requests via the Learn Engineering team's [feature request form](#)

Q: How do I find out about new Learn feature announcements?

Subscribe to the lists below. These distribution lists will help us do exactly as their name implies, provide you with partner announcements- such as feature release notes, updates, notices, etc..

- Learn Partner Announcements [Subscribe/Unsubscribe](#) for updates related to the Microsoft Learn product
- Learn Authoring Partner Announcements [Subscribe/Unsubscribe](#) for updates related to the authoring on Microsoft Learn product

Q: What all services do you provide for partners?

Once you are approved to onboard to Learn, you will be introduced to various teams who will assist you with the following activities:

- Setup and configure Learn infrastructure, such as
 - Repo(s), RedTiger, LOD, AzDo, localization pipeline
 - Grant permissions to all applicable systems
 - Add to all distribution lists
 - Create new localization project
- Help you get ready by
 - Project managing onboarding tasks with support, labs, governance, instructional design, engineering, and localization
 - Partnering new LOC team with the DevRel loc and training
 - Managing content author and content manager training
- Request the following engineering and design work:
 - Badge artwork
 - New metadata values
 - New landing pages, navigation nodes

In addition to this, we also offer the following services as you need:

- [Video](#)

Q: Can I integrate this into my product?

Yes. Multiple groups have chosen to work their product teams to include a link to Learn content directly from the product. If you have any questions, ask in the [Microsoft Learn Teams group](#).

Q: How does this fit in with our LinkedIn Learning strategy?

LinkedIn Learning is the place at MSFT for long-form video content and more internal-only training, for things like compliance and business skilling. MS Learn is where all customer-focused, public training content is available. LinkedIn Learning is also a place to host paid training materials while Learn is forever free.

Q: Why did Microsoft partner with Pluralsight?

Microsoft partnered with Pluralsight to provide customers and partners with a complete training solution to prepare for Azure certification, giving customers the choice of using self-paced courses and the Pluralsight Role IQ to clearly measure proficiency, identify skills gaps and create a clear path to certification.

Q: What learning content is provided in partnership with Pluralsight?

We started our partnership with Pluralsight 2 years ago to provide free Azure courses. In FY19, we are expanding our partnership with Pluralsight to provide a clear path for Azure role-based certification by building the courses to cover all the skills required for certification. Free Azure courses will be available for 3 Azure role-based certifications: Azure Administrator, Azure Solution Architect and Azure Developer. Each Azure role includes the Pluralsight Role IQ which measures a person's proficiency in the collection of technology skills required to succeed in the specific Azure role. Each Azure role has a set of skills required for certification and the Pluralsight Skill IQ online assessment identifies the skills gaps and provides a path of courses required to build the Azure expertise. The collection of Skill IQs combines to provide a person's Azure role IQ. Each Azure Skill IQ and Role IQ are represented with a digital badge that can be shared on social media.

Q: Will the old Azure, self-paced hands-on Labs and workshops offered via the <https://www.microsoft.com/handsonlabs> website still be provided in FY20?

Yes, self-paced hands-on labs and workshops will continue to be provided in FY20 via the <https://www.microsoft.com/handsonlabs> site. But, in FY20, we are migrating the self-paced hands-on labs to Learn.

Q: Can we design our own badges?

No. The Learn design team creates all badge imagery for Learn content so that they are consistent across the platform. There is no charge for this service. Once you start creating content, we will help you get integrated into the badge request process.

Q: Who owns the content, including ensuring it is up to date and relevant?

Content teams themselves are required and held accountable for keeping the content up-to-date. Typically, the content author will be notified if there are any issues, but as a part of the onboarding process, a Microsoft FTE must be identified for the entire portfolio in case the author cannot be found.

Content Development

Q: Is long-form video supported?

No. Long-form videos are not supported at this time on Learn. Video must be used in addition to text and other interactive components. The ideal length of a video is 2 minutes but it can go up to 5 minutes, if required. It's always better to have videos that are short, concise and to the point.

Q: What is the minimum and maximum length for a module?

There is no minimum length for a module, but the maximum length is 60 minutes. A typical Learn module is around 45-60 minutes long.

Q: What is the minimum and maximum length for knowledge checks?

A typical knowledge check should have between two to five questions. There are no limits on the number of knowledge checks within a module.

Q: Do knowledge checks have to be standalone?

No, knowledge checks can be added at the end of a learning unit, or as a standalone unit. Whatever the choice, the knowledge check questions/answers are always added in the unit YAML file. Content teams typically choose the method that works best for their content.

Q: Is there a checklist of requirements for content?

The content requirements for Learn are categorized under P0, P1, and P2. Content must meet P0 requirements

before initial publication and P1 requirements within 30 days of initial publication. The detailed checklist of the content requirements is documented in the [Content Requirement](#) article.

Q: How do I get help with {X}?

Ask in the Microsoft [Learn Teams group](#) and the Learn contributor community will assist you.

Q: Is there support for optional modules in a learning path?

Not at this time. In order for a user to earn a trophy for a learning path, they must complete all modules in the learning path. However, the Learn system isn't forcing or messaging to learners that they have to complete any modules or learning path – the learners can take whatever they want.

Q: Can I have different knowledge checks or task validation with zone pivots?

No. Zone pivots does not currently support this functionality; it only displays different text/video content when pivoted. There is an open customer suggestion to the Engineering team to support this functionality.

Q: What hosted lab environments does MS Learn support?

We currently have support for:

- Azure sandbox to create Azure services and resources.
- Dynamics 365 sandbox to support Sales and CE labs.
- Learn on Demand VMs to interact with custom VM images from a browser.
- Learn on Demand Cloud Slice to perform Azure tasks unavailable in the Azure sandbox.
- Try.NET to execute C# code in the browser
- Jupyter Notebooks (coming FY21H2) to run Notebooks in a Learn module.

Q: What capabilities are available in the Azure sandbox?

The Azure sandbox gives learners a safe, hosted environment to use Azure resources to complete Learn Lab exercises. The Azure sandbox provides the following capabilities:

- Sandboxes can be embedded through multiple units in a module. However, only one sandbox environment can only be used for one module and no other hosted environment can be used in that module.
- Task validation ability can be embedded within the sandbox units, if required. See [Add task validation](#) article to learn more.
- Once activated by the learner, a Resource Group is automatically created in a subscription owned by Microsoft for which the learner is given contributor-level access.
- Learners can use Azure Portal, Cloud Shell, or any other tool to create resources inside of the resource group created by the Azure sandbox.
- Once the learner completes the module (or after the lab's time limit has elapsed), Learn automatically cleans up the resources and removes the learner's access.

See the [Add support for the Azure sandbox](#) article for more details.

Q: What is the time limit for Azure sandbox environment?

Once activated, the Azure sandbox environment can be configured to be used for 1-4 hours depending on the scenario required. Note that VM and container-based labs have a strict 1-hour time limit to reduce fraud.

Q: What capabilities are available with VM labs?

VM labs gives learners access to a secure, pre-configured virtual environment to complete Learn Lab exercises. VM labs are hosted by Learn's lab partner, Learn on Demand Systems (LODS). The following capabilities are available:

- VM lab can span across multiple units in a module, but a single lab cannot be extended to multiple modules.
- Upon activation, a new browser window is launched that displays a LODS VM.
- All exercise units with instructions and other lab resources are displayed side-by-side in an iframe.

- Exercise units display sequentially as configured in the module index.yml.
- Content authors have the freedom to create their own VM lab or work with LODS to get one created.
- Once the learner completes the module (or after the lab's time limit has elapsed), LODS automatically cleans up the resources and removes the learner's access.

See [Add a VM lab](#) article for more details.

****Q Can I include a knowledge check within the VM lab? Yes, but add knowledge checks as a separate unit and not as part of an exercise unit. This may not render properly.**

Q: What is the time limit for VM lab?

Once activated the VM lab is activated for 4 hours or until the learner finished the module, whichever happens first. The 4-hour time limit is default and can be configured for each lab as required.

Q: Is there a limit on the number of resources a learner can deploy in the VM lab?

TBD

Q: How are localized hosted labs being handled? Are we running different deployments for localized labs?

The lab exercises presented to the learner are simply the Learn module's unit pages that contain those exercises, rendered side-by-side with the lab environment. As a result, following Learn's localization standards for your module content means that the lab exercises are localized by default, and no specific action is required to localize a lab. However, the compute environments provided to complete the lab are **not** localized at this time.

Q: My module was added to a subfolder of a content repo and is now hosting to users from that folder name. How do I get my content correctly published to /learn/modules/{module-name} instead?**

Once a module is published to the wrong URL, resolving the URL correctly becomes significantly more difficult. Normally, adding the redirect only requires a rule added to the repo's docfx.json file. Since this content has been published already, though, you also need redirects to take users from the old URL to the new URL. But due to a limitation in the redirect system, you also have to rename your folder first.

- Rename the folder to something new. This is a result of a limitation of the redirect system; it won't allow redirects from content that still exists in that location. Because you need to add redirects from the old URLs to the new "modules" URL, we need to move the folder so it no longer exists at the prior location.
- Add a rule for the renamed folder to **build > content** section of the docfx.json file in your content repo, replacing `{new-folder-name}` in this example below with the name you used when you renamed the folder.

```
{
  "src": "{new-folder-name}",
  "files": [
    "**/*.yml",
    "**/*.md"
  ],
  "exclude": [
    "**/includes/**",
    "**/design/**",
    "**/resources/**"
  ],
  "dest": "modules"
},
```

- Add JSON redirect rules to the `.openpublishing.redirection.json` file for every unit within every module contained in your newly renamed folder. While this can be done manually, it is much easier with a script to generate the resulting JSON. For example, after changing `{your-old-folder-name}` to your prior folder name, you can navigate to your renamed folder and run this PowerShell script to generate the redirect rules in JSON that you can copy and paste into the end of your repos redirection file.

```

$oldFolderName = "{your-old-folder-name}"
ForEach ($moduleDirectory in Get-ChildItem -Directory) {
    $unitYamlFiles = Get-ChildItem $moduleDirectory -Filter *.yml
    $unitYamlFiles | % {
        $yamlName = $_.Name
        $mdName = $yamlName.Replace(".yml", ".md")
        $noExtensionName = $_.Name.Replace(".yml", "")
        $redirectLastSegment = "$noExtensionName"
        Write-Output "    {`n        ""source_path"": ""learn-
pr/($oldFolderName)/($moduleDirectory.Name)/$mdName""`, `n        ""redirect_url"": "
"https://docs.microsoft.com/learn/modules/($moduleDirectory.Name)/$redirectLastSegment""`, `n
""redirect_document_id"": true`n    },"
    }
}

```

Q: My content contains US dollar signs (\$) and is rendering in an unexpected way. How do I keep LaTeX from rendering my content differently?

If the repo you are working in has LaTeX support enabled by default in the docfx.json file (`build > globalMetadata > show_latex` set to `true`), it will sometimes try to render content between USD currency signs (`$`) as if it contains LaTeX typesetting syntax, often involving lots of HTML `` elements you might not want handled by the LaTeX system.

If you need LaTeX rendering on the affected page, but not on all `$` characters, you will have to escape the character with two backslashes: `$` becomes `\$`. If you don't need LaTeX processing on the affected page at all, you can opt out of it by adding the `metadata > show_latex` field to the unit's YAML file, setting it to false.

```

metadata:
...
show_latex: false

```

Alternatively, if you need LaTeX support on a unit inside a repo where it isn't enabled by default, you can opt in to LaTeX processing for your unit by adding the `metadata > show_latex` field to the unit's YAML file, setting it to true.

```

metadata:
...
show_latex: true

```

Caution

This `show_latex` metadata field is required on the unit YAML to ensure the rendering happens correctly every time on units using LaTeX. Your content may render properly on the first load without the `show_latex` value set to `true`, but it can break for our users. If you navigate away from and back to the same unit, though, the syntax will not render correctly.

Working on Learn content in GitHub

1/14/2022 • 4 minutes to read

There's a series of tasks/commands to use when working in GitHub to submit new content and edits to existing content. The standard workflow can be roughly broken down to the following steps:

1. Fork the repo
2. Follow the workflow for working against `master` or release branches
3. Do the work in Markdown
4. Submit the work to GitHub

Terminology

For common GitHub terms, like **fork**, **remote**, and **repository**, check out <https://help.github.com/articles/github-glossary>.

Fork the repo

Create a fork of the repo on GitHub. Follow the steps in the [this Docs contributor guide doc](#) to create a fork and clone the forked repo locally.

Daily workflow

The first time you work on a given repo

COMMAND	WHAT IT DOES
<code>git remote -v</code>	Confirm that the remote repo locations are set up properly. This should output <code>fetch</code> and <code>push</code> lines for <code>origin</code> and <code>upstream</code> . Origin should have <code>your-git-username</code> in the URL and upstream should have <code>MicrosoftDocs</code> .
<code>git fetch upstream</code> or <code>git fetch upstream <release-branch></code> .	The addition of <code>upstream</code> or <code><release branch name></code> is to tell git to track against your release branch instead of master.

For working against release branches (e.g., `release-ignite18`)

COMMAND	WHAT IT DOES
<code>cd folder/repo</code>	Changes directory to wherever your repo is cloned to on your local system.
<code>git checkout master</code>	Check out <code>master</code> branch.
<code>git pull upstream master</code>	Pulls the current copies of all files, including new files, from <code>master</code> into <code>current-branch</code>
<code>git checkout -B <your new local branch name, like "barlan0511"> upstream/<release-branch-name></code>	Check out a new branch and set its upstream against the non-master branch you're interested in working with.

NOTE

If you get the error `fatal: 'upstream/branch-name' is not a commit and a branch 'branch-name' cannot be created from it`, you need to do `git fetch upstream` before you `git checkout`. This pulls down any upstream branches onto your local system that you can then check out.

At this point, you're ready to write your [content in Markdown](#).

After writing your content, you need to complete a set of commands to push the content from your local branch to the remote branch on GitHub.

COMMAND	WHAT IT DOES
<code>cd folder/repo</code>	Changes directory to wherever your repo is cloned to on your local system.
<code>git pull upstream <release branch name></code>	Confirm that you have the newest content on your local copy of the branch by pulling any new changes that have come in since you started working.
<code>git add -A</code>	Add all content you've been working on to your pull request.
<code>git commit -m "commit message"</code>	Write the first comment for your pull request and for this individual commit
<code>git push origin <branch name></code>	Submit your content to GitHub
Open https://github.com/MicrosoftDocs/repo-name	Create your PR at https://github.com .
Go to Pull Requests or look for the yellow notification bar	The base branch will show as master; change it to the release branch on the "Open a pull request" page. This will confirm that your content lands in the upstream branch where you were tracking changes from.

For working against master (e.g., ongoing changes)

COMMAND	WHAT IT DOES
<code>git checkout master</code>	Sets you up in your local copy of <code>master</code>
<code>git pull upstream master</code>	Pulls the current copies of all files, including new files, from remote <code>master</code> into local <code>master</code>
<code>git checkout -B <name of new branch></code>	Check out a new branch and sets its upstream against <code>master</code>
<code>git add -A</code>	Add all content you've been working on to your pull request.
<code>git commit -m "commit message"</code>	Write the first comment for your pull request and for this individual commit
<code>git push origin <branch name></code>	Submit your content to GitHub
Open https://github.com/MicrosoftDocs/repo-name	Create your PR at https://github.com

COMMAND	WHAT IT DOES
Go to Pull Requests or look for the yellow notification bar	The base branch will show as master; change it to the release branch on the "Open a pull request" page. This will confirm that your content lands in the upstream branch where you were tracking changes from.

At this point, you're ready to write your [content in Markdown](#).

After writing your content, you need to complete a set of commands to push the content from your local branch to the remote branch on GitHub. Note that the steps below are the same as those used in the release branch process.

COMMAND	WHAT IT DOES
<code>cd folder/repo</code>	Changes directory to wherever your repo is cloned to on your local system.
<code>git pull upstream <release branch name></code>	Confirm that you have the newest content on your local copy of the branch by pulling any new changes that have come in since you started working.
<code>git add -A</code>	Add all content you've been working on to your pull request.
<code>git commit -m "commit message"</code>	Write the first comment for your pull request and for this individual commit
<code>git push origin <branch name></code>	Submit your content to GitHub
Open https://github.com/MicrosoftDocs/repo-name	Create your PR at https://github.com .
Go to Pull Requests or look for the yellow notification bar	The base branch will show as master; change it to the release branch on the "Open a pull request" page. This will confirm that your content lands in the upstream branch where you were tracking changes from.

Cleaning up your branches

After your PR is merged, you can clean up the remote branch and local branch. To do this run the following git commands:

COMMAND	WHAT IT DOES
<code>git checkout <another_branch_name></code>	Switch to a different branch. You can't delete a branch you're actively checked out in.
<code>git push --delete origin <branch_name></code>	Delete remote branch. This can also be done via the delete branch button that appears in your PR on GitHub.
<code>git branch -d <branch_name></code>	Delete local branch

GitHub resources

- [GitHub commands cheat sheet](#)

- GitHub overview: Docs contributor guide

Microsoft Learn GitHub repositories

1/14/2022 • 3 minutes to read

Microsoft Learn content is hosted in GitHub repos under the MicrosoftDocs GitHub organization and are managed by the Microsoft Learn team. We are organizing around products/product families where teams that are producing content for that specific product will deliver content into that repo. We're not splitting out by organizations that are delivering the content. We try to keep the content in as few repos as possible.

For example, no matter whether Azure Learn content is developed by the Cloud Developer Advocates, Worldwide Learning, Azure Learn content team, or other team - they will all work in the learn-pr (Azure Learn) repo.

Learn Repo Managers:

- Adrian Stevens
- Adam Patridge
- Mark Smith

Contact learn-repo-managers@microsoft.com for any questions about our GitHub processes or repos.

Get started with Git and MicrosoftDocs

For information about how to get started in GitHub or the MicrosoftDocs organization, see the [Set up your GitHub account](#) article.

Permissions

All GitHub users that are a part of the GitHub MicrosoftDocs organization ([instructions to setup](#)) will automatically have read-access to all repos within that organization (i.e. all of ours). You do not need to request any special permissions to clone or submit PRs to these repos.

Landing page and shared asset repos

REPOSITORY	PURPOSE	STATUS	CONTENT ORG	LEARN TEAM OWNER	LOC OWNER	ACROLINX ENABLED?
LearnShared	Private repo that hosts content that is shared across the Learn product and/or repos, such as the Learn home page, support page, product landing pages, zone pivot files, etc.	Live	Developer Relations	Mark Smith, Adrian Stevens, Adam Patridge, Barry Langer, Ashley Johnson	Alisha Acharya	No

REPOSITORY	PURPOSE	STATUS	CONTENT ORG	LEARN TEAM OWNER	LOC OWNER	ACROLINX ENABLED?
learn-certs-pr	Private repo that hosts certification content. This content is primarily sourced from the WWL team.	Live	WWL	Michael Sullivan, Ashley Johnson	Bernadette Kelly	No

Module and learning path content repos

REPOSITORY	PURPOSE	STATUS	CONTENT ORG	CONTENT OWNER	LOC OWNER	ACROLINX ENABLED?
learn-pr	Private repo that hosts Azure Microsoft Learn content. This content is primarily sourced from the Microsoft Learn content development team.	Live	Developer Relations	Adrian Stevens	Nobuko Kurashige	Yes
learn-bizapps-pr	Private repo that hosts Business Applications Microsoft Learn content. This content is primarily sourced from the BAG team of content developers.	Live	Split - Developer Relations and Dynamics 365	Margo Crandall	Vince Nolan, Luciana Martins Franco, Nobuko Kurashige	Yes
learn-dynamics-pr	Private repo that hosts Dynamics 365 content.	Live	Dynamics 365	Margo Crandall	Vince Nolan, Luciana Martins Franco	Yes

Repository	Purpose	Status	Content Org	Content Owner	Loc Owner	Acrolinx Enabled?
learn-internal-pr	Private repo that hosts internal-only content from a preview build of the default branch ([example]/learn/paths/contribute-to-learn/?branch=main).	Internal live	Content & Learning (C&L)	Niveditha Narva	N/A	Yes
learn-m365-pr	Private repo that hosts Microsoft 365 content. This content is primarily sourced from various teams that support products in Microsoft 365, including some contributions from Developer Relations.	Live	Microsoft 365	Liza Poggemeyer	Tina McNaboe, Hideatsu Hosokai	Yes
learn-sandbox-pr	Private repo that hosts training content and is also used for testing internal tooling (e.g., WWL publishing web hooks). You should not consider adding new content to this repo. It is listed here only for reference.	Live	Contributor Success	Barry Langer		Yes
learn-windows-pr [deprecated]	Private repo that hosts Windows Developer content.	Deprecated/Deleted	Developer Relations			Yes

REPOSITORY	PURPOSE	STATUS	CONTENT ORG	CONTENT OWNER	LOC OWNER	ACROLINX ENABLED?
learn-xamarin-pr [deprecated]	Private repo that hosts Xamarin content. This content is primarily sourced from the XamU team.	Deprecated/ Deleted	Xam U			Yes

Code sample repos

Every module that has code samples or downloadable files gets its own supporting code repo. To create a new repo for code supporting a module, please see the [Learn code sample repos](#) article.

Code repos for individual modules follow the naming convention of <https://github.com/MicrosoftDocs/mslearn-module-name>. To find them all, search in GitHub for all repos in the MicrosoftDocs GitHub organization that start with `mslearn-`.

Content publishing schedule

Once content is merged into the **main** branch, it needs to be merged into the **live** branch for it to be published and live on the Learn site. To set this process up for new repos, see the [go-live tasks section of the Creating new Learn repos](#) document for details. Once enabled for a content repo, these merge activities happen automatically on the following schedule:

main -> live - Daily at 10:00 AM and 4:00 PM Pacific time

Add code samples to Microsoft Learn content

1/14/2022 • 6 minutes to read

This document outlines the guidance for creating and maintaining code samples for Microsoft Learn. The content below is applicable to all products and teams, regardless of the organization.

General guidelines

These principles set the baseline for quality standards of code that we ship for Learn. This list is less of a checklist, and more of a guiding set of recommendations that will help you build content that resonates with customers.

- Use the least amount of code possible to achieve your goal.
- Assume students will copy code into production apps.
 - Always follow industry good/best practices even if it means additional code.
 - Use proper [C# coding conventions](#) and refer to language style guides such as the [Python Style Guide](#).
 - Prefer to include appropriate error checking unless it significantly distracts from the core concepts.
- Always use appropriate method and variable names. As an example, for C#, the [Microsoft C# Naming Conventions](#) should be a good point of reference for what those should be.
- Avoid code comments (*if possible*) - the explanation should be in the unit itself.
- Use temporary variables if it helps convey meaning.
- Split a method to clarify usage intent - don't write one giant function to get things done.
- Always use real-world examples, especially when naming classes, methods, and variables.
- Use an ellipsis (...) in place of boiler-plate code or code not relevant to the current task/topic.

Organizing code samples for a module

Starter code, or code used in a module should always be placed into a GitHub repository in the [MicrosoftDocs](#) organization. Samples should be organized around the **solution** and use the naming convention `mslearn-{shortened-solution-name}`. For example, if the solution is "store-app-data-with-azure-blob-storage", a repository to hold the code might be `mslearn-store-data-in-azure`.

Creating new code repos

You can create a new MicrosoftDocs GitHub repository using the [Create a new repository](#) feature of the [internal open source tooling](#).

1. Select "GitHub for Open Source".
2. Give the repo a name using the naming standards (`mslearn-{solution-name}`).
3. Indicate it's a sample for a Microsoft Learn module in the description.
4. While it will need to be a **Public** repo before the module goes live, but it can start as **Private**.
5. For **Assign a Service or Opt-Out**, select the **Search services** field to open up the options.
6. At the top, select **Opt-out:No appropriate service**. This will add new fields for maintainers below this section.
7. For classifying the repository, select **Production**.
8. In the **Maintainers** section, make sure your name is added if you need to be able to manage this repo.
9. For the **Maintainers > Security group** field, set it to `learn-code-repos` and select the **MSFT Learn - Code Repo Maintainers** group option.

10. In the **Open source release approval** section, you'll make several selections.
 - a. New project
 - b. Yes, public open source
 - c. Sample code
 - d. MIT license
 - e. Yes, all created by my team
 - f. No to the project sending any telemetry
 - g. No to any cryptography (unless your sample actually does implement cryptography)
 - h. < 5,000 lines of code

11. For **Administrator Team permissions**, check the box next to **MSFT Learn Repo Managers**.

12. Skip the **Write permissions** section.

13. Confirm the **Repository template** is set to an MIT license with copyright set to Microsoft.

14. Pick an appropriate **.gitignore template** for the most common code that will be in your code repository (e.g., **VisualStudio** for a .NET project).

15. When you have confirmed everything, select **Create repository**.

Once the repo is created, each maintainer who needs **write** access to the repo should use just-in-time access to give their GitHub account **administrator** access. To do this, navigate to the **Direct owner access** tab and follow the instructions on the screen to elevate your access.

TIP

You should make it public initially unless the content is embargoed. Private repos can't be cloned without authentication which makes testing modules in the review site harder. You can always change the repo status after creation if you decide to change your mind later.

Guidelines for publishing code

CHECKLIST ITEM	DESCRIPTION
In public GitHub repo	<p>✓ DO Host your code in GitHub. Make the repo public. Use the MicrosoftDocs organization. Use the internal open source tooling to create a new repo. Name it with the prefix <code>mslearn-</code></p> <p>⊖ DO NOT Use a personal account/repository for the sample. Use VSTS/BitBucket/etc. for the sample. Make the repo private.</p>
Using simple branching	<p>✓ DO Use simple branching - <code>main</code> for content that is visible/public, any other branches for tests/experiments/validation. Place all sample content for a learning path in one branch.</p> <p>⊖ DO NOT Use elaborate branching for publication (e.g. one unit in one branch, another unit in a different branch).</p>

CHECKLIST ITEM	DESCRIPTION
One repo per solution	<p>✓ DO Create one repo per each solution. This could mean by module, multiple modules, or learning path. If the repo is for multiple modules, organize the code in your repo by module, where the folder name matches the module UID. Name your repo in the format of <code>mslearn-{module-id}</code>. Use your best judgment to determine a short name for the repo, with the <code>mslearn-</code> prefix. Use one repo for all languages in a module.</p> <p>⊖ DO NOT Create multiple repos for different languages within the same learning path. Use folder names that are inconsistent with the names of modules within the learning path.</p>
Include a <code>README.md</code>	<p>✓ DO Create a <code>README.md</code> file in the root of the repo. Outline the purpose of the samples. Include a link to the published module where the sample is used. Include a README for each full sample that is in the repo.</p> <p>⊖ DO NOT Leave the sample with no <code>README.md</code> Host documentation from within the sample repo. Have a very non-descriptive <code>README.md</code> file (e.g. "<i>This is a Learn Sample.</i>").</p>
Place all code in a <code>/src/</code> folder	<p>✓ DO Place all source in one parent folder.</p> <p>⊖ DO NOT Have the code scattered in the repository.</p>
Include an open-source license	<p>✓ DO Include a <code>LICENSE</code> file. Use MIT or Creative Commons licenses.</p> <p>⊖ DO NOT Skip a sample without a license. Use GPL or GPL-flavored licenses.</p>
Work with your community	<p>✓ DO Regularly check for PRs and issues. Respond to issues and PRs promptly. Deprecate the repository if sample is not used or no longer relevant.</p> <p>⊖ DO NOT Abandon the repository with no communication. Not respond to PRs and issues.</p>

CHECKLIST ITEM	DESCRIPTION
Prepare for indexing to the samples portal	<p><input checked="" type="checkbox"/> DO Create a manifest for the final, complete sample. Add a webhook to get your sample repo.</p> <p><input type="checkbox"/> DO NOT Onboard every unit sample to the samples portal. Ship a final, complete sample without a manifest file.</p>
Ensure Code is Validated	<p><input type="checkbox"/> DO NOT More details on this requirement will be available once the Constructors infrastructure is launched in the next months.</p>

Picking the organization

All samples need to be placed in the [MicrosoftDocs](#) organization.

Including code in Docs

Refer to our official guidance on [how to include code in content](#). We prefer that you include code from code files directly, via the `[!INCLUDE]` syntax instead of using code fences/inline code. That ensures that we are always able to build the content from the latest and greatest code, and the author does not have to re-visit every single article that has the snippet to add changes. Additionally, this also allows us to test the code and ensure that it works before being shipped to customers.

When it comes to including code within your documentation set, you should rely on [Cross-Repo References](#).

Additional questions

Why can't I have samples in my own repo?

To make sure that learning paths and any associated content can be maintained in the long run, we need to ensure that we retain ownership of the code at all times, hence the requirement to place the code into one organization.

Other questions

When in doubt, please reach out to the [Learn General Channel](#), we will be able to point you in the right direction.

Troubleshooting build issues

1/14/2022 • 5 minutes to read

The list below is to help you troubleshoot your build errors. Not all error messages may be worded exactly the way they are written below and not all error messages have a single solution that goes along with them, so there are tips below to hopefully help get you in the right direction. We are actively working with Engineering to help make these error messages friendlier and give a bit more guidance.

ERROR TITLE	ERROR STRING	SUGGESTED RESOLUTION
While scanning a simple key, could not find expected ':'.	[Error] Unable to load file: module-name/index.yml via processor: Module: azure/build-a-serverless-web-app/index.yml is not in supported format: (Line: 12, Col: 77, Idx: 708) - (Line: 12, Col: 77, Idx: 708): While scanning a simple key, could not find expected ':'.	Confirm that you don't have any additional whitespace on the location of the line.
While parsing a flow sequence, did not find expected ';' or ']'.	[Error] module-name/index.yml is not in supported format: (Line: 6, Col: 8, Idx: 359) - (Line: 6, Col: 9, Idx: 360): While parsing a flow sequence, did not find expected ';' or ']'.	May not support [includes] or missing expected final characters.
LearningPath can only have Modules as children.	[Error] Invalid children: (learn.uid). LearningPath can only have Modules as children.	UID must be a module UID. Will not work with achievement or unit UID.
Required properties are missing from object: iconName. Path 'achievements[4]'.	learn-pr/achievements.yml [Error] Unable to load file: achievements.yml via processor: Achievements: Validation against "https://dotnet.github.io/docfx/schemas/v1.0/schema.json#" failed: Required properties are missing from object: iconName. Path 'achievements[4]'.	Need to include iconName with each achievement in the achievement.yml file.
Children Uid(s) (learn.manage-resouces-with-azure-portal,learn.principles-cloud-computing,learn.explore-azure-services-features) can't be found.	paths/cloud-fundamentals/index.yml [Error] Children Uid(s) (learn.manage-resouces-with-azure-portal,learn.principles-cloud-computing,learn.explore-azure-services-features) can't be found.	The UIDs being referenced in the module's index.yml don't exist as a UID for a unit. Verify that the unit YML files are using the exact same UIDs as what's in the module index.yml. Usually, it's been worded slightly differently, like <i>learn-pr:sample-module.1-first-unit</i> , when it needs to be <i>learn.sample-module.1-first-unit</i> . If your UIDs match, verify that your unit files have the correct file extension. For example, the build system will only look for the UID in <code>*.yaml</code> files. It's very easy to accidentally name those files <code>*.yam1</code> instead, which will not be found.

Error Title	Error String	Suggested Resolution
Unit must belong to a Module.	azure/monitor-azure-cosmos-db/3-how-indexing-works.yml [Error] Unit (learn.monitor-azure-cosmos-db.3-how-indexing-works) must belong to a Module.	This is usually paired with the issue above. The unit UID does not appear in the module's index. Verify that the unit YML files are using the exact same UIDs as what's in the module index.yml. Usually, it's been worded slightly differently, like <i>learn-pr:sample-module.1-first-unit</i> , when it needs to be <i>learn.sample-module.1-first-unit</i> .
Unable to find file with uid "learn.create-cosmos-db-for-scale.2-run-performance-testing".	learn-pr/azure/monitor-azure-cosmos-db/index.yml [Warning] Unable to find file with uid "learn.create-cosmos-db-for-scale.2-run-performance-testing".	This is usually paired with the issue above. The unit UID does not appear in the module's index. Verify that the unit YML files are using the exact same UIDs as what's in the module index.yml. Usually, it's been worded slightly differently, like <i>learn-pr:sample-module.1-first-unit</i> , when it needs to be <i>learn.sample-module.1-first-unit</i> .
Property 'description' has not been defined and the schema does not allow additional properties. Path 'description'.	learn-pr/azure/azure-functions-and-io-bindings/knowledge-checks/2-unit-text-and-knowledge-check.yml [Error] Unable to load file: azure/azure-functions-and-io-bindings/knowledge-checks/2-unit-text-and-knowledge-check.yml via processor: Quiz: Validation against " https://dotnet.github.io/docfx/schemas/v1.0/schema.json# " failed: Property 'description' has not been defined and the schema does not allow additional properties. Path 'description'.	The description property is either not indented properly in the YML file, or may be misspelled. Check that it has the correct number of spaces in front of it.
Value "Azure" is not defined in enum. Path 'tasks[0].environment'.	learn-pr/azure/azure-functions-and-io-bindings/3-read-data-from-blob-storage-portal-lab.yml [Error] Unable to load file: azure/azure-functions-and-io-bindings/3-read-data-from-blob-storage-portal-lab.yml via processor: ModuleUnit: Validation against " https://dotnet.github.io/docfx/schemas/v1.0/schema.json# " failed: Value "Azure" is not defined in enum. Path 'tasks[0].environment'.	The value that was specified for environment was not a specified value .
Property 'type' has not been defined and the schema does not allow additional properties. Path 'tasks[0].type'.	learn-pr/azure/azure-functions-and-io-bindings/3-read-data-from-blob-storage-portal-lab.yml [Error] Unable to load file: azure/azure-functions-and-io-bindings/3-read-data-from-blob-storage-portal-lab.yml via processor: ModuleUnit: Validation against " https://dotnet.github.io/docfx/schemas/v1.0/schema.json# " failed: Property 'type' has not been defined and the schema does not allow additional properties. Path 'tasks[0].type'.	This can happen if the property is not indented properly in the YML file, or if it is misspelled. Check that it has the correct number of spaces in front of it.

ERROR TITLE	ERROR STRING	SUGGESTED RESOLUTION
Property 'name' has not been defined and the schema does not allow additional properties. Path 'tasks[0].name'.	learn-pr/azure/azure-functions-and-io-bindings/3-read-data-from-blob-storage-portal-lab.yml [Error] Unable to load file: azure/azure-functions-and-io-bindings/3-read-data-from-blob-storage-portal-lab.yml via processor: ModuleUnit: Validation against "https://dotnet.github.io/docfx/schemas/v1.0/schema.json#" failed: Property 'name' has not been defined and the schema does not allow additional properties. Path 'tasks[0].name'.	This can happen if the property is not indented properly in the YML file, or if it is misspelled. Check that it has the correct number of spaces in front of it.
Property 'conditions' has not been defined and the schema does not allow additional properties. Path 'tasks[0].conditions'.	learn-pr/azure/azure-functions-and-io-bindings/3-read-data-from-blob-storage-portal-lab.yml [Error] Unable to load file: azure/azure-functions-and-io-bindings/3-read-data-from-blob-storage-portal-lab.yml via processor: ModuleUnit: Validation against "https://dotnet.github.io/docfx/schemas/v1.0/schema.json#" failed: Property 'conditions' has not been defined and the schema does not allow additional properties. Path 'tasks[0].conditions'.	This can happen if the property is not indented properly in the YML file, or if it is misspelled. Check that it has the correct number of spaces in front of it.
Property 'level' / 'role' / 'product' has not been defined and the schema does not allow additional properties. Path 'levels'.	learn-bizapps-pr/customer-service/customer-service-cases/index.yml [Error] Unable to load file: customer-service/customer-service-cases/index.yml via processor: Module: Validation against "https://dotnet.github.io/docfx/schemas/v1.0/schema.json#" failed: Property 'level' has not been defined and the schema does not allow additional properties. Path 'levels'. Property 'role' has not been defined and the schema does not allow additional properties. Path 'roles'. Property 'product' has not been defined and the schema does not allow additional properties. Path 'products'. Required properties are missing from object: levels, roles, products. Path ''.	This occurs if the property is singular rather than plural. In this case, the properties should be "levels", "roles", and "products".
Invalid type. Expected Object but got String. Path 'achievement'.	learn-bizapps-pr/customer-service/customer-service-cases/index.yml [Error] Unable to load file: customer-service/customer-service-cases/index.yml via processor: Module: Validation against "https://dotnet.github.io/docfx/schemas/v1.0/schema.json#" failed: Invalid type. Expected Object but got String. Path 'achievement'.	This could be a result of accidentally including single ('') or double ("") quotes around the achievement UID.

ERROR TITLE	ERROR STRING	SUGGESTED RESOLUTION
Invalid type. Expected Boolean but got String. Path 'questions[2].choices[1].isCorrect'	learn-bizapps-pr/flow/flow-approvals/knowledge-checks/quiz-approvals.yml [Error] Unable to load file: flow/flow-approvals/knowledge-checks/quiz-approvals.yml via processor: Quiz: Validation against " https://dotnet.github.io/docfx/schemas/v1.0/schema.json# " failed: Invalid type. Expected Boolean but got String. Path 'questions[2].choices[1].isCorrect'.	This could be a result of accidentally including single ('') or double ("") quotes around the achievement UID.
Property 'Content' has not been defined and the schema does not allow additional properties.	[Error] Unable to load file: powerapps/powerapps-get-started/1-unit.yml via processor: ModuleUnit: Validation against " https://dotnet.github.io/docfx/schemas/v1.0/schema.json# " failed: Property 'Content' has not been defined and the schema does not allow additional properties. Path 'Content'. Required properties are missing from object: content. Path ".	This is likely the result of including content as a required property, which was necessary when there were units that only contained knowledge checks. You shouldn't see this error frequently.

Track Learn content development in Azure Boards

1/14/2022 • 15 minutes to read

We track Learn content development in the Azure Boards service of our [Microsoft Learn Azure DevOps](#) project. There are hundreds of people (and growing!) contributing to Learn as a whole, via content development and supporting roles. The Learn Azure Boards are our holistic "source of truth" to provide the information that the different teams need to be successful:

- **Content teams** can see if there's other content that you could use in your learning paths and avoid extra effort or duplication of content.
- **Supporting partner teams**, like Design (achievement creation), Localization (DevRel, Microsoft 365, Dynamics, and so on), Labs, Instructional Design, and others can see the scope of what you're supporting for forecasting budget and resources.
- **Anyone** requesting work from the Learn team, like achievement images, instructional design reviews, lab setup, will have a closed feedback loop with Learn. They can easily check the status of their requests.

The Learn team has several standard queries to use to look for published content, in-progress content, or planned content. Here are a few of the most popular queries:

- [Published modules](#)
- [Completed last month/sprint](#)

Content teams are responsible for creating and updating their module and learning path work items up until the point of publication.

IMPORTANT

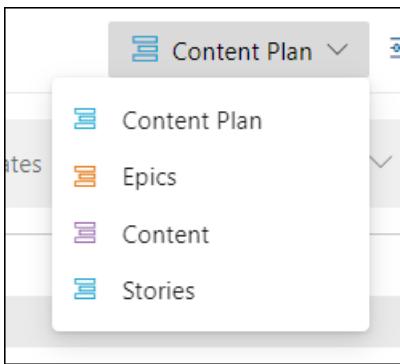
As soon as you start planning content and know roughly how many modules or learning paths you are going to create, start creating work items for them with as much information as you know. Update the work items as you finalize content.

If you can't access the [Learn Azure DevOps project](#), [request access to the CEAPEX DevOps board](#).

Planning content

Some teams prefer to create a comprehensive content plan that includes budgetary information before defining individual modules. We have a **Content Plan** work item type to collect the requirements and information into a single entry in the Azure Board. To create a content plan:

1. Go to the backlog appropriate for your content team:
 - Azure - [Azure Content](#)
 - Everyone else - [Partner Content](#)
2. In the upper right corner, select **Content Plan**. Don't select **Epics**, **Features**, or **Stories**.



3. Select **New Work Item** to create a new tracking work item.

The system should select **Content Plan** as the type automatically. If not, change it to **Content Plan**, then select **Add to top**.



4. Fill in the details in the presented form:

- The description should include overall information about the content you're creating.
- You're required to include the planned cost. It should be an estimate.
- You don't have to define modules at this level, but the expected module count is a required field.

5. Once you have all the details established, select **Save & Close**.

6. When you start defining modules for your plan, create relationships between the **Module** work items and the content plan. Use the **Related** link type. It's the most versatile approach.

Modules

Create **ModuleWorkItem** entries to represent *new* modules you plan to build. Create a single entry for each module and update the information in the work item as work progresses on the module. DevRel will automatically keep the Azure Boards representation synced with GitHub once you publish your content. Keep in mind that, while it's in-progress, you're responsible for all updates.

IMPORTANT

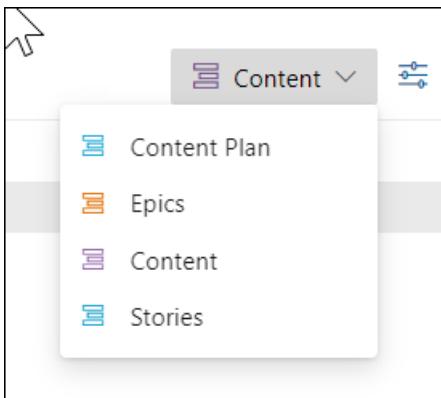
Do not create a new Module work item to represent an *update* to a published module. Instead, use a **Module Update** work item and associate it to the original work item with the **Related** link type. This lets us to maintain a history of prior work and versioning on modules in Azure Boards.

Create new module work items for new content

1. Go to the backlog appropriate for your content team:

- Azure - [Azure Content](#)
- Everyone else - [Partner Content](#)

2. In the upper right corner, select **Content**. Don't select **Epics**, **Content Plan**, or **Stories**.



3. Select **New Work Item**.
4. In the dialog box that appears, select **Module** from the drop-down list.

A screenshot of a 'New Work Item' dialog box. The 'Module' dropdown is open, showing several options: Feature, Learning Path, Module (selected and highlighted in grey), Module QA, and Module Update. The 'Module' option is currently selected. Other fields in the dialog include a search bar and an 'Add to top' button.

5. In the text box, enter the title of the module.

NOTE

It's ok if you don't know the exact title yet. Enter your best guess now and update it once you know more about it. For more information about good titles, see [How to write titles for learning paths, modules, and units](#).

6. Select **Add to Top**.
7. A new form will appear, prompting you to fill out the details for the module.

Description of the notable module work item fields

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Assignee	Optional - Highly recommended	Person creating the content, managing the content project, or point of contact for the content team.
Summary	Optional - Highly recommended	Summary of the module.
Design Notes	Optional	-
Learning Objectives	Optional - Highly recommended	Module learning goals. For more information about writing learning objectives, see How to write learning objectives .

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Prerequisites	Optional - Highly recommended, if known	Module prerequisites. For more information about writing prerequisites, see How to write prerequisites .
Justification	Optional - Highly recommended, if known	Justification for creating the module. Why do we need it on Learn?
Tech Reviewers	Optional - Highly recommended, if known	People slated for and capable of doing technical (content accuracy validation) reviews for this module.
Level	Required	Select a level of the module from the drop-down list: Beginner, Intermediate, Advanced.
Role	Required	Audiences that you're targeting this module for. Select from the drop-down list. You can choose more than one. If the audience that you need isn't listed, select Other and reach out to your content manager or Learn team point of contact for assistance. Role is a required field in the content itself and uses an allowlist. If we need to add an extra value, we'll need to work with Engineering to get it implemented.
Lab Enabled	Optional - Highly recommended	Toggle to True if you plan to implement an interactive lab with the module. (NOTE: <i>We have a limited set of interactive lab options at this time, so it only applies to Azure content.</i>)
Lab Type	Only required if Lab Enabled is True .	Select the type of lab you plan to implement or have implemented for the module from a pick list. If you don't know what type you need, select Unsure .
Vendor	Optional	If you're using a managed service vendor team, add their information here.
Content Team	Required	Select the name of your team. Content teams and supporting partner teams use the Content Team option to filter queries for their applicable content.
GitHub Repository	Optional - Highly recommended	Repository that the module will live in. If you're unsure - leave it blank or ask your content manager.

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Module UID	Optional - Highly recommended	UID for the module. The Module UID option is important. It helps us automate the updating of this work item for you. If you fill in the UID here and don't change it in the content, it's okay. We have tools to automate keeping the work item and the content synced. The title, summary, learning objectives, and so on, in this work item will remain synced with the content once it's published. You won't have to update it in two places.
Target Release Date	Required	Date that you're planning to release the module.
Published Date	Optional	Date that the module published to the live site. (NOTE: <i>We have tools to update it if you don't do it at the time of pushing your PR to the main branch.</i>)
Design Review Completed	Optional	Used to help identify if you've done a design review for the module. Using this option can help you query the status for this module across your content roadmap.
Technical Review Completed	Optional	Used to help identify if you've done a technical review for the module. Using this option can help you query the status for this module across your content roadmap.
Editorial Review Completed	Optional	Used to help identify if you've done an editorial review for the module. Using this option can help you query the status for this module across your content roadmap.
Related Work	Optional - Highly recommended if part of a learning path	Can link other work items to this one: achievement requests, video requests, learning paths, and so on. If the module is a part of one or more learning paths, it's highly recommended to link it to the learning path(s) so that we can see the higher-level plan. For information on how to link/relate work items, see the Link work items together section in this article, below.

8. Update the module state:

- **New** - Proposed item. If it hasn't been approved, you can also add a "Proposal" tag to the item.
- **Committed** - You're planning to do it, but haven't started the effort.

- **In Review** - In QA review phase.
- **In Progress** - In content development.
- **Blocked** - Content development is blocked.
- **Closed** - Module published to the live site.
- **Declined** - Decided not to move ahead with the proposed module.
- **Removed** - Module was published to the live site then deprecated.

9. Select **Save & Close**.

Update existing module work items

IMPORTANT

The instructions below are for tracking module creation progress while the content is being developed. If you are updating an *existing* published module, use the instructions in the next section.

1. Find the module work item that you want to update. There are multiple options in Azure Boards, but the most common are queries and backlogs:

Queries

- [Link to Learn Shared Queries](#)
- [Create your own query](#)

Backlogs

- Azure - [Azure Content](#)
- Everyone else - [Partner Content](#)

2. Select the module work item to open it.

3. Update the fields.

4. Select **Save and Close**.

Create a work item to update an existing published module

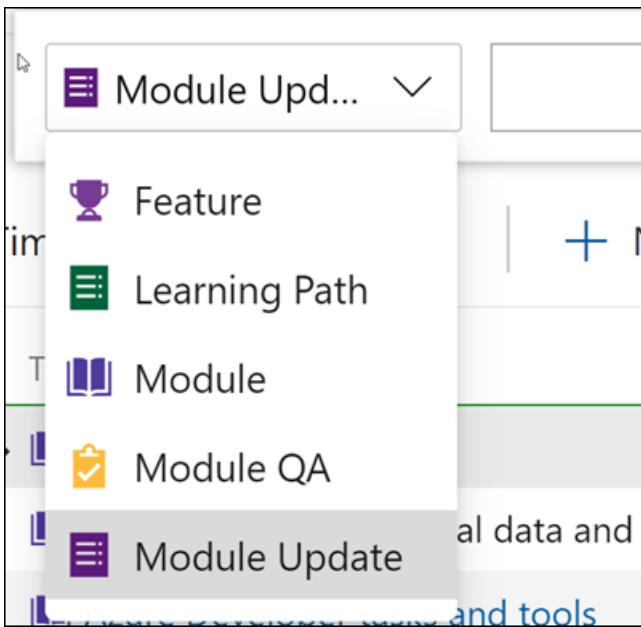
1. Go to the backlog appropriate for your content team:

- Azure - [Azure Content](#)
- Everyone else - [Partner Content](#)

2. In the upper right corner, select **Content**. Don't select **Epics**, **Content Plan**, or **Stories**.

3. Select **New Work Item**.

4. In the dialog box that appears, select **Module Update** from the drop-down list.



5. Fill in the Module GitHub repo and UID information from the existing, published module.
6. Add the details related to the change you're making. Most of these fields are the same or similar to creating a new module.
7. Select the **Content team** that owns the published module.
8. Add a link relationship to the existing **Module** work item that represents the published content. Use the **Related** link type.
9. Once you're done filling in the details, select **Save & Close**.

Learning Paths

Create new learning path work items

1. Navigate to the backlog appropriate for your content team:
 - Azure - [Azure Content](#)
 - Everyone else - [Partner Content](#)
2. In the upper right corner, select **Content**. Don't select **Epics**, **Content Plan**, or **Stories**.
3. Select **New Work Item**.
4. In the dialog box that appears, select **Learning Path** from the drop-down list.
5. In the text box, enter the title of the learning path.

NOTE

It's ok if you don't know the exact title yet. Enter your best guess now and update it once you know more about it. For more information about good titles, see [How to write titles for learning paths, modules, and units](#).

6. Select **Add to Top**.
7. A new form will appear, prompting you to fill out the details for the learning path.

Description of the learning path work item fields

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Assignee	Optional - Highly recommended	Person creating the content, managing the content project, or point of contact for the content team.
Summary	Optional - Highly recommended	Summary of the learning path.
Design Notes	Optional	-
Prerequisites	Optional - Highly recommended, if known	Learning path prerequisites. For more information about writing prerequisites, see How to write prerequisites .
Level	Required	Level of the module: Beginner, Intermediate, Advanced.
Role	Required	Audiences that you're targeting this module for. Select from the drop-down list. You can choose more than one. If the audience that you need isn't listed, select Other and reach out to your content manager or Learn team point of contact for assistance. Role is a required field in the content itself and uses an allowlist. If we need to add an extra value, we'll need to work with Engineering to get it implemented.
Content Team	Required	Select the name of your team. Content teams and supporting partner teams use the Content Team option to filter queries for their applicable content.
GitHub Repository	Optional - Highly recommended	Repository that the learning path will live in. If you're unsure - leave it blank or ask your content manager.
Module UID	Optional - Highly recommended	UID for the learning path. The Module UID option is important. It helps us automate the updating of this work item for you. If you fill in the UID here and don't change it in the content, it's okay. We have tools to automate keeping the work item and the content synced. The title, summary, learning objectives, and so on, in this work item will remain synced with the content once it's published. You won't have to update in two places.
Target Release Date	Required	Date that you're planning to release the learning path.

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Published Date	Optional	Date that the learning path published to the live site. (NOTE: <i>We have tools to update it if you don't do it when you push your PR to the main branch.</i>)
Related Work	Optional - Highly recommended	Can relate other work items to it (achievement requests, modules, and so on). For information on how to link related work items, see the Link work items together section in this article, below.

8. Select the drop-down arrow next to **Save and Close**.

NOTE

If you accidentally select **Save and Close**, it will save the work item in the new state. If you need to update the state, you can open up the work item again, update the state, and select **Save** or **Save and Close** again.

9. Update the learning path state:

- **New** - Proposed item.
- **Committed** - You're planning to do it, but haven't started the effort.
- **Published** - Learning path published to the live site.
- **In Progress** - In content development.
- **Blocked** - Content development is blocked.
- **In Review** - In QA review.
- **Rejected** - Decided not to move ahead with the proposed learning path.
- **Removed** - Learning path published to the live site then deprecated.

10. Select **Save**.

Update existing learning path work items

1. Find the learning path work item that you want to update. There are multiple options in Azure Boards, but the most common are queries and backlogs:

Queries

- [Link to Learn Shared Queries](#)
- [Create your own query](#)

Backlogs

- Azure - [Azure Content](#)
- Everyone else - [Partner Content](#)

2. Select the learning path work item to open it.

3. Update the fields.

4. Select **Save and Close**.

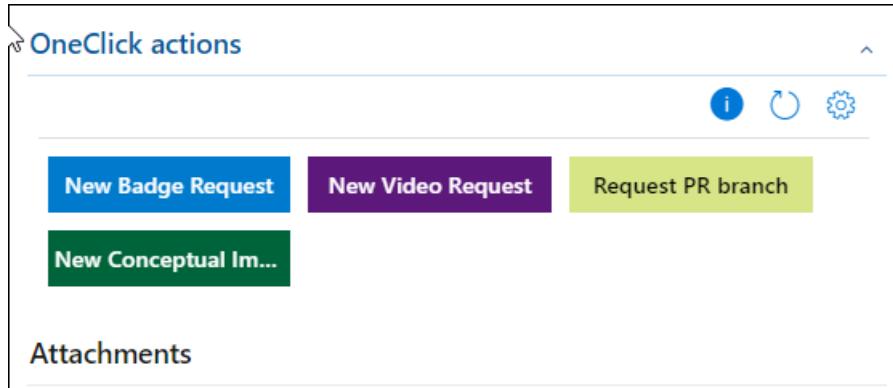
Other required work items

Create the following required work items at the same time you create a new module/learning path work item.

Badge or trophy

For a new module, you'll need to request a badge, and a new learning path needs a new trophy. Since the badge-design team is heavily backlogged and it may take some time to get your badge/trophy artwork completed, put in these requests as soon as you have the required details about the module.

Use the OneClick action within the module work item to open a badge request.



Release branch

Release branches are required for each new Learn module. Create the request as soon as you have finalized the module name and the repo location where the module should reside.

To request a new release branch, file a ticket with the [Content & Learning content production team](#). Provide the following details in the ticket:

- Check the boxes for **Microsoft Learn upstream module branch** and the sub-item for the desired Learn repo.
- Next to the Learn repo you checked, add the desired branch name prefixed with "NEW-" for new modules.
- Add any relevant aliases to the **FTE contacts** section.
- Set the **Requesting Group** to an appropriate value.

Conceptual artwork

For complex conceptual artwork, you can raise a request with the Content & Learning graphics artist (v-tstubbss@microsoft.com). Use the OneClick option within the module work item to open a request for a new conceptual image. Add the details of the required artwork and links to any existing images to help the artist. If you are creating this request on a contributor's behalf, tag them in the work item so they get notifications when the work item is updated.

Link work items together

There are a couple of ways to link work items together, depending on whether the work item exists already or not. Each is helpful for different scenarios.

If you created a one-off and forgot to link it, you may prefer the first option. Maybe you created a bunch that you need to sort out. You can use the second option. Are you the type of person who's organized ahead of time? You'd probably prefer option number three.

Link one work item to another

The first option for existing items is to link one work item to another.

1. Select the **work item** you want to relate another one to (could be Module or a Learning Path).
2. To the far right of the item, there's a section called **Related Work**. Select **Add link**.
3. Select **Existing Item**.

4. Follow prompts to link as a related item (**Link type = Related**).

NOTE

We don't link parent/child module and learning path work items since modules may appear in multiple learning paths.

5. Select **Save and Close**.

Update many work items to link to the same item

The second option for existing items is to update many work items to link them to the same item at the same time.

1. In the backlog view, **Ctrl+click** the items that you want to link. For example, select five module work items to link them to the same learning path work item.
2. **Right-click** the items you want to link.
3. Select **Add parent**.
4. Select **Existing Item**.
5. Follow prompts to link as a related item (**Link type = Related**).

NOTE

We don't link parent/child module and learning path work items since modules may appear in multiple learning paths.

6. Select **Save and Close**.

Creating new linked items associated with an existing work item

The third and final option is to create new linked items associated with an existing work item.

1. Select the work item you want link new items to.
2. To the far right of the item, there's a section called **Related Work**. Select **Add link**.
3. Select **New Item**.
4. Follow prompts to link as a related item (**Link type = Related**).

NOTE

We don't link parent/child module and learning path work items since modules may appear in multiple learning paths.

5. Select **Save and Close**.

Bulk creation tools

View the [Bulk add or modify work items with Excel](#) article on Docs. The process isn't simple, but we're constantly improving it. The process can save you significant time if you're creating more than 10-20 items.

Content queries

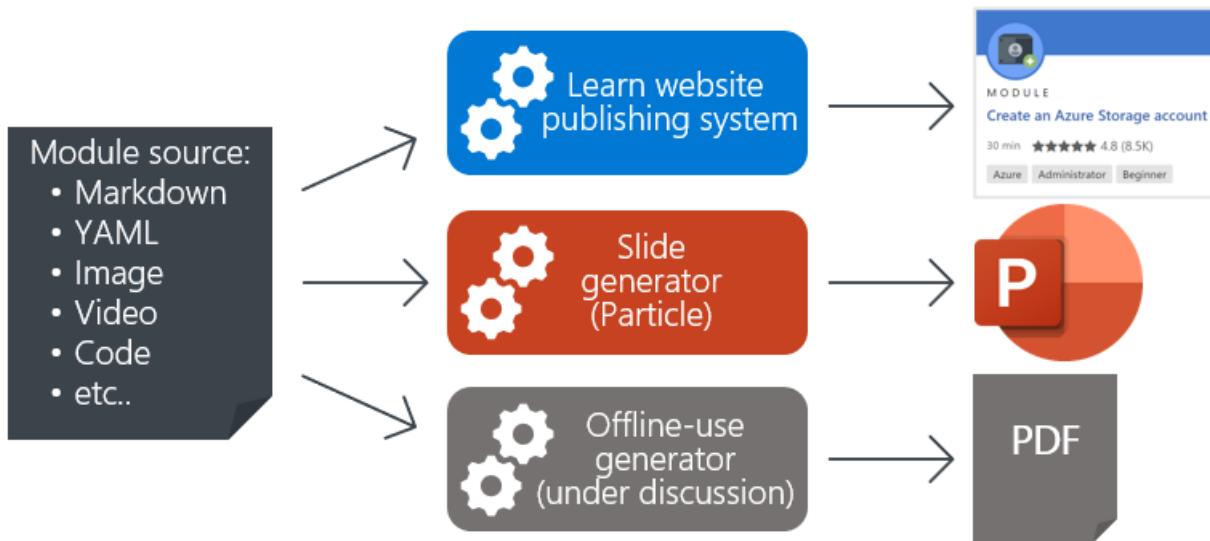
See the [View Learn content portfolio and roadmap](#) article.

Single-source content (Particle)

1/14/2022 • 2 minutes to read

Microsoft offers an unmatched portfolio of cutting-edge technologies that enable individuals and enterprises to build world-changing solutions. To be successful, our customers need relevant, up-to-date training. This training includes Microsoft Learn, instructor-led training, and online or in-person events.

We're working towards a single-source content model. We write our content using Markdown and YAML and automatically transform it into multiple output types. Authors write the content once and generate Learn website materials, instructor-led training slides, offline-study guides, and so on.



The single-source model reduces the need to create nearly duplicate content in alternate forms. It lets us create a repeatable and automatable process for asset creation across all our training modalities.

Slide publishing system

The following pages document the transformation process of Learn content into PowerPoint slides suitable for instructor-led training.

- [Module transformation rules](#)
- [Transformation tooling](#)
- [Authoring best-practice](#)
- [Template creation](#)
- [Previewer for VS Code \(Quark\)](#)

Single-source: module-to-PowerPoint transformation rules (Particle)

1/14/2022 • 5 minutes to read

Caution

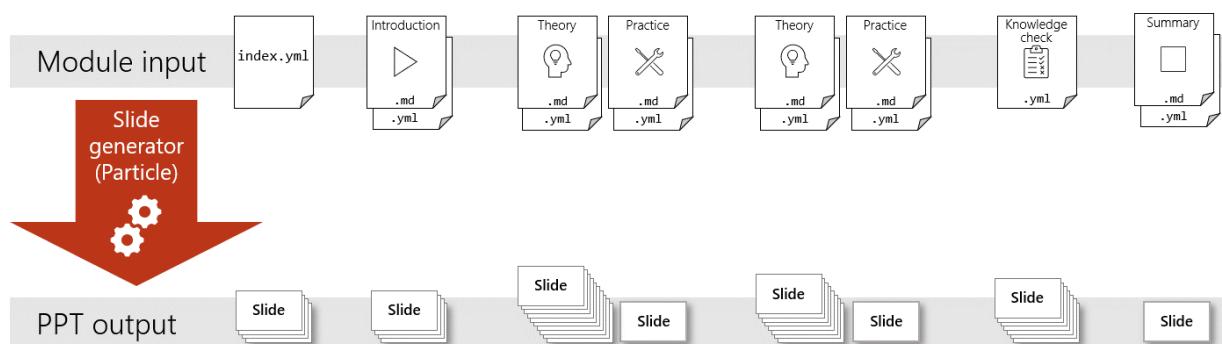
This content is subject to change. The Particle tooling is a pilot project and is under active development.

By the end of this article, you'll be able to:

- Identify how the elements of a Learn module map to PowerPoint slides

Overview

The *single-source tooling* (Project Particle) converts Learn content into PowerPoint slides. This article documents the mapping of one Learn module into PowerPoint. The following image provides an overview of the generated slides.



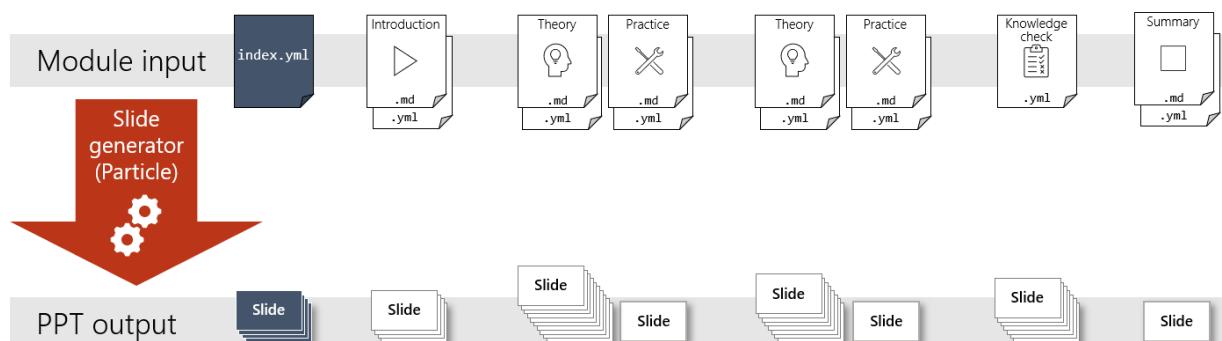
The source content itself completely determines the PowerPoint slides. There are no extra markdown or YAML elements you can add to influence the way the slides are created.

The generated slides are intended for use in instructor-led training or other live presentations. The focus on live presentation means the slides contain only a select portion of the source content. For example, they have the key points from the theoretical content but only a brief summary of the practice exercises.

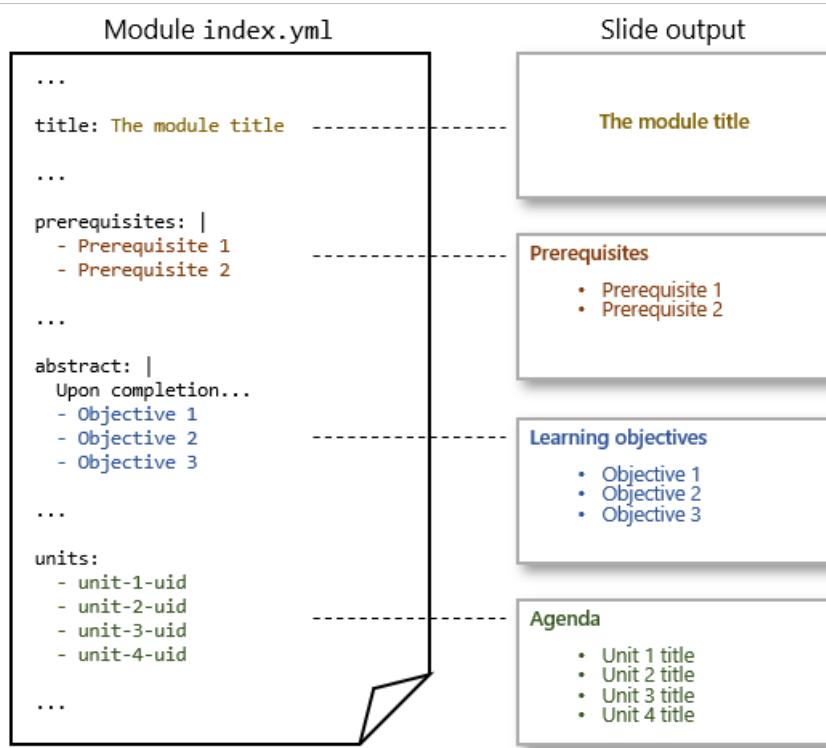
The following sections show the mapping for every supported source element.

Front matter

Slides for the module front matter are generated from select values in the `index.yml` metadata file.



The attributes `title`, `prerequisites`, `abstract` (learning objectives), and `units` each yield one slide. All other attributes in `index.yml` are ignored.

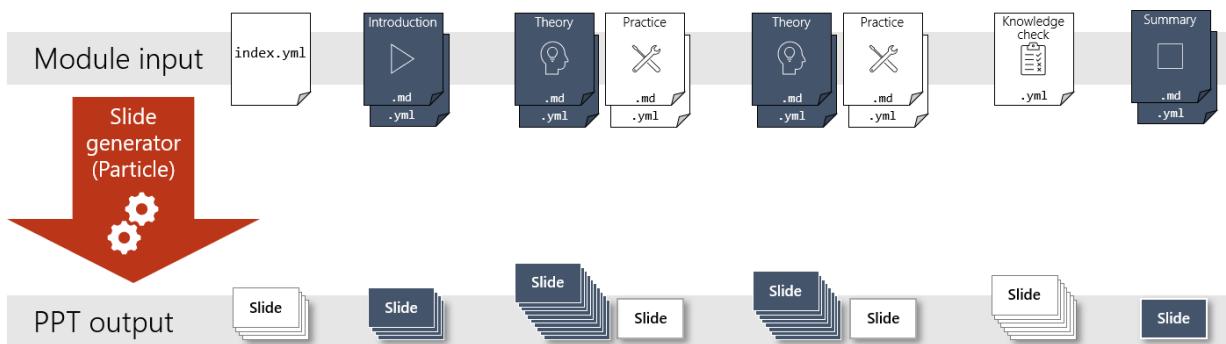


The following table contains the details of the mapping.

SOURCE ATTRIBUTE	TEMPLATE LAYOUT	TITLE PLACEHOLDER MAPPING	SUBTITLE PLACEHOLDER MAPPING
<code>title</code>	<code>title_layout</code>	Value of <code>title</code> attribute.	Not applicable.
<code>prerequisites</code>	<code>bulleted_layout</code>	The literal text "Prerequisites".	The values in the <code>prerequisites</code> attribute as a bulleted list.
<code>abstract</code>	<code>bulleted_layout</code>	The literal text "Learning objectives".	The values in the <code>abstract</code> attribute as a bulleted list.
<code>units</code>	<code>bulleted_layout</code>	The literal text "Agenda".	Unit titles pulled from the <code>title</code> attribute in the unit YML files.

Introduction, theory, and summary unit mapping

This section covers the mapping rules for three types of units: introduction, theoretical learning content, and summary.



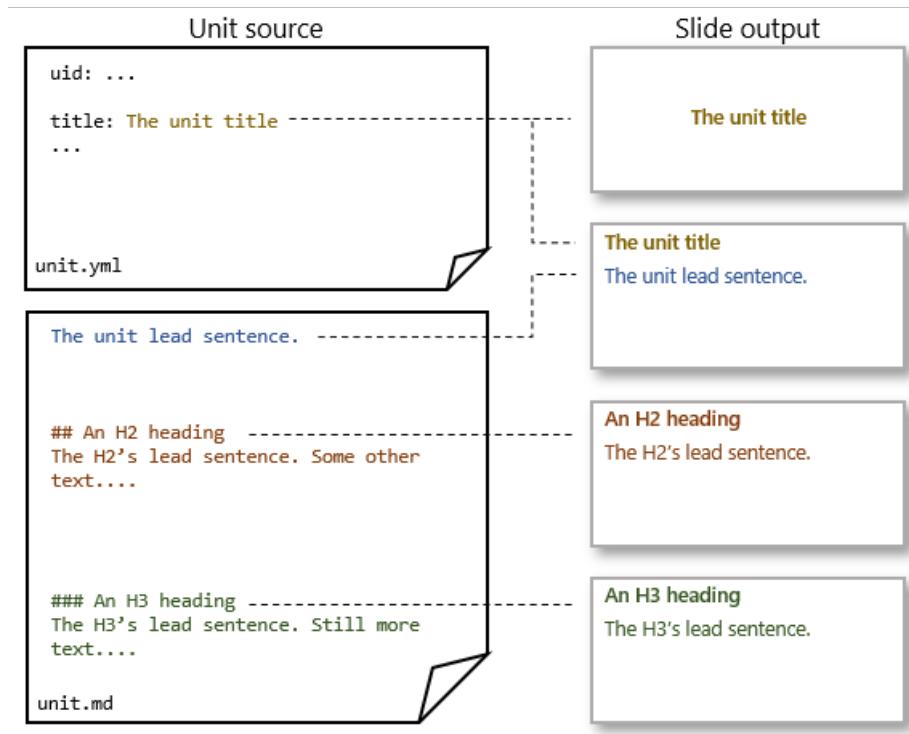
NOTE

There is one special case in the mapping rules for introduction units. If an introduction unit contains the headings "Learning objectives" or "Prerequisites", those areas are omitted from the slides.

Text-element mapping

Each unit begins with a section-title slide and an introductory slide. After that, each H2 and H3 in the markdown source generates one slide. Content in H4s or greater isn't mapped to slides.

Slide text is taken from the unit `title`, heading text, and lead sentences. The following image gives an overview of the text used in the generated slides. Note how the unit `title` appears on two slides, once in the section slide and once as the title of the introductory slide.



The following table contains the details of the mapping for textual elements.

SLIDE	TEMPLATE LAYOUT	TITLE PLACEHOLDER MAPPING	SUBTITLE PLACEHOLDER MAPPING
First	<code>section_layout</code>	Value of unit's <code>title</code> attribute.	Not applicable.

SLIDE	TEMPLATE LAYOUT	TITLE PLACEHOLDER MAPPING	SUBTITLE PLACEHOLDER MAPPING
Second	content_layout	Value of unit's title attribute.	First sentence of first paragraph.
All others	content_layout	Value of the H2 or H3.	First sentence of first paragraph.

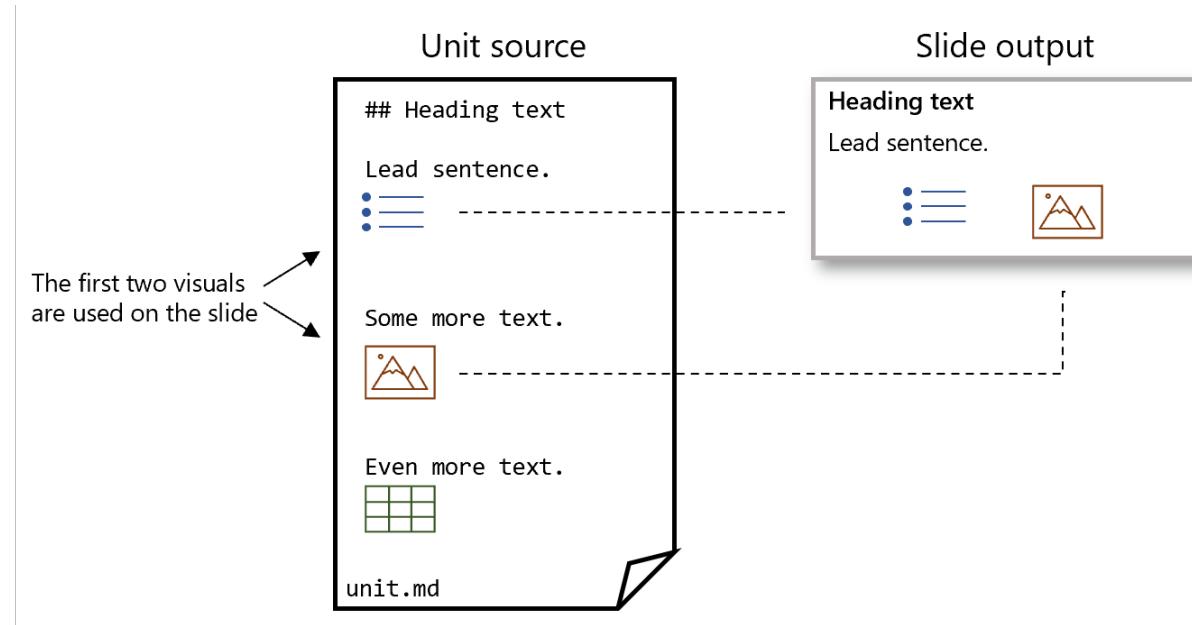
Also, all of the source text is copied to the Notes area below the slide. This text won't be visible during a slideshow.

Visual-element mapping

Each slide that uses the content_layout template will include up to two visual elements. The source items that are considered visual elements are:

- Image
- Bulleted list
- Numbered list
- Blockquote
- Code block
- Table

The visual elements are taken from the source in textual order. The visuals for a slide are selected only from the corresponding area of the source. In other words, a slide generated from an H2 will only use visuals that appear directly within that H2. The following diagram shows a heading with three associated visual elements. Notice how the first two appear on the slide.

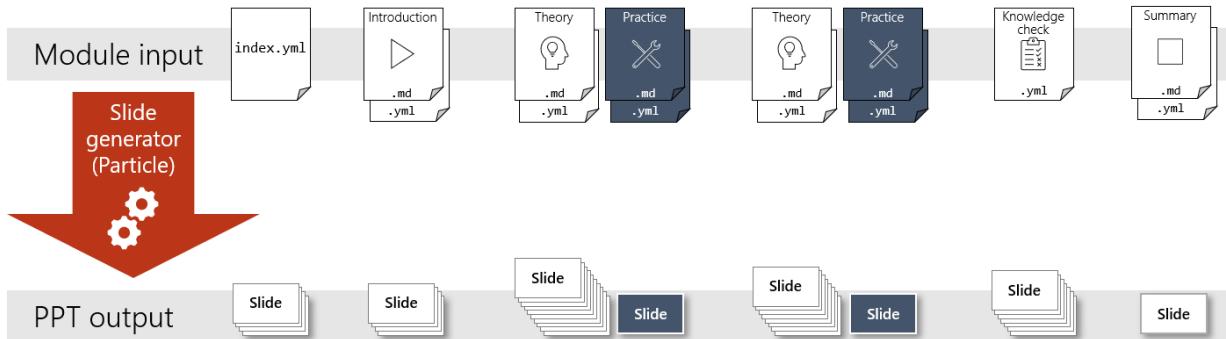


There are two cases for headings with fewer than two associated visual elements:

- If there is only one visual element, then that element will appear on the slide.
- If there are no visual elements, then the corresponding slide will contain only text.

Practice exercise unit mapping

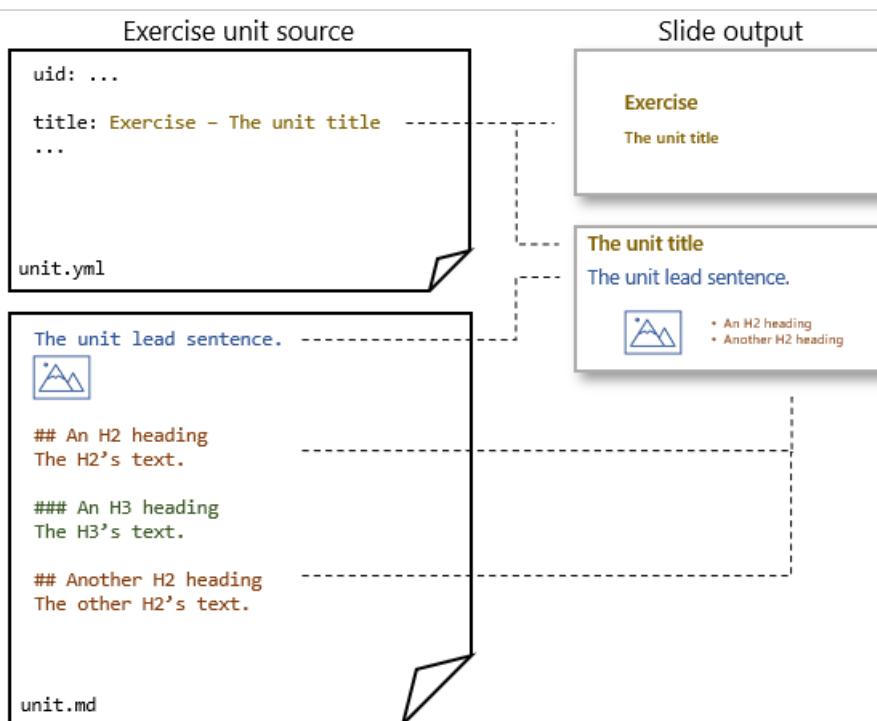
This section covers the mapping rules for practice exercise units. Exercise units are identified either by the "Exercise" prefix in the unit title or the use of the Learn sandbox.



Each exercise unit generates two slides:

1. **Unit-title slide**
 - a. Displays the word "Exercise"
 - b. Displays the unit title.
2. **Exercise-summary slide**
 - a. The slide title is the unit title.
 - b. The main bullet is the unit lead sentence.
 - c. If there's a visual element at the beginning of the markdown (positioned anywhere before the first H2), that visual is added to the slide.
 - d. A bulleted list of the unit H2s is added to the body of the slide.

The image below summarizes the two slides.



The following table contains the details of the mapping for textual elements of the two slides.

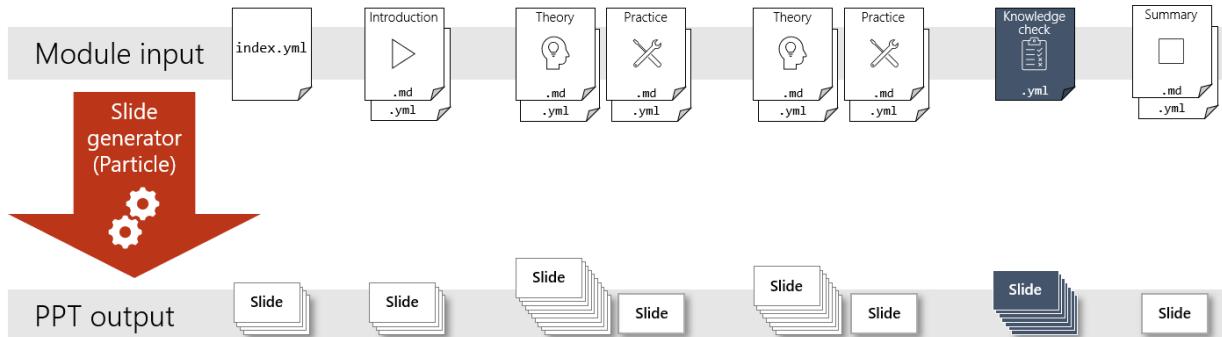
SLIDE	TEMPLATE LAYOUT	TITLE PLACEHOLDER MAPPING	SUBTITLE PLACEHOLDER MAPPING
Unit-title	section_layout	The literal text "Exercise".	Value of unit's title attribute without the "Exercise" prefix.

SLIDE	TEMPLATE LAYOUT	TITLE PLACEHOLDER MAPPING	SUBTITLE PLACEHOLDER MAPPING
Exercise-summary	content_layout	Value of unit's title attribute.	First sentence of first paragraph.

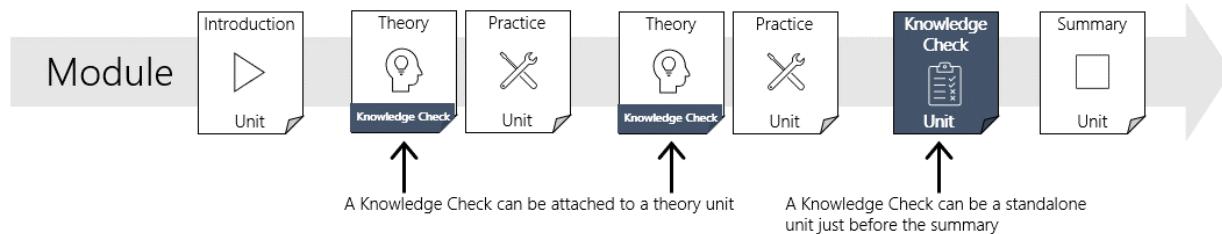
Also, a link to the published unit on the Learn website is added to the Notes area of the two slides. This text won't be visible during a slideshow.

Knowledge check mapping

This section covers the mapping rules for knowledge checks.



A knowledge check is typically written as a standalone unit positioned just before the summary. It's also possible to incorporate a knowledge check into a theory unit as shown in the following image. The mapping rules are the same for both cases.



Each knowledge check generates a title slide followed by two slides per question:

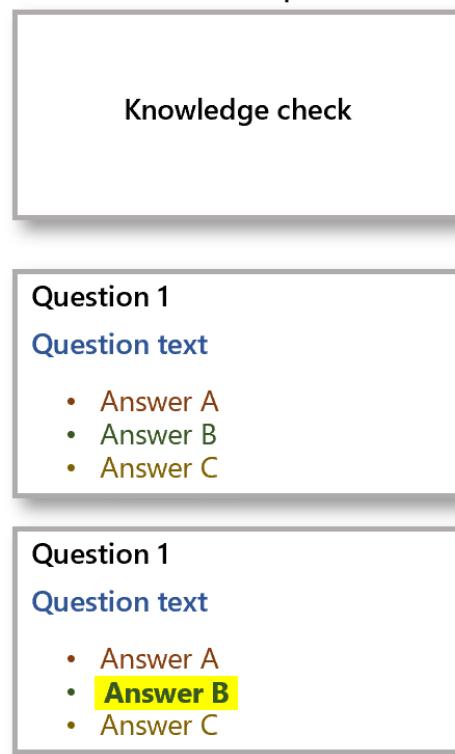
1. **Title slide**
 - a. Displays the phrase "Knowledge check"
2. **Question slide**
 - a. The slide title is "Question N" where N is the number of the question.
 - b. The main bullet is the question text.
 - c. The slide body contains a lettered list (A, B, C) of the answers.
 - d. The Notes view contains the explanation for the correct answer.
3. **Answer slide**
 - a. Same as the question slide except the correct answer is Bold and Highlighted.

The image below shows a mockup of the title slide and the two slides per question.

Unit yml

```
quiz:  
  
    questions:  
  
        - content: "Question text"  
          choices:  
  
            - content: "Answer A"  
              isCorrect: false  
              explanation: "Explanation A"  
  
            - content: "Answer B"  
              isCorrect: true -----  
              explanation: "Explanation B"  
  
            - content: "Answer C"  
              isCorrect: false  
              explanation: "Explanation C"  
  
    ...
```

Slide output



The following table contains the details of the mapping for textual elements of the three slide types.

SLIDE	TEMPLATE LAYOUT	TITLE PLACEHOLDER MAPPING	SUBTITLE PLACEHOLDER MAPPING
Title slide	section_layout	The literal text "Knowledge check".	Not applicable.
Question/Answer slides	content_layout	The text "Question N" where N is the question number.	Question text.

Non-supported source elements

The following source elements are partially or fully unsupported:

- **Video embedding:** each video link currently generates one slide with the video link as the slide content. The video itself isn't embedded in the slide. No thumbnail image is used for the link.
- **Zone pivots:** all slides for all pivots are currently generated and placed in one output slide deck.

Single-source: PowerPoint transformation usage (Particle)

1/14/2022 • 2 minutes to read

Caution

This content is subject to change. The Particle tooling is a pilot project and is under active development.

By the end of this article, you'll be able to:

- Select a PowerPoint template for use during slide generation
- Run the transformation tooling on your content

Overview

This document discusses how to generate PowerPoint slides from your Learn source content. The system is web-based and located at the Learn [content metrics portal](#).

You specify the GitHub address for your source content and the tooling will generate a PowerPoint file. You can preview the slides directly in the portal or download the file for offline use.

The tooling transforms one Learn module into one PowerPoint slide deck. Transformation of individual units, multiple modules, and learning paths isn't currently supported.

Source content location

The source content must be stored in GitHub. The content can be on any GitHub branch (that is, it doesn't need to be on the `main` branch nor does it have to be published).

You'll need to locate the URL for the root folder in GitHub that contains your content. The tooling will use that address to retrieve your YAML and Markdown source from GitHub and transform it into slides. For example, to transform the [Create serverless logic with Azure Functions](#) module, you would use the following GitHub address as input:

```
https://github.com/MicrosoftDocs/learn-pr/blob/master/learn-pr/azure/create-serverless-logic-with-azure-functions
```

PowerPoint template selection

The tooling injects a PowerPoint (`.pptx`) file into the transformation process. The file acts as a template that determines the fonts, colors, and layouts of the generated slides.

You can choose from a small selection of predefined templates. These typically include one using the Microsoft Learn branding and one for the next Microsoft conference such as Ignite or Build. Support for custom templates is on the feature backlog but not currently implemented.

Authentication

Valid Microsoft credentials are required to run the transformation tooling. You'll need to log in with your credentials or single-sign-on to access the portal.

How to transform Learn content to PowerPoint

Steps to generate slides:

1. Navigate your web browser to the [Learn Metrics Portal](#).
2. Log in using your Microsoft credentials or single-sign-on.
3. Enter the module's GitHub source address into the **GitHub URL** text box.
4. Select a PowerPoint template.
5. Click the **Search** button
6. Click on the **PowerPoint** tab.
7. Click the **Download** button to save the file.

Single-source: PowerPoint authoring best-practices (Particle)

1/14/2022 • 2 minutes to read

Caution

This content is subject to change. The Particle tooling is a pilot project and is under active development.

By the end of this article, you'll be able to:

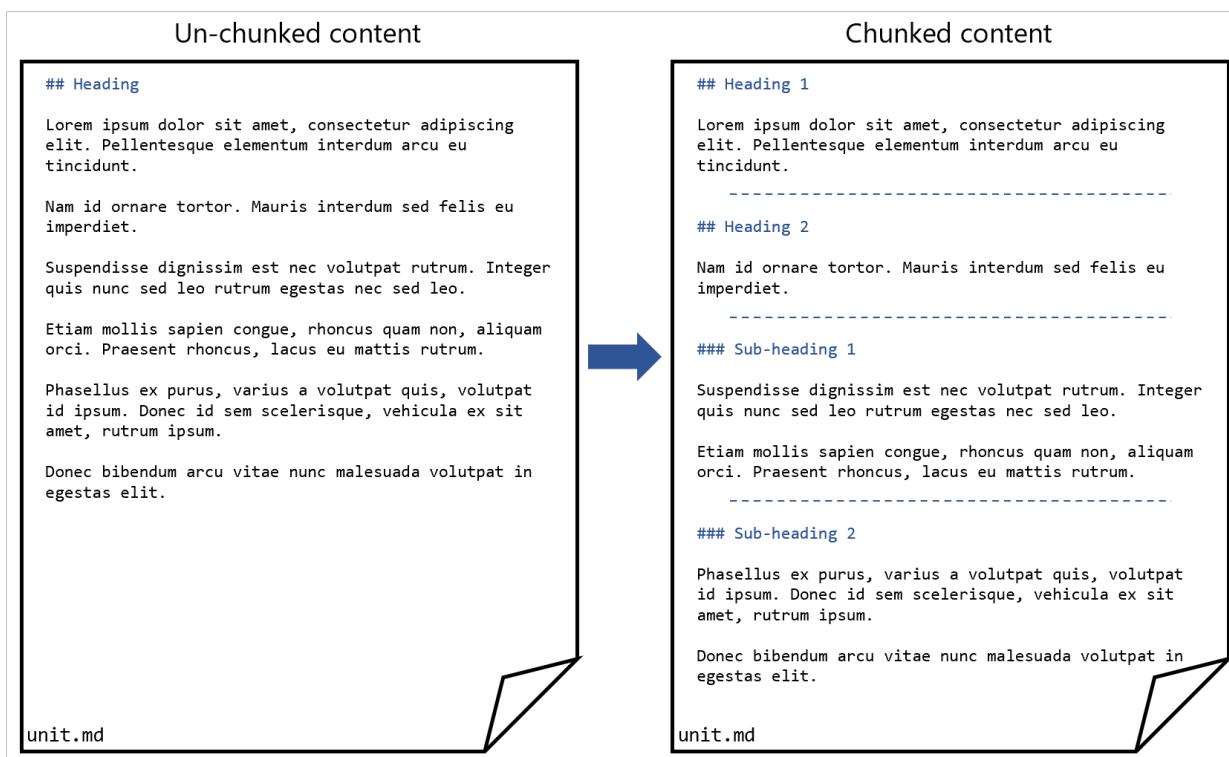
- Create Learn content that transforms into effective PowerPoint slides
- Update existing Learn content to enhance the resulting slides

Overview

This document discusses suggested best-practices to create effective single-source content. The techniques described here are targeted at three types of units: introduction, theoretical learning content, and summary.

Guideline: break your content into chunks

Divide your content into logical pieces by adding headings. The instructional-design community often calls these *chunks*.



This technique is standard in technical writing (for more information, see our [guidance page](#) for more details). Multiple headings help avoid the infamous *wall-of-text* style. The headings also make it easier for users to navigate and consume your content.

Let the content itself determine the size of the chunks; for example, each concept or procedure would get its own chunk. Basing your chunks on the content helps avoid micro-optimizing for one delivery modality such as PowerPoint.

Guideline: provide a strong lead sentence in every chunk

The first sentence of each chunk should summarize the main idea of the entire chunk. This sentence will then be used as the main point on the corresponding slide.

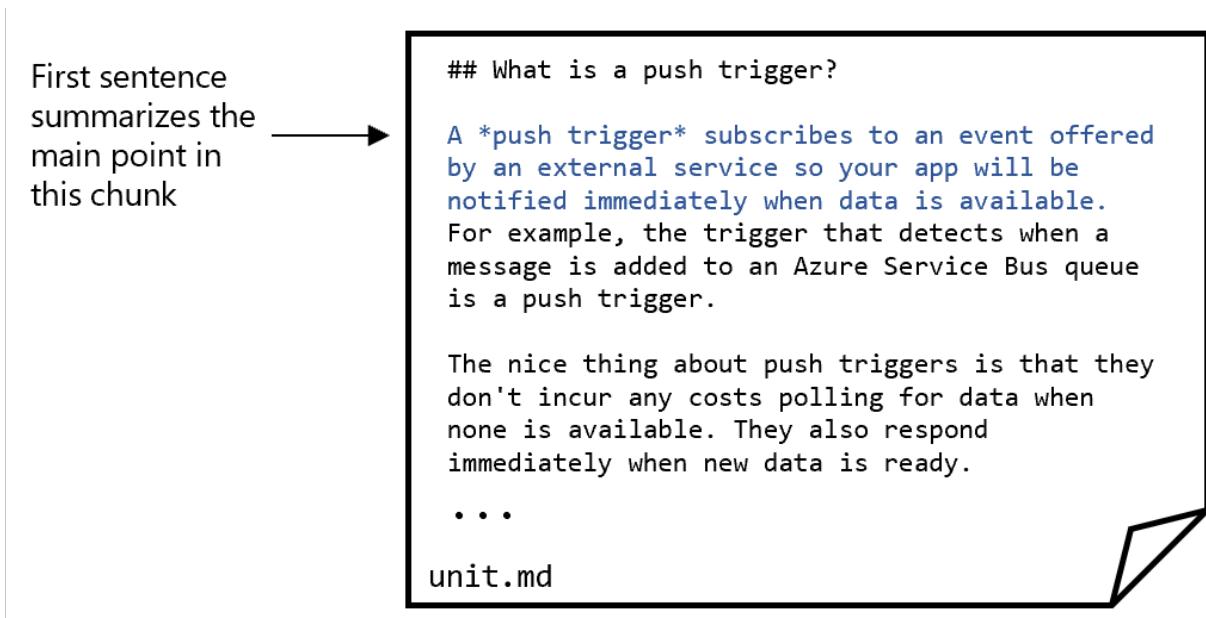


Diagram of markdown source text using the product Azure Logic Apps as an example. The content chunk describes a Logic Apps feature called "push trigger". The lead sentence in the example summarizes the entire concept and reads as follows: A push trigger subscribes to an event offered by the external service to get notified immediately when data is available."

A strong lead sentence has three positive effects when rendered as the main point of a slide:

1. Reminds the instructor of what to cover.
2. Concisely tells the students what they're about to hear.
3. Increases the chance that the visuals used on the slide will logically follow from the sentence.

If it's difficult to write one sentence that covers the entire chunk, that may be an indication the chunk is too large. Consider splitting it into two or more smaller chunks.

Guideline: provide visuals with every chunk

Include at least one visual element in every chunk. The source items that are considered visual elements are:

- Image
- Bulleted list
- Numbered list
- Blockquote
- Code block
- Table

The first two visual elements in each chunk render on the slide. If you have more than two visual elements that are important to the student experience, consider adding another heading so those elements render on a slide. Chunks without any visual elements yield text-only slides.

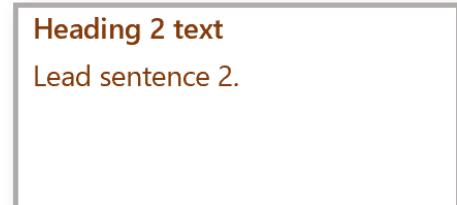
Unit source

```
## Heading 1 text  
  
Lead sentence 1.  
•••  
  
Some more text.  
  
  
## Heading 2 text  
  
Lead sentence 2.  
  
Even more text.  
  
unit.md
```

Slide output



← Two visuals

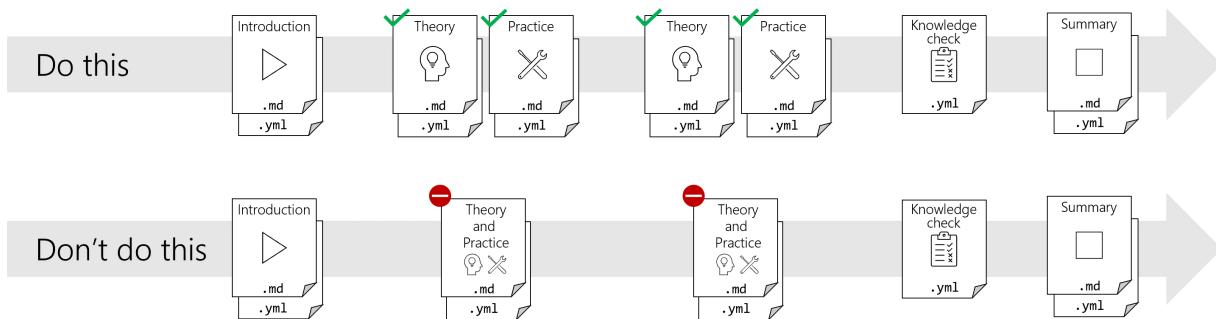


← No visuals

Bulleted lists, numbered lists, and blockquotes are easy ways to add visual elements to textual content. Keep list items short to avoid text-heavy slides. For long list items, consider a bolded 1-2 word summary at the beginning of each item.

Guideline: separate theory and practice content

Split your theoretical learning content into separate units from your practice exercises.



The tooling maps theory units to slides for use as the lecture portion of an instructor-led training course. The practice exercises are intended for students to work through on their own so they yield only a single slide in the output deck.

- If you include practice content in your theory units, that content will render into slides. Detailed, step-by-step instructions are typically inappropriate for lecture in instructor-led training.
- If you include theory content in your practice units, that content won't appear in the slides. The lectures are then likely to be incomplete; that is, they won't contain enough information to prepare the students to complete the practice exercises.

Single-source: PowerPoint template creation (Particle)

1/14/2022 • 2 minutes to read

Caution

This content is subject to change. The Particle tooling is a pilot project and is under active development.

By the end of this article, you'll be able to:

- Create a PowerPoint template with your preferred layouts, fonts, colors, etc.
- Include the required slide layouts and placeholders
- Include any desired optional slide layouts

Overview

The tooling that converts Learn content to PowerPoint uses a template file to determine the look-and-feel of the generated slides. The template controls the slide layouts, the branding elements, and the fonts and colors. This document describes how to create your own PowerPoint template.

Format

The template is a standard PowerPoint file (.pptx) file. You use PowerPoint to create the file and define several slide layouts. Each layout contains named placeholders for text. The tooling will select a slide layout based on the type of source content. It will then fill in the placeholders with select values from the source content.

How to define layouts

You use the slide master view in PowerPoint to create, arrange, and name slide layouts. The layout names are defined elsewhere in this article. Here's a brief overview of the steps:

1. Open the slide master (on the **View** tab, select **Slide Master**).
2. In the thumbnails pane, select an existing layout or insert a new layout (on the **Slide Master** tab, select **Insert Layout**).
3. Set the name of the layout to one of the required or optional layout names (on the **Slide Master** tab, select **Rename**).

For more information, see [What is a slide layout?](#) and [Add, edit, or remove a placeholder on a slide layout](#) for more information.

How to define placeholders

You use the slide master view in PowerPoint to add and configure text placeholders on your slide layouts. The number of placeholders and their required names are defined elsewhere in this article. Here's a brief overview of the steps:

1. Select a layout from the thumbnails pane of the slide master.
2. Insert text placeholders onto your layout (on the **Slide Master** tab, select **Insert Placeholder** and then **Text**).
3. Set the properties of the placeholders like font, position, and so on.
4. Open the selection pane (on the **Home** tab, select **Arrange** and then **Selection Pane**).
5. Select the placeholder in the selection pane. Select the placeholder a second time in the selection pane to enable the edit function. Enter the required placeholder name.

For more information, see [Add, edit, or remove a placeholder on a slide layout](#) and [Manage objects with the Selection Pane](#) for more information.

Required layouts

The following table describes the required slide layouts.

NAME	PLACEHOLDER(S)	APPLICATIONS
<code>title_layout</code>	<code>Title</code> and <code>Subtitle</code>	Used only for the first slide (after the optional <code>cover_layout</code>) in the generated presentation.
<code>section_layout</code>	<code>Title</code>	Used as the first slide for units within a module.
<code>bulleted_layout</code>	<code>Title</code> and <code>Subtitle</code>	Used for agenda, prerequisites, and learning objectives.
<code>content_layout</code>	<code>Title</code> and <code>Subtitle</code>	Used for all slides that don't have a specialized layout define.

Optional layouts

The following table describes the required optional layouts.

NAME	PLACEHOLDER(S)	APPLICATIONS
<code>cover_layout</code>	None	Used only for the first slide in the generated presentation.
<code>exercise_layout</code>	<code>Title</code> and <code>Subtitle</code>	Used for exercises only; <code>section_layout</code> used as fall back if this layout isn't provided.
<code>closing_layout</code>	None	Used only for the last slide in the generated presentation.

Single-source: PowerPoint previewer for Visual Studio Code (Quark)

1/14/2022 • 2 minutes to read

Caution

This content is subject to change. The Particle tooling is a pilot project and is under active development.

By the end of this article, you'll be able to:

- Install the PowerPoint previewer for VS Code
- View your Learn markdown files as PowerPoint from within VS Code
- Customize the previewer to your preferences

Overview

The PowerPoint previewer for Visual Studio Code is an extension that displays your Learn markdown files as PowerPoint slides. The previewer runs in its own view so you can position it just like you do with the standard previews.

It renders the slides using the Particle conversion web service. The preview will be an accurate representation of the final, converted content. You can even select your preferred PowerPoint template from the pre-installed options to ensure that the preview matches the final experience.

Install the previewer

The following are the steps to install the previewer into your local instance of VS Code.

1. Download the [VSIX file](#).
2. Open VS Code.
3. Open the **Extensions** pane by selecting the **Extensions** entry in the **View** menu.
4. Select the **Views and more actions...** entry at the top of the pane. The icon for this entry is an ellipsis.
5. Select the **Install from VSIX...** entry.
6. Use the file navigation window to select the VSIX file you downloaded in the first step.

Use the previewer

The following are the steps to use the previewer.

1. Open a Learn unit markdown file in the text editor.
2. Open the VS Code command palette by selecting the **Command Palette...** entry in the **View** menu.
3. Select the **Particle: Open preview window** entry from the Command Palette. The preview generation should launch in its own view. The conversion is done online via a web service so it may take a few seconds for the slides to appear.

Troubleshoot the previewer

The previewer requires the following conditions in order to run:

- You must have a unit markdown file open in the text editor.
- There must be a folder named `includes` in your source structure.
- The associated unit YML file must exist and be error free.
- The module YML file `index.yml` must exist but doesn't need to be complete or error free.

Customize the previewer

The user-controlled settings for the previewer are located in the preferences area of VS Code. They are in a section titled **Live Preview for Particle** (the menu sequence to locate them is **File > Preferences > Extensions > Live Preview for Particle**).

The following are the settings that control the behavior:

- **Template Name:** lets you specify the PowerPoint template to use in the conversion. You can use any of the templates listed in the **PowerPoint Template** section of the [online conversion portal](#).
- **Update On Save:** lets you control when the slide preview re-renders. If this option is selected, the preview updates automatically when you save the unit markdown file. If this option isn't selected, you'll need to manually re-render the slides by running the command **Particle: Force regeneration of preview for current document** from the **Command Palette**.

Author starter kit

1/14/2022 • 17 minutes to read

Welcome to Microsoft Learn. This article is meant to help guide you through the documentation you will need to author your module(s). The links are provided in the recommended order of content creation. As you move through the process you can refer to this page to find what you need, when you need it. Microsoft Learn strives to be "Learner-obsessed." As such, we have guidelines that aid the learner in acquiring and retaining new skills that they can use over and over again to complete their work. Below is an image of the overall process of putting content on Microsoft Learn. This article focuses on the design process through the monitoring and editing processes.



Start here

You have an idea for content you'd like to publish on Microsoft Learn, great! The first step in the process is to submit a proposal and our Partnerships team will reach out to you to help scope and start the process.

[Request to publish content on Learn](#)

NOTE

Do not continue working through this starter kit until your contribution request has been approved by our Partnerships and Learn Content Experience (CX) teams.

Authoring prerequisites

Make sure you are familiar with the following before beginning a Learn module:

- Ability to read and edit existing YAML
- [Ability to write Markdown](#)
- Basic GitHub skills, including: fork, clone, commit, push, pull request, and merge

Set up your authoring environment

Below are the steps to set up your authoring environment. We'll go through the steps in detail, from the installation of the editor to the configuration of the extension authoring pack.

Before you begin, it's important to notice that all Learn modules are written in a combination of YAML and Markdown. This is why we recommend you to bookmark this page: [Markdown for Docs](#).

Install the editor

Install [Visual Studio Code \(VS Code\)](#).

You can technically use whatever tool you like. If you choose to use tools other than VS Code, we may not provide troubleshooting assistance. All the steps below are for VS Code. There's no support for the Docs Authoring Pack extension for other editors. In this case, you can skip the next section entirely.

Use the authoring pack extension

Microsoft created and supports a VS Code extension to aid in authoring Learn modules. This pack is called the [Docs Authoring Pack](#) and it's available for free.

Install it in your VS Code by clicking on the **Install** button in the extension's page.

After the installation, you'll configure the extension by opening the editor's settings with **Ctrl/Cmd + ,** and expand the **Extensions** menu and select any extension that starts with **Docs** from there.

For the authoring pack to work, you'll need to input your own data in the settings:

1. Select the **Docs Article Templates Extension** from the list
 - a. Change the `Docs > Templates: Alias` configuration to your Microsoft alias.
 - b. Change the `Docs > Templates: Githubid` configuration to your GitHub username.
2. Select the **Docs Scaffolding Extension** from the list
 - a. Change the `Docs > Scaffolding: Alias` configuration to your Microsoft alias.
 - b. Change the `Docs > Scaffolding: Githubid` configuration to your GitHub username.
 - c. Change the `Docs > Scaffolding: Prefix` configuration to the prefix of the module you'll be creating (for modules hosted in `learn-pr` the value should be `learn`). This prefix will be appended to the beginning of every module UUID.
 - d. Change the `Docs > Scaffolding: Product` configuration to reflect the product you'll be writing about, most common is `azure`, however this may vary according to the type of learn module you're focusing on (for more information, see [Product taxonomy](#))
3. Select the **Doc Validation** from the list
 - a. Change `User Type` to `Microsoft Employee`

You can also tweak the other configurations to your needs and tastes. In the next sections, you'll see how to use the authoring pack extension to author your module.

Design your module

All Learn modules go through a design process. The design document helps you think through the content and flow of your module and to optimize it for best instructional design practice.

- [What is the Design document?](#)
- [Design document Markdown template](#)
- [View an example Design document](#)

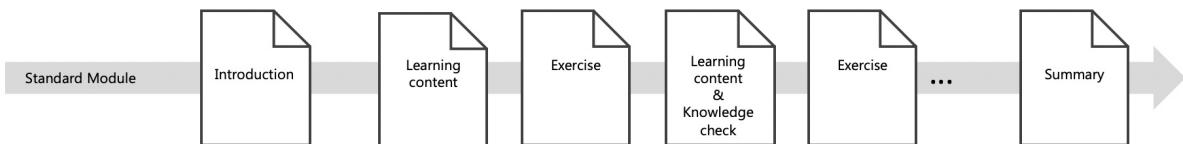
Sections of the Design document

- [Title](#)
- [Roles](#)
- Level: Beginner, intermediate or advanced (documentation coming soon)
- [Product\(s\)](#)
- [Prerequisites](#)
- [Summary](#)
- [Learning objectives](#)
- Chunk your content into subtasks (documentation coming soon)

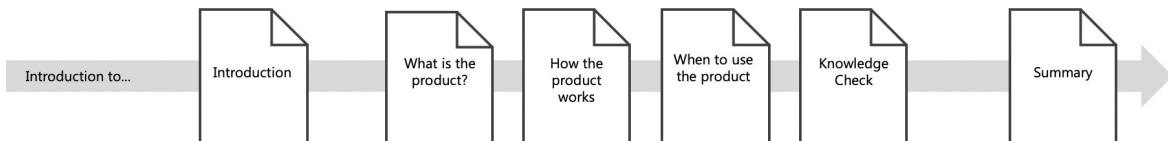
Outline the units

Microsoft Learn modules follow one of three basic patterns, depending on the type of module. The following diagrams describe the patterns.

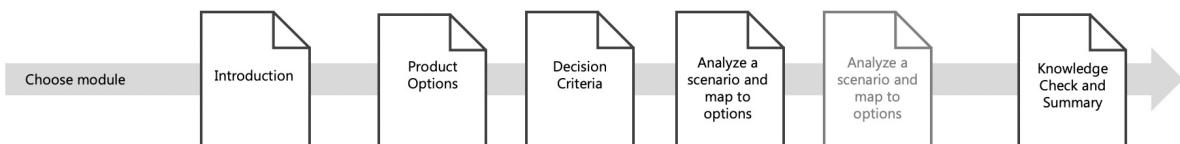
Standard module:



Introduction to ... module



Choose module



Listed below are sections of content you'll need to outline for your Design document. Depending on the type of module you are designing, you may use each section more than once.

- Introduction
 - Scenario
- Learning content unit **title** *repeat for each unit in the module*
 - List enabling **objectives** - These are lower-level objectives that define what needs to be taught in order for the learner to accomplish the unit task. - For each enabling objectives, create a list of topics that will be covered. - Consider and note places where video could be helpful in explaining a concept.
- **Knowledge Check** *repeat for each knowledge check in the module*
 - You don't need to write your knowledge checks here. Just note what you want to test. Remember that your knowledge check questions should align directly with your learning objectives.
- Exercise *repeat for each exercise in the module* (documentation coming soon)
 - List the steps to complete the proposed exercise, make sure all of the content needed to understand why they are doing these steps is covered in the unit content preceding the exercise.
- **Summary**
 - Provide a simplified outline of your summary.

Build module scaffolding

Once your Design document is complete, you'll need to scaffold your module. The scaffolding is a skeleton of your module in Markdown and YAML files.

- [Scaffold Manually](#)
- [Scaffold with the Docs Authoring Pack](#)

IMPORTANT

Module SEO is important for a successful Learn Module. Please check the [SEO Best Practices guide](#) to ensure your module is SEO-friendly and can be discovered by the audience.

Fork the appropriate Learn GitHub repository

Learn modules are submitted via GitHub. Each content area has its own repository. It's helpful to have a local copy of the repository up and running before you author so you can save your work and collaborate with the Learn team.

- [Microsoft Learn GitHub repositories](#)
- [Contact learn-repo-managers@microsoft.com](mailto:learn-repo-managers@microsoft.com)

When forking the appropriate repository, clone it to your local machine (it can take a few hours) and add a new remote that points to the original Microsoft Learn GitHub repository. This will ensure that you can always keep your local fork updated with the latest changes.

You can add a new remote by running the following command:

```
git remote add upstream <the original repository URL>
```

And then update by fetching from the remote:

```
git pull upstream <branch>
```

TIP

If you're starting a new module or want to contribute to an existing one, it's highly recommended to run

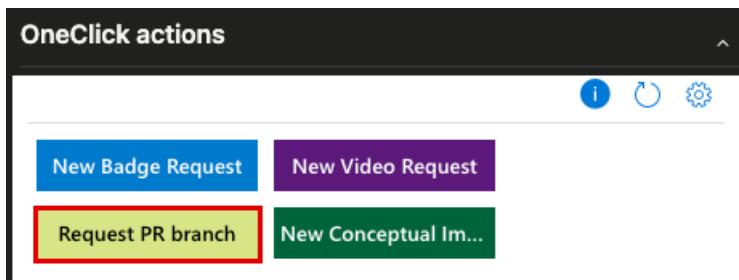
```
git pull upstream master
```

before starting any changes or creating any new branches.

Request an upstream branch

All new modules need to live in a new branch that starts with `NEW-` (for example `NEW-aks-deploy-container-app`). This branch needs to be created both on your fork and on the original repository. However, since the Docs repositories are huge and receive thousands of commits every hour, it's recommended to create a new branch **only** on your fork first and then submit the request for a new branch to the original repository only after the module is completed and ready to be merged.

A new upstream branch can be requested in the *work item* created for your module in your DevOps board (ask your PM where this work item lives) by clicking the "**Request new PR Branch**" button in the "Oneclick Actions" section.



This will prevent further errors down the pipeline, since the build system can get confused when there are many different changes between the upstream and your fork.

The final flow should be:

1. Fork the appropriate repository
2. Add the upstream origin

3. Create a new branch called `NEW-<your module name>` in your fork
4. Push this branch to your fork
5. Design and create your module
6. Once the module is finished, request the creation of the upstream branch
7. Once the branch is created, open a new PR to merge **your branch** into the **upstream branch** (note: **never** merge new modules to the main branch)

IMPORTANT

Avoid merging the main upstream branch into your new branch once you create it. You can update your main branch with `git pull upstream master`, but don't merge it into your module branch, not even after the PR submission. Since your branch is behind the `master` branch, and the upstream branch will be created after it, it'll have all the newest changes, and yours won't conflict with them. Failing to do so, may result in build failures.

Request and add Achievement artwork

All Learn modules are required to have achievement artwork/badges. To have this ready in time for publishing, we recommend you put in the request before you begin authoring.

- [Request and add achievement artwork](#)
- [Add achievement to module](#)

Author your module

There are two main approaches to authoring the content:

- Linear
 - With this approach, the author starts with the introduction and works their way forward until they reach the summary.
- Backwards
 - With this approach, the author begins with the culminating activity and supporting exercises. They then write the learning content to support each exercise. The introduction and summary are completed last. This approach is recommended by the Learn ID Team, particularly for highly technical content.

The following is a list of all of the instructional guidance for each portion of your module. They are presented in the order they will appear in the module, but you should use each as you work on that section of the module.

Acrolinx

Acrolinx is Docs' spell and grammar checker. Acrolinx is always enabled in all PRs and modules you'll submit. It's also included in the Docs Authoring Pack, and its icon can be seen in the top right of the editor when editing a Markdown file for a learn module.

To have a valid module, all the checks should be over the 80-point threshold. It's recommended to run Acrolinx manually once you finish a file to ensure that it's valid. If any of the files present a score that's below 80, you'll need to fix the spelling and grammar errors presented until the score is 80 or higher.

IMPORTANT

Modules with Acrolinx scores lower than 80 will not be accepted as a valid PR once submitted.

You don't need to fix all warning it gives you, but you should fix the ones that are important, and all the ones that are presented as errors. Some of the errors are due to the content you're writing, and others are due to the

way you're writing it. So most solutions include rewriting the phrase or the paragraph to make it shorter or more readable.

Other errors are more soft-proofed, like the use of an object after "this/that/these/those". In most cases, these errors won't change your overall score and can be safely ignored.

Be mindful of the Acrolinx scores while writing, and you can always check its scores in the PR after a successful commit:

The screenshot shows a GitHub pull request interface with an Acrolinx scorecard. The title bar says "acrolinxatmsft1 (Acrolinx at Microsoft 1) 8 days ago". The main content is titled "Acrolinx Scorecards" and includes a note: "A minimum Acrolinx score of 80 is required. Click the scorecard links for each article to review the Acrolinx feedback on grammar, spelling, punctuation, writing style, and terminology." Below this is a table with the following data:

Article	Acrolinx score	Lowest category score	Spelling issues	Scorecard	Processed
learn-pr/azure/aks-network-design-with-azure-cni/1-introduction.yml	100	100	0	link	✓
learn-pr/azure/aks-network-design-with-azure-cni/2-overview-of-kubernetes-networking.yml	100	100	0	link	✓
learn-pr/azure/aks-network-design-with-azure-cni/3-estimating-cluster-size-pods.yml	100	100	0	link	✓
learn-pr/azure/aks-network-design-with-azure-cni/4-estimating-cluster-size-nodes.yml	94	67	0	link	✓
learn-pr/azure/aks-network-design-with-azure-cni/5-sizing-the-network.yml	100	100	0	link	✓
learn-pr/azure/aks-network-design-with-azure-cni/6-exercise-deploy-aks.yml	100	100	0	link	✓
learn-pr/azure/aks-network-design-with-azure-cni/7-knowledge-check.yml	98	93	0	link	✓
learn-pr/azure/aks-network-design-with-azure-cni/8-summary.yml	100	100	0	link	✓
learn-pr/azure/aks-network-design-with-azure-cni/includes/1-introduction.md	94	76	0	link	✓

NOTE

Note that the scores are also given to YML files. These can be ignored.

Introduction unit

- [Create a unit](#)
- [Introduction](#)
- [Learning objectives](#), can be copied over from Design document
- [Prerequisites](#), can be copied over from Design document

Learning content units

- [Create a unit](#)
- [How to structure your learning content](#)
- Video *if applicable* - [Video guidance](#)

- Images *if applicable* - Add a screenshot - Add an expandable screenshot - Add Conceptual Art - Add an animated gif

Exercise units

One of the best features of Learn is the ability to add exercises that allow the learner to practice what they've learned. The following resources will help you to create your exercises.

- [Create a unit](#)
- [Structure of exercise units](#)
- [Types of labs](#)

Hosted environments (also known as: Sandboxes)

If your content uses Azure and can be hosted in a sandbox, this is a recommended approach. Sandbox modules will allow the learner to go through your module without needing to bring a subscription to try the exercises and you, as an author, will have more control over the environment used to deploy the resources needed to build the module environment.

You can create a new sandbox by following the steps below:

- [Create an exercise using a hosted environment](#)

Bootstrap scripts

It's a good practice to include your sample repository in the [Azure Sample](#) organization. This will ensure that the code is secured and it's easy to find and use.

If your sample needs more complex code or previous infrastructure, you can create a bootstrap script that will be used to build the environment for your module. This script should be run in the "Introduction" unit of your module like [in this example](#).

Best Practices

- If using bash, make sure your learners will be using either a sandbox environment or instruct them to use bash as their shell.
- Create an `aka.ms` link for your bootstrap script and use `CURL` or other request library to fetch the script and run it in a one-liner, removing the need to clone the repository or download the file.
- Try to use a stateless bootstrap script, so it does not need any external dependency and can be reused in other modules.
- Avoid fixed names. Let all naming open as environment variables and use `$RANDOM` to avoid name collisions on resources that cannot be duplicated (like ACRs).
- Export all the needed variables (like names, FQDNs, DNS names, resource groups) from the script so they can be used in the shell afterwards.
- Make the command output information to the user at every step, so they know what's happening.
- Avoid the use of `sudo` in your bootstrap script. It's not necessary and it can cause problems.

Docker images and containers

If your content uses Docker and can be hosted in a container, this is a recommended approach. Using a container image guarantees that your module will have the same dependencies every time it's deployed. This is a great way to ensure that your module will always be functional and using the desired dependency versions.

However, unless your module teaches or needs the user to build a new Docker image, it's recommended to host the image in a registry. Microsoft has the MCR (Microsoft Container Registry) available to all Learn content under `mcr.microsoft.com/mslearn`. There's no formal process on how to add a new image to the registry since it requires non-delegated permissions. For now, reach out to [Lucas Santos](#) for more information.

IMPORTANT

The above process is prone to change at any time. Please be sure to check this page often for updates.

No modules should ever pull content from a public container registry (for example ghcr.io, docker.io, gcr.io) other than MCR. Public container images should either come from MCR where available, or be `az acr import` ed into an ACR instance deployed as part of the module. If it is necessary to use a public container image, add a disclaimer with a reference to [the concerns of public content](#)

Zone pivots

Zone pivots allow you to reach a wider audience by providing distinct instructions for things like programming language or operating system.

- [Add zone pivots](#)

Knowledge checks

- [Create a Knowledge check](#)
- [Knowledge check guidance](#)

Summary unit

- [Create a unit](#)
- [Summary format](#)
- You may add a section on further reading or references in this unit

Add metadata to index.yaml

The index.yaml file defines two categories of information about the module. First, it contains metadata about the module (for example, the title, the author). Second, it contains information about the module's structure (for example, the filenames for the units, achievement artwork, and module icon). The Docs build system parses this file to publish your module.

- [Create the UID](#)
- [Metadata documentation](#)
- Title: copy final title over
- [Summary](#)
- Level: include as a list (defined in Design document)
- Role: include as a list (defined in Design document)
- Product: include as a list (defined in Design document)
- Units: include as a list in the following format: [UID].filename
- Badge: [UID].badge

Prepare for publishing

Once your module is complete, use the following docs to help you prepare it for publishing. The closer your content aligns to these lists, the more successful the publishing process will be.

- [Build a Learn url](#)
- [Pre-publishing checklist](#)
- [PR Review cheat sheet](#)
- [Optimize for SEO](#)
- [Content requirements](#)
- Localization (documentation coming soon)

Submit for publishing

Once the module is fully completed and the PR branch is created, you can submit a new PR for reviewing. The target branch is always the branch with the same name in the upstream repository, whilst the base branch is the branch you're currently working on.

Follow the steps below to submit a PR for review:

- [Pull request best practices](#)
- [Review and publishing schedules](#)

Once the PR is created and your peers (your PM and other people from your local team) have reviewed it, you can request a review from the editorial team by commenting `#sign-off` in the PR. (This can only be done after a successful Acrolinx review and a successful build.)

This comment will tag the PR as ready for review and start the editorial review process described in the documents above.

If you ever need to hold your PR again, just comment `#hold-off`, and the PR will be taken out of the review queue.

Monitor and get data about your content

Once your module is published, you'll want to keep an eye on how it's doing and being perceived by your customers. The following are links to help you monitor and get reporting on your module.

- [Kusto query library](#)
- [MS Learn Data Warehouse for AAD users](#)
- [Microsoft Learn Power BI report](#)
- [MDM data](#)

Field tips from other authors

We've gathered a few tips from other authors to help you get started.

1. Remove all comments on the `yml` files created by the Docs Extension as it will cause your module to be sent back for review and removal in the future.
2. Be careful with the naming of your fictional companies/products for the samples. Some names can have negative connotations in some places and can be confusing in others, for this, **always** use the [fictional naming guide](#) to name all the resources within your module text.
3. Create upstream (PR) branches the latest as possible, as it will help your module be less prone to errors.
4. Never merge the main branch into the PR branch after you've created it, unless needed or told to do so by the build or docs team.
5. Keep your fork updated by running `git pull upstream master` in your master branch. This will save you much time from having to download thousands of accumulated commits at once.
6. In your modules, never hardcode names and/or paths. Instead, use environment variables.
7. Use `$RANDOM` in places where naming collision needs to be avoided. But mind that not all shells possess this feature.
8. Prefer bootstrap scripts than bootstrap units.
9. Request the achievement right after you start your module, it can take a while.
10. Remove all language marking in Microsoft websites when pasting them. Instead of <https://docs.microsoft.com/en-us/learn>, use <https://docs.microsoft.com/learn>.
11. Store your samples in the Azure Samples organization in GitHub.

Simplified flowchart of the publishing process

Below is an simplified version above guidance of the publishing process. It's not meant to be a complete guide to the process and you're encouraged to read this whole documentation to know about a few important details.

1. Get access to your DevOps Board so you can track your content
2. Open your assigned work item in the DevOps Board and click to request a new badge
3. Fork the learn repository
4. Create a new local branch from the main branch using the `NEW-<module uid>` naming convention
5. Use the Docs Authoring pack to scaffold your module
6. Copy the design document layout from this page into the `resources` folder
7. Create the design document and review it
8. Update the YAML files with the module's metadata and info from the design document
9. Start writing your module's units and exercises
10. Upon finishing, check if your module needs a sandbox, if so, use the PPE sandbox to test if it will have any issues
11. Open a new PR branch request
12. Request a module review from your peers
13. If your badge is already completed, add it to the module YAML files and to the learn repository folder
 - a. If not, you'll need to add it later on either in other PR or wait until the badge is completed before signing off
14. Open a new PR in the learn repository targeting the PR branch
15. If your module needs a sandbox and you've already tested it in the PPE environment, fill the form to request a sandbox instance (you'll need the module UUID and the PR number)
16. Wait for the build and Acrolinx processes to finish
 - a. Upon failed builds or Acrolinx issues, correct those issues and commit again
 - b. If the build still fails, submit a request to your PM or the learn support team
17. `#sign-off` the PR and await the review. Once the review is completed, correct any errors and `#sign-off` again
18. After the PR is approved and merged your module will be published

Authoring guidelines

1/14/2022 • 2 minutes to read

This page lists the content design guidelines for Learn contributors.

Each guideline has an embedded explanatory video. The videos are also listed on our channel <https://aka.ms/learn/idvideos>.

Please [email](#) with questions and feedback. Thank you!

Instructional design office hours

We have weekly office hours on Tuesday from 11am-12pm Pacific time. Whether you're currently writing content for Learn or just have some questions about how it works, please stop by and say hello! We'll help you plan your content or answer specific questions about how to write titles, learning objectives, knowledge checks, etc. Office hours are first-come-first-served with no appointment needed.

- [Office hours Teams meeting link](#)

Module planning

- Module design ([overview](#), [template](#))

Module structure

- [How to structure standard modules](#)
- [How to structure "Introduction to ..."](#) modules
- [How to structure "Choose ..."](#) modules

Unit structure

- [How to structure a learning content unit](#)
- [How to structure an exercise unit](#)

Detailed guidelines

- [Quick reference: guideline summary all on one page](#)
- [How to write titles](#)
- [How to write introductory summaries](#)
- [How to write learning objectives](#)
- [How to write prerequisites](#)
- [How to write scenarios](#)
- [How to write introductions](#)
- [How to write knowledge checks](#)
- [How to write module summaries](#)

Content requirements for Microsoft Learn

1/14/2022 • 6 minutes to read

This page summarizes the requirements for content hosted on Learn.

Priority 0 requirements

Content must meet all of these Priority 0 requirements before initial publication or it can't be hosted on Learn. Content must also meet the Priority 1 requirements within 30 days of initial publication.

1. Content must meet the [pull request quality criteria](#) in the Docs Contributor Guide. Please note that the linked page includes places where the rules for Learn and Docs differ; for example, H1 headings are not allowed in Learn Markdown files.
2. Content can't violate any [legal](#) or [accessibility standards](#).
3. Content must be readable and checked for spelling and grammar.
4. Each published file must have a minimum Acrolinx score of "80".
5. Modules must follow the core Learn structure: begin with the standard introduction unit, contain learning content and interactivity units, and end with a summary unit.
6. Exercise instructions must be complete and clear (that is, anyone should be able to follow the instructions to complete the exercise without any additional required steps).
7. Content cannot be copied directly from other sources such as <https://docs.microsoft.com>.
8. Titles must follow the [How to write titles](#) guidance.
9. Modules must include conceptual learning content (for example, can't solely be a walk-through).
10. Modules must include at least one Knowledge Check or exercise with Task Validation (we don't want a module that can be completed with "view only").
11. Modules must be standalone; that is, they must be able to exist in multiple learning paths (for example, exercises should not span across modules).
12. Modules must be organized around 1-2 focused tasks with a total completion time of less than 60 minutes.
13. Modules may include short supplementary videos but the content can't be primarily video-based.

Priority 1 requirements

Content must be written to follow the guidelines for Microsoft Learn content.

1. Introductory summaries follow the [How to write introductory summaries](#) guidance.
2. Prerequisites follow the [How to write prerequisites](#) guidelines.
3. Module learning objectives follow the [How to write learning objectives](#) guidance.
4. Module and unit introductions follow the [How to write introductions](#) guidance.
5. Knowledge Checks follow the [How to write knowledge checks](#) guidance.
6. Module summaries follow the [How to write module summaries](#) guidance.
7. All video closed captioning files must be localized into all the languages that your Learn textual content is translated into.
8. Content must be kept up to date with any concepts, functionality or user interface changes to the latest version of the software that is displayed in images or videos.

Priority 2 requirements

These are rules which we expect most content to follow.

1. Modules must be organized around job tasks (for example, "task oriented" rather than "topic oriented").
2. Content must be factual rather than opinion or marketing (for example, technical content is "engineers teaching engineers").

Detailed requirements

This section summarizes the full set of Learn rules and ideal guidelines. It expands upon the above rules with additional details and requirements.

Learning path

- Between 3 and 8 modules
- Modules are all related to solving one high-level job task
- Modules must be standalone; that is, they must be completable if the learner has the prerequisite knowledge even if they have not completed the previous modules in the path (for example, exercises should not span across modules, no statements like "as you saw in the previous module...")
- Title: follows the rules in the "Title" section
- Summary: follows the rules in the "Summary" section
- Prerequisites: follows the rules in the "Prerequisites" section

Module

- 60 minutes or less
- Between 4 and 10 Units
- Must start with a unit named "1-introduction.md" consisting only of introduction, learning objectives, and prerequisites (no learning content, no exercises)
- Must be standalone; that is, they must be able to exist in multiple Learning Paths (exercises should not span across Modules)
- Must include conceptual learning content (that is, can't solely be a walk-through)
- Units are all related to solving one task
- Includes at least one Knowledge Check or exercise Task Validation
- Does not have two or more units in a row without an active-learning component (Knowledge Check or Exercise)
- Provides cleanup instructions for any Azure resources created in the learner's account
- Title: follows the rules in the "Title" section
- Summary: follows the rules in the "Summary" section
- Prerequisites: follows the rules in the "Prerequisites" section
- Learning objectives: follows the rules in the "Learning objectives" section
- Introduction: follows the rules in the "Introduction" section

Unit

- No more than 10 minutes for learners to consume
- Learning content must include everything needed for the learner to complete any subsequent active-learning activity
- Graphics may be included
- Video may be included to supplement the textual content
- Images must have appropriate license/approvals
- Screenshots must follow standard Microsoft Docs guidance for borders and callouts
- No more than 4 screenshots
- Graphics/images/screenshots must have alt-text that follow standard [Microsoft Docs guidance](#)
- Include definitions for all introduced terminology

- Include definitions for all used acronyms at first use
- Content must not be copy/paste from other sources such as docs.microsoft.com
- Azure tasks that can be validated must have Task Validation included in the Unit .yml file
- Company names used in examples or exercise must be taken from the Microsoft CELAWeb [Approved fictitious names & guidelines](#)
- Links should not be localized
- Title: follows the rules in the "Title" section
- Introduction: follows the rules in the "Introduction" section

Videos

See the [detailed requirements](#).

Exercises

- Must be based on real-world scenarios (for example, no "hello, world")
- Code samples should include multiple programming languages
- Coding exercises should ask the learner to write some original code (that is, not solely cut-and-paste)
- Provided source code should be in GitHub
- Provided source code must be under the MIT license
- Code on GitHub should be public and not require authentication/authorization to access
- Code samples must follow modern coding practice
- Don't use code comments to present new learning content (you can use them to connect previous learning content to the code sample)

Sandbox

- State is not maintained across modules, exercises can't span modules
- Include the azure-sandbox-cleanup file in the summary if the module uses the Azure sandbox

Title (applies to learning path, module, and unit)

- Follows the [How to write titles](#) guidance
- Written in sentence case (that is, only the first word and product names should be capitalized)

Introductory summary (applies to learning path and module)

- Follows the [How to write introductory summaries](#) guidance
- Format and concept are similar in learning path and module, only the scale is different (that is, the tasks will be larger at the learning path level)

Prerequisites (applies to learning path and module)

- Follows the [How to write prerequisites](#) guidelines
- Placed in the `prerequisites` property in index.yml
- For a learning path, the prerequisites are the union of the component module prerequisites
- For a learning path, the prerequisites statement may be "None" if all component modules have no prerequisites

Learning objectives (applies to module)

- Follows the [How to write learning objectives](#) guidance
- Placed in the `abstract` property in index.yml

Introduction (applies to module and unit)

- Follows the [How to write introductions](#) guidance
- Includes the learning objectives, placed after the prose content
- Includes the prerequisites, placed after the learning objectives

Knowledge check

- Follows the [How to write knowledge checks](#) guidance
- A Knowledge Check embedded in a unit must have 2-3 questions
- A Knowledge Check in a standalone unit must have 3-5 questions
- A standalone Knowledge Check unit is allowed only when it is the penultimate unit in a module
- The module summary unit cannot contain a Knowledge Check

Module summary unit

- Follows the [How to write module summaries](#) guidance
- Restate the scenario problem(s)
- Show how you solved the scenario problem(s)
- Describe the experience without the product
- Describe the business impact

Module design

1/14/2022 • 2 minutes to read

In this article, you will:

- Identify the purpose of the module design document

What is a module design?

A module *design* is your plan for how you'll write your content. The design captures core information like your title, learning goals, prerequisites, and unit structure. It also shows how you'll add interactivity to your module. We have completed examples as part of the reference content on our instructional design [landing page](#).

What format should I use?

Use our standard format that includes sections for title, learning goals, prerequisites, units, exercises, and knowledge checks. We recommend you start with our [template](#). Our standard implementation is Markdown because it's compact and works well in GitHub. You can use Word or plain text if you'd prefer. But, be sure to include all the standard sections from the markdown template!

Why create a design?

Our modules teach things our customers do at work. We teach our customers by organizing the content around job tasks. As such, our content is *task-centered* and not *topic-centered*. The tasks we teach won't be an exact match to what our customers are doing on-the-job. That's why we always include the facts and concepts that underlie what we teach. The combination of task-based instruction and underlying principles helps our customers transfer the new skills to their own work.

A good module design document helps us by:

- Guiding the author to create a task-centered module, which makes the content easier to learn.
- Simplifying the writing process since more planning is done up front.

How to structure standard modules

1/14/2022 • 5 minutes to read

In this article, you will:

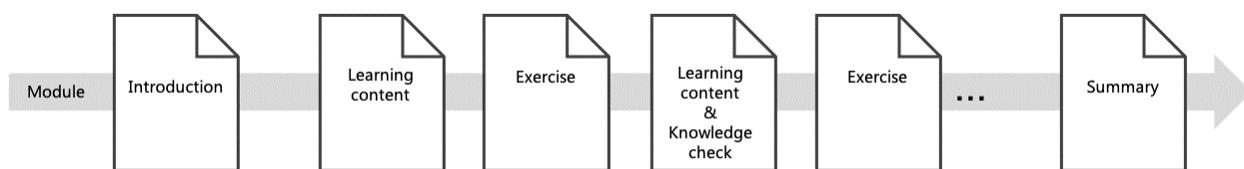
- Organize content in an approved sequence

Overview

It's common for learners to take online training and then be unable apply it to a real-world scenario. Why is this? One reason is that training is often structured as a walkthrough. Learners are given steps to follow with no explanation of what they are doing or why. Walkthroughs can help experts but aren't ideal for novices:

- Novices don't have enough existing knowledge to organize and contextualize the new information. They tend to focus on surface details (like the steps themselves) and can't see the larger patterns involved in the solution.
- The nature of a walkthrough requires concrete steps that lead to a specific goal. This constraint means they tend to cover *well-structured* problems. That is, they are problems with clearly defined solutions. Most real world problems tend to be *ill-structured*. They have many possible solutions and require deep knowledge and analytical skills to solve. Novices are generally unable to extrapolate the steps of a walkthrough to solve a real-world problem.

We want novices to be able to apply what they learn in our modules to solve ill-structured tasks. This is how we help them be successful at their own work. Learn modules should teach learners what they are doing in each step and why they are doing it. We use the term *learning content* for content that explains this *what* and *why*. We've created some rules for module structure to ensure every module contains learning content. The basic layout looks something like this:



Definition

- A Learn **module** is a sequence of learning-content units mixed with opportunities to practice and/or test the new knowledge.

Importance

- **Learner:** Provides the learner with the competencies and skills they need to perform tasks in real world situations.
- **Author:** The standard format simplifies module planning and helps the author build effective training content.

Video

The following video summarizes this guideline.

Rule: Provide learning content units

Details

- For any exercise unit, there must be at least one unit of learning content immediately preceding it.
- The learning-content unit will cover the facts, concepts, processes, etc. the learner will apply in the exercise.

Example 1

Module: Route and process data automatically using Logic Apps

UNIT TYPE	UNIT NAME	UNIT CONTENT
Learning content	Detect an external event using a trigger	Trigger types What is a polling trigger? What is a push trigger? Trigger parameters and return values How to create a Logic App in the Azure portal How to add a trigger using the Designer
Exercise	Exercise - Create the social-media tracker Logic App	Create an Azure Logic App Add a Twitter trigger Run the app and examine the results

Example 2

Module: *How Plain English improves your writing*

UNIT TYPE	UNIT NAME	UNIT CONTENT
Learning content	Be direct using the <i>active voice</i>	What is the active voice? Active voice vs. Passive voice Why use the active voice? How to change text from passive voice to active voice
Exercise	Exercise - Rewrite a paragraph in the active voice	Examine a paragraph Identify where the passive voice is used Rewrite paragraph in the active voice

Rationale

- Learning-content units provide learners with the information they need to solve ill-structured problems in their real-world tasks.

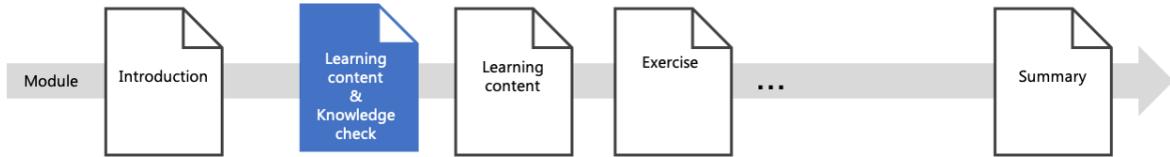
Rule: Provide opportunities for practice

Details

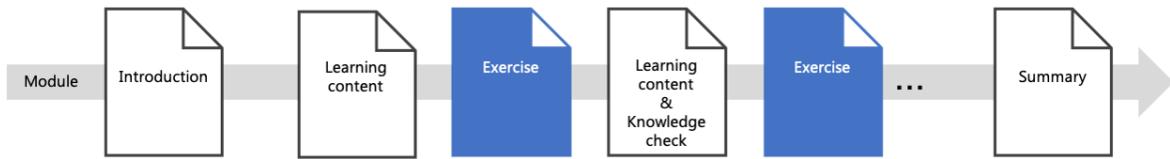
- All Learn modules must contain interactive elements that allow the learner to practice or assess what they've learned. These can be:
 - Knowledge checks embedded in the learning content units
 - Exercise units following the learning content units
 - Standalone knowledge check unit titled "Knowledge check" as the penultimate unit of the module

Examples

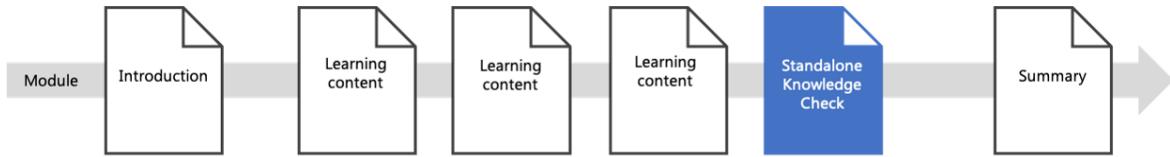
- Knowledge checks embedded in the learning content units



- Exercise units following the learning content units



- Standalone knowledge check unit titled "Knowledge check" as the penultimate unit of the module



Rationale

- Learning content is most effective when the learner gets the opportunity to apply what they've learned. This helps to put it in the context of the product environment and provides a concrete way to teach the steps of the task.
- Learners need to actively think about the content in order to remember it. Practice keeps the new content active in memory which helps encode information into long-term memory.

Rule: Use a unit sequence that follows the content and ordering rules

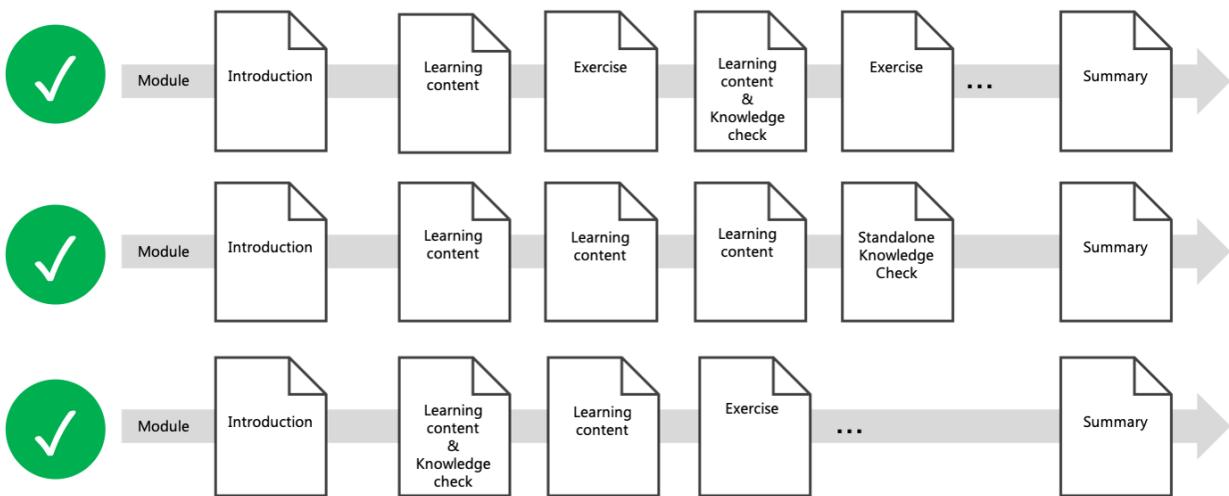
Content and ordering rules

- Must begin with an introduction unit that follows the [introduction](#) rules
- Must include learning content
- Must include opportunities for practice
- Must end with a summary unit that follows the [summary](#) rules

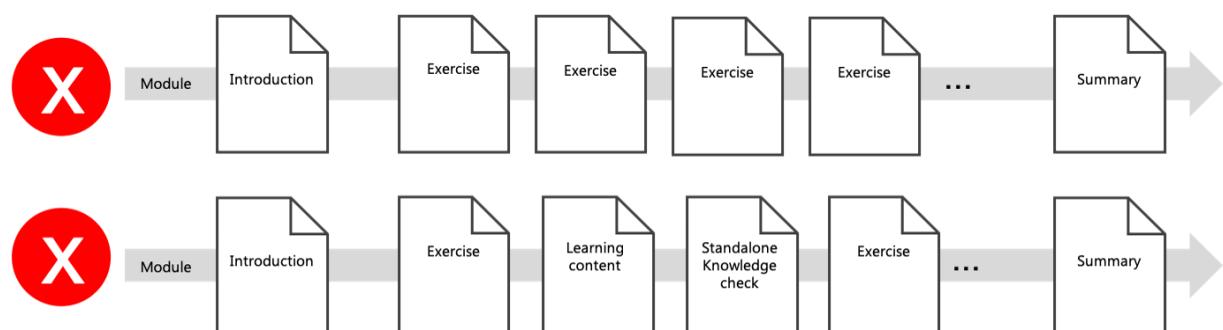
Example

Module patterns

Analysis



- Modules include approved sequence of learning content and practice.



- Module has no learning content, only exercises
- Module contains a standalone knowledge check which isn't penultimate

Rationale

- The introduction answers "what's in it for me?" by describing a real-world problem.
- A logical mix of learning content and practice helps the learner achieve competency in the product.
- The summary recaps how to solve the problem posed in the introduction.

Rubric

Use this rubric to evaluate your work. For each row, find the column that most-closely matches your content and note the number of points in the header of that column. Repeat the learning-unit row for each learning unit. Repeat the exercise-unit row for each exercise unit. Add up the points earned from the rows. The goal is a score of $(2 * \text{number of learning units} + 2 * \text{number of exercise units} + 9)$ or higher.

POINTS	3	2	1
Module sequence follows the content and ordering rules	Fully e.g. <i>Module begins with introduction, ends with a summary, and has a mix of learning content and opportunities for practice</i>	-	Not at all e.g. <i>Module is missing an introduction or summary and/or does not have a mix of learning content and practice</i>

POINTS	3	2	1
Use the standard module-introduction format	Fully e.g. <i>Module begins with an introduction that follows the scenario guidelines.</i>	-	Not at all e.g. <i>No introduction included or includes an introduction that doesn't meet the guidelines</i>
Provide learning content units	Fully e.g. <i>Every exercise has a learning content unit that immediately precedes it</i>	Mostly e.g. <i>Most exercises have learning content units that precede them</i>	Not at all e.g. <i>Very little or no learning content included</i>
Provide opportunities for practice	Fully e.g. <i>Module has opportunities for student to practice or test their knowledge on most of the content in the module</i>	Mostly e.g. <i>Module provides some practice or a knowledge check, but some learning-content units do not have practice</i>	Not at all e.g. <i>No exercises or knowledge checks included</i>
Standalone knowledge check is penultimate unit	Fully e.g. <i>If module contains standalone knowledge check, it is the penultimate unit</i>	-	Not at all e.g. <i>Standalone knowledge check is not penultimate</i>
Use the standard summary-unit format	Fully e.g. <i>Module ends with a summary that meets the summary guidelines</i>	-	Not at all e.g. <i>Module does not have a summary or includes a summary that doesn't meet the guidelines</i>

How to structure "Introduction to <product>" modules

1/14/2022 • 8 minutes to read

In this article, you will:

- Structure your *Introduction to <product>* modules in the standard format
- Determine which content to include in your *Introduction to <product>* modules

Overview

Suppose you hear about a new product that sounds interesting. You've never used it and don't know much about it. Where do you start your research? What are your immediate goals?

We think customers in this situation are trying to decide whether the product is right for them. We'd like to help them make their decision. We also want to put a START HERE indicator on this content so it's easy to find. From our customers' point of view, helping them make this decision is what an *Introduction to <product>* module is for.

As an author, you get a place to put the fundamental information about your product. For example, what features does it offer? What vocabulary do experts use when talking about it? What types of problems does it solve? Learners need to know these basics to do anything with your product. *Introduction to <product>* modules give you a natural way to factor out this content from your other modules and consolidate it in one spot.

The module structure and content are precisely defined. This consistency lets us make a guarantee to our customers: every *Introduction to <product>* module helps them decide if the product will meet their needs. Here's the basic organization:

UNIT	CONTENT
1	Introduction
2	What is the product?
3	How the product works
4	When to use the product
5	Knowledge check
6	Summary

This article shows how to write units 2, 3, 4, and 5.

Definition

An *Introduction to <product>* module is a Learn module that has the following properties:

- Teaches the *evaluate* task; that is, it enables the customer to evaluate whether the product will meet their needs.

- Follows the precise format for module title, unit titles, and unit content.
- Covers a single product (the pattern doesn't apply to role, suite of products, solution, etc.).

Importance

- **Learner:** Standard title tells learners where to start. Standard content means learners can rely on these modules to help them evaluate a product.
 - **Author:** Standard format helps the author determine which content to include.
-

Video

The following video summarizes this guideline.

Rule: Use the standard format for the module title

Details

- The module title must be "Introduction to <product>".

Examples

	TITLE
Do	Introduction to Logic Apps
Don't	Evaluate whether Logic Apps is right for your work
Don't	Is Logic Apps right for you?
Don't	Get started with Logic Apps
Don't	Getting started with Logic Apps
Don't	Foundations of Logic Apps
Don't	Logic Apps fundamentals

Rationale

- A standard format provides uniformity and enables learners to quickly identify our modules that teach the *evaluate product* task.
-

Rule: Define *what the product is* in unit 2

Details

- Unit title must be: "**What is <product>?**".
- Provide a definition of the product.

- Discuss the value proposition of the product but avoid marketing content.

Example

UNIT TITLE	WHAT IS AZURE LOGIC APPS?
Definition	Azure Logic Apps is a cloud service that automates the execution of your business processes...
Value	Let you connect to hundreds of external services...
Value	Extensible model means you can connect to non-standard services...
Value	Graphical tool to build apps with no coding required...

Rationale

- Requiring a separate unit for the product definition ensures that authors include the definition in every *Introduction to <product>* module.
 - Placing this unit at the beginning should match customer's expectations. Someone reading an *Introduction to <product>* module likely expects a product definition to be the first thing they read.
-

Rule: Describe *how the product works* in unit 3

Details

- Unit title must be: "How <product> work(s)". If your product name doesn't fit syntactically with the *How it works* format, you may use a variation that preserves the spirit of the rule.
- Provide a high-level explanation of the parts of the product.
- Explain how the parts work together.
- Show how the product would work when solving the scenario in the module introduction.

UNIT TITLE	HOW LOGIC APPS WORKS
Part	Define connectors
Part	Define triggers, actions, and control actions
Part	Describe the requirements to create a custom connector
Part	Define the Logic Apps Designer
Part	Define the Workflow Definition Language
Work together	App must start with a trigger, followed by one or more actions and control actions...
Work together	Trigger launches your app when conditions are satisfied and passes data to the first action...

UNIT TITLE	HOW LOGIC APPS WORKS
Work together	Actions have inputs and outputs; output data flows to the next action...
Solve the scenario	Show the trigger, actions, and control action used to solve the social-media monitor scenario...

Rationale

- Each customer's situation will be unique. The scenarios and examples we use won't match their work exactly. Knowing how a product works lets our customers accurately estimate the effort involved in adapting the product to their situation. This knowledge will help them evaluate if the product will meet their specific needs.

Rule: Describe *when to use* the product in unit 4

Details

- Unit title must be: "**When to use <product>**". If your product name doesn't fit syntactically with the *When to use...* format, you may use a variation that preserves the spirit of the rule.
- Describe the criteria customers should use when deciding whether the product will meet their needs. Include both strengths and weaknesses.

Example

UNIT TITLE	WHEN TO USE AZURE LOGIC APPS
Criterion	System integration
Criterion	Scalable performance
Criterion	Complex conditional logic
Criterion	Access to nonstandard services

Rationale

- Knowledge of the product's strengths, weaknesses, and the key decision criteria helps the learner evaluate the product against their requirements.

TIP

It can be helpful to provide a flowchart that guides the learner through the criteria and helps them decide *when* and *when not* to use the product.

TIP

You can embed some hands-on interactivity within this unit. The goal must be to deepen the learner's understanding of *when* to use the product. Do not teach *how* to use the product. For a technical module, this could involve asking the learner to run a pre-built application and examine the results.

Rule: Use a knowledge check to validate learning

Details

- Use a knowledge check to add interactivity and validate learning.

TIP

The learning objectives for an *Introduction to <product>* module are *When to use...* and *How it works....* Focus your questions on validating that the learner has met those objectives. Include scenario-based questions; for example, you could present a few scenarios and ask the learner to tell you how the product could be applied in that situation or if the product isn't appropriate to the task.

Examples

LOGIC APPS	EXAMPLE	RATIONALE
Do	Suppose you work for a chain of athletic clubs. You want to automate your lesson sign ups and notifications. You use a proprietary in-house customer management system to store customer phone numbers and email addresses. The customer management system does have a REST API, but Logic Apps doesn't provide a connector for it. How can you use Logic Apps to automate this process?	Provides a use-case where the product is applicable. This question tests the students' comprehension of <i>When to use...</i> the product.
Do	Suppose you work for a brokerage company and need to implement a system for trading of financial instruments. Your system must monitor market conditions, detect changes, and execute trades. You'll need to handle a large volume of transactions and you'll need to do it quickly. The faster you can complete trades, the more of an advantage you will have over your competitors. Which requirement of this system would be difficult for Logic Apps to satisfy?	Provides a use-case where the product is not applicable. This question tests the students' comprehension of <i>When to use...</i> the product.
Do	What differentiates an <i>action</i> from a <i>control action</i> ?	Tests whether the student knows that they can include conditional logic in a Logic App. Knowing that they can include conditional logic helps them decide which types of business problems Logic Apps works for.

Rationale

- A knowledge check provides an efficient way to validate if a student has achieved the goals of the module. It adds interactivity without requiring the student to actually use the product in a hands-on exercise. Including an exercise requires you to include *how to use...* content that is outside the scope for our *Introduction to <product>* modules.

Rubric

Use this rubric to evaluate your work. For each row, find the column that most-closely matches your content and note the number of points in the header of that column. Add up the points earned from the rows. The goal is a score of 14 or higher.

POINTS	3	2	1
Module title is <i>Introduction to <product></i>	Fully e.g. <i>Introduction to Azure Logic Apps</i>	N/A	Not at all e.g. <i>Getting started with Azure Logic Apps</i>
Unit 2 title is <i>What is <product>?</i>	Fully e.g. <i>What is Logic Apps?</i>	N/A	Not at all e.g. <i>Logic Apps defined</i>
Unit 3 title is <i>How <product> works</i>	Fully e.g. <i>How Logic Apps work</i>	N/A	Not at all e.g. <i>Behind the scenes with Logic Apps</i>
Unit 4 title is <i>When to use <product></i>	Fully e.g. <i>When to use Logic Apps</i>	N/A	Not at all e.g. <i>Logic Apps applications</i>
Define <i>what the product is</i> in unit 2	Fully e.g. <i>Includes definition of the product and at least one aspect of the value proposition</i>	Mostly e.g. <i>Includes only a definition of the product with no discussion of value</i>	Not at all e.g. <i>Does not clearly define the product or reads like a marketing pitch</i>
Describe <i>how the product works</i> in unit 3	Fully e.g. <i>Includes definitions for key parts, explains how the parts work together, shows how the parts solve the scenario</i>	Mostly e.g. <i>Include two of the three components: parts, work together, solve the scenario</i>	Not at all e.g. <i>Include zero or one of the three components: parts, work together, solve the scenario</i>
Describe <i>when to use the product</i> in unit 4	Fully e.g. <i>Includes criteria and a short discussion of each</i>	Mostly e.g. <i>Lists criteria but doesn't include discussion to help the learner apply them</i>	Not at all e.g. <i>Seems like a marketing list of features instead of criteria and discussion</i>
Unit 5 is a knowledge check	Yes	N/A	No

How to structure "Choose ..." modules

1/14/2022 • 7 minutes to read

In this article, you will:

- Write titles in the standard formats
- Structure choose modules using scenarios
- Structure units in the standard format

Overview

Choose modules help our customers make an educated decision about the best product to solve their problem. They guide customers at the start of their projects, which means they can have a large impact on the success of those projects. These modules have precisely defined titles and structure to help ensure they're effective. Here's the basic organization:

UNIT	CONTENT
1	Introduction
2	Product options
3	Decision criteria
4+	Analyze a scenario and map it to a product option
N-1	Knowledge check
N	Summary

This article shows how to write each of these unit types.

Definition

- A *Choose...* module guides the learner through a decision-making process.

Importance

- **Learner:** Learners may not know which product will meet their needs. *Choose* modules help them quickly select the best product for further study.
- **Author:** *Choose* modules let authors focus on teaching *when* to use the product. They don't include learning content or interactive exercises that teach *how* to use the product.

Video

The following video summarizes this guideline.

Rule: Use the standard format for the module title

Details

- Write module titles in this format: "Choose the best <*product category*> for your <*purpose*>".

Examples

	TITLE	ANALYSIS
Do	Choose an Azure data storage approach for your relational data	Follows the rule.
Don't	Select an Azure data storage approach for your relational data	Uses Select instead of Choose.
Don't	Choose an Azure data storage approach	Purpose is unspecified; learners might make incorrect assumptions and be disappointed.
Don't	Choose between Azure SQL, PostgreSQL, CosmosDB, and on-premises SQL Server for your relational data	Uses a list of products rather than a product category. The list of products can go in the <code>summary</code> if needed.

Rationale

- A standard format provides uniformity. It enables learners to quickly identify a "Choose" module.

Rule: Provide a broad scenario in unit 1

Details

- Provide a broad scenario in the introduction. Make it large enough that it includes one subtask that will be solved by each product option.

Example

CATEGORY	DESCRIPTION	TYPE OF DATA	PRODUCT
Module title	Choose a data storage option in Azure for your business data	-	-
Scenario	Suppose you work in a hospital and need to provide cloud storage solutions for each department.	-	-
Subtask	Store patient billing data	Structured data	SQL Server
Subtask	Store product and inventory data for retail gift shop	Semi-structured data	CosmosDB

CATEGORY	DESCRIPTION	TYPE OF DATA	PRODUCT
Subtask	Store X-ray images	Unstructured data	Blob storage

Rationale

- A unified, overarching scenario reduces the amount of setup required for each unit. Separate scenarios would require the learner to spend time studying a new scenario in each unit.
-

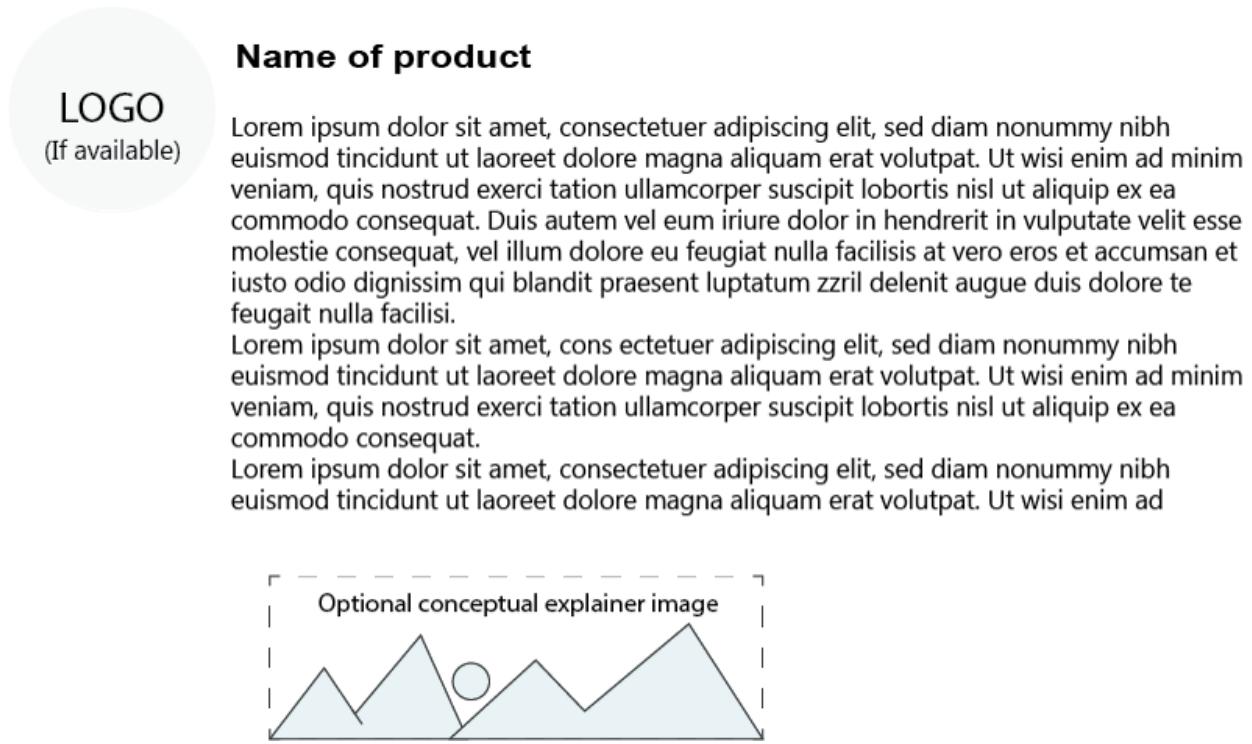
Rule: Identify the product options in unit 2

Details

- Unit title must be: "**Identify the product options**"
- For each product option, provide a 150-200 word overview, the product logo (optional), and a conceptual-explainer image (optional). The goal is to introduce what each product *is* and what it can *do* without describing *how* to use it.

Example

The following mockup shows the scale of the content in this unit for each product option. If your module helped the learner choose among four product options, you would repeat this pattern four times within this unit.



Rationale

- This unit is targeted at learners who are unfamiliar with the products. It gives a quick overview of each option the module will cover.
-

Rule: Analyze the decision criteria in unit 3

Details

- Unit title must be: "**Analyze the decision criteria**"
- State the criteria that experts use to make an educated decision among the product options. The format is unconstrained (prose, list, etc.).

- Tell learners how to analyze their problem to reveal these criteria. The format is unconstrained (prose, list of questions, flowchart, etc.).

Example

Module title: Choose a data storage option in Azure

CRITERIA FOR DECISION-MAKING PROCESS
Data classification
Lookup style
Scale
Performance

QUESTIONS TO HELP ANALYZE THE PROBLEM
What type of data is it?
Will you be doing simple lookups using an ID?
Do you need to query the database for one or more fields?
How many create, update, and delete operations do you expect?
Do you need to run complex analytical queries?
How quickly do these operations need to complete?

Rationale

- The criteria needed to make the best choice is essential to the decision-making process.

Rule: Analyze one subtask in units 4+

Details

- In units 4+, analyze one subtask and show how it's the best choice for one of the product options.
- Use the following unit structure:
 1. **Title format:** Use <product> to <solve problem>
 2. **Introduction:** Statement of the subtask using a scenario based on unit 1.
 3. **Problem analysis:** Analyze the subtask using criteria from unit 3.
 4. **Analysis of the criteria:** Explain how it leads to the product best suited for the problem.
 5. **Analysis of why other option(s) are sub optimal:** Explain why the other products aren't the best choice for the problem.

Examples

UNIT	UNIT TITLE	SCENARIO	ANALYSIS OF THE CRITERIA	ANALYSIS OF WHY OTHER OPTION(S) ARE SUBOPTIMAL
4	Use Blob Storage for unstructured data	Require storage solution for all X-rays, which can be searched by patient ID	unstructured data, lookup with ID, not querying multiple fields, etc. leads to Blob Storage	Not the right type of data for a structured database, doesn't require CosmosDB capabilities...leads to Azure SQL
5	Use AzureSQL for structured data	Require storage solution billing, which can be queried from multiple fields	structured data, not using IDs for lookup, querying from multiple fields, etc. Leads to AzureSQL	It's structured data so not right for Blob storage, you have standard fields for every record, which makes CosmosDB unnecessary
6	Use Cosmos DB for semi-structured data	Require retail storage solution for gift shop	semi-structured data, not using IDs, querying from multiple fields, etc. leads to CosmosDB	It's semi-structured so not right for Blob storage, diverse product lines with differing field needs makes it inappropriate for SQL

Rationale

- Keeps the content focused on the decision-making process.
- Provides the learner with intuition of the type of situation for which a given product is most appropriate.
- This format ensures that both the customer problem and the matching product are covered in the unit. This style helps both customers that are new to the product and customers that already know a bit about it.

NOTE

Consider providing a video in which an expert analyzes each scenario.

Rule: Use a knowledge check for validation

Details

- Use scenario-based knowledge checks to add interactivity and validate learning. Don't use exercises.

Examples

	KNOWLEDGE CHECK	ANALYSIS
Do	Suppose you have an existing application running locally on your own server. You need higher capacity but prefer to move to Azure instead of buying upgraded on-premises hardware. Which compute option would likely give you the quickest route to getting your application running in Azure?	Follows the rule

	KNOWLEDGE CHECK	ANALYSIS
Don't	Which is the best definition for a "container"?	Doesn't validate if a student can choose the correct solution

Rationale

- Scenario-based knowledge check questions provide an efficient way to validate if a student has achieved the goal of the module.
- Exercises aren't appropriate. *Choose* modules teach the *decision*, they don't teach how to *use* the product.

Rubric

Use this rubric to evaluate your work. For each row, find the column that most-closely matches your content and note the number of points in the header of that column. Add up the points earned from the rows. The goal is a score of 14 or higher.

POINTS	EXAMPLE	3	2	1
Format module title as "Choose the best <product category> for your <purpose>"	E.g. <i>Choose an Azure data storage approach for your relational data</i>	Fully	Partly <i>Either module not named with the word Choose, or product category or purpose are missing</i>	Not at all e.g. <i>more than two elements missing Choose an Azure data storage approach</i>
Provide a broad scenario in unit 1	E.g. <i>Suppose you work in a hospital and need to provide cloud storage solutions for each department.</i>	Fully e.g. <i>Scenario can be broken into smaller scenarios appropriate for each product choice</i>	Partly e.g. <i>Scenario provided but is too focused and can't be broken down</i>	Not at all e.g. <i>no scenario provided</i>
Identify the product options in unit 2	E.g. Title formatted as: " Identify the product options ", 100-200 word descriptions of each product	Fully, e.g. <i>title formatted correctly and descriptions of each product included</i>	Partly e.g. <i>title formatted correctly but lacks enough information in descriptions</i>	Not at all e.g. <i>Incorrectly formatted title, no descriptions</i>
Analyze the decision criteria in unit 3	E.g. <i>Provide learners with the questions they should ask about their project to make the right choice</i>	Fully e.g. <i>All decision criteria included</i>	Partly e.g. <i>some decision criteria included</i>	Not at all e.g. <i>decision criteria not included</i>
Analyze one subtask in units 4+	E.g. <i>Each unit analyzes a scenario and chooses a product based on the decision criteria laid out in unit 3</i>	Fully e.g. <i>scenario and decision criteria included</i>	Partly e.g. <i>decision criteria or scenario included</i>	Not at all e.g. <i>No decision criteria or scenario included</i>

POINTS	EXAMPLE	3	2	1
Use knowledge checks for validation	E.g. <i>Use scenario-based knowledge checks to add interactivity and validate learning.</i> <i>Don't use exercises.</i>	Fully e.g. <i>Scenario-based knowledge check included</i>	Not Applicable	Not at all e.g. <i>Includes exercise and/or doesn't use Scenario-based questions</i>

How to structure a learning content unit

1/14/2022 • 3 minutes to read

In this article, you will:

- Organize learning content in an approved sequence

Overview

When learning content is structured properly it can increase retention and aid in transfer of learning. The goal is to help the learner apply what they've learned to tasks outside the context of the module.

Take a look at the sets of numbers below, which do you think is easier to remember?

1. 8776967786
2. (877) 696-7786

TAKE A LOOK AT THE TEXT BELOW, WHICH DO YOU THINK IS EASIER TO READ?

Microsoft Learn, One Microsoft Way, Redmond, WA 98052

Microsoft Learn
One Microsoft Way
Redmond, WA
98052

In all likelihood you chose the phone number format, and the second format for address. This is because they are both grouped into small *chunks*. Instead of trying to remember 10 random numbers, you only have to remember 3 *chunks* of numbers. Instead of having to parse a long line of text, you have 4 short lines of text to read.

It's a well studied theory that working memory (the memory we use to process information) is limited in capacity. The specific amount of information is debateable, but it's agreed that the number is somewhere between 3-7 chunks of information at a time. A chunk may include more than one piece of information, however the information is logically grouped together.

Presenting too much information at a time can overwhelm a learner. As a course author, it's our responsibility to help students manage working memory so they can retain what we have to teach. We do this through breaking up our content into manageable chunks, just like we do with phone numbers and addresses.

When planning content for a module, the first step is to break the largest chunk (module) into smaller pieces (units). We then have to further break up our learning content units into logical chunks of related information. This article describes how to structure a learning content unit into manageable chunks.

Definition

- A **learning content unit** is a unit that teaches a learner the knowledge needed to solve a portion of the problem presented in the module.

Importance

- **Learner:** Breaking content into chunks makes the topics easier to read and remember.
- **Author:** Creates an organized method of presenting content.

Video

The following video summarizes this guideline.

Rule: Break content into chunks

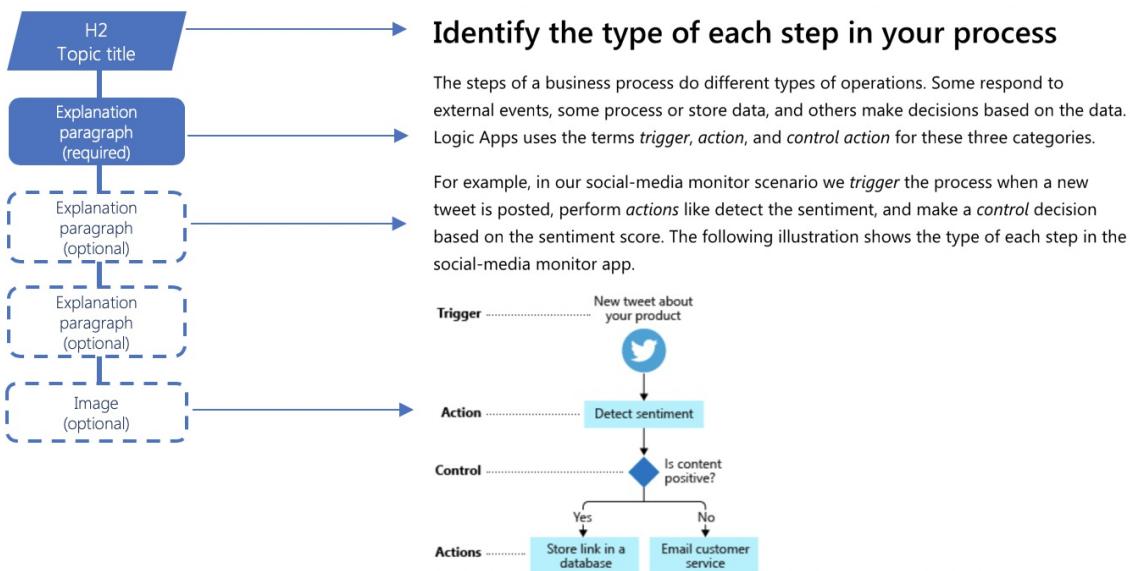
Details

- A *chunk* of Learn content includes:
 - H2 Chunk title
 - 1-3 explanatory paragraphs
 - Optional image

Examples

Standard chunk of content

Chunk from "Route and process data with Logic Apps"



Rationale

- Helps the learner identify the most important concepts in the unit.
- Chunking content breaks up walls of text and increases readability.
- Groups like-content together making it easier to remember.

Rule: Use an approved sequence for a learning content unit

Details

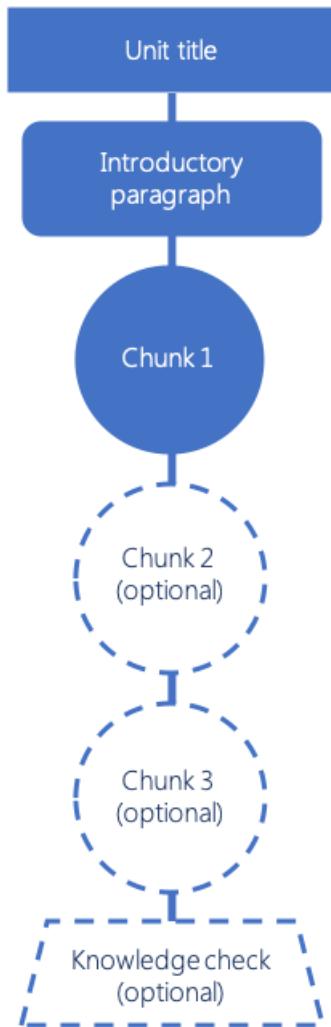
- Must begin with an introductory paragraph that follows the [introduction](#) rules
- Must include at least 1 to 3 chunks of learning content

- Does not include a unit summary
- May or may not include a knowledge check

Example

Approved pattern

Analysis

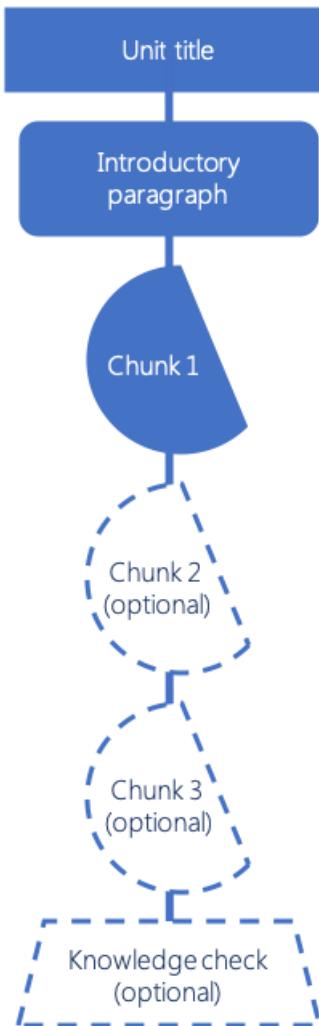


Unit structure includes:

- Introductory paragraph, H2 titles, explanatory paragraphs and images
- Unit structure includes all necessary items and a knowledge check

Rejected pattern

Analysis



Missing or inappropriate elements:

- Introductory paragraph
- H2 title
- H2 titles and contains a summary

Rationale

- A standard format allows the learner to anticipate how all Learn units are organized.

IMPORTANT

Depending on the lengths of your chunks, your unit(s) *may* have more than 3 chunks. The sequence above provides a unit sequence that assumes 5-10 minutes of learning content.

Rubric

Use this rubric to evaluate your titles. For each row, find the column that most-closely matches your title and note the number of points in the header of that column. Sum the points earned from each row. The goal is a score of 12 or higher.

POINTS	3	2	1
Unit sequence follows the content and ordering rules	Fully e.g. <i>Unit begins with introduction, and has a mix of learning content and opportunities for practice</i>	-	Not at all e.g. <i>Module is missing an introduction or summary and/or does not have a mix of learning content and practice</i>
Use the standard unit-introduction format	Fully e.g. <i>Unit begins with an introduction that follows the introduction guidelines.</i>	-	Not at all e.g. <i>No introduction included or includes an introduction that doesn't meet the guidelines</i>
Chunks include an H2 title	Fully e.g. <i>Content is broken into topics with H2 Titles</i>	Mostly e.g. <i>Most topics are titled with an H2</i>	Not at all e.g. <i>No H2 titles included</i>
Chunks include explanatory elements	Fully e.g. <i>Unit chunks include explanatory paragraph(s) and optional images</i>	Mostly e.g. <i>Explanations are included for some but not all chunks</i>	Not at all e.g. <i>No explanations included</i>
No summary included	Fully e.g. <i>No unit summary is included</i>	-	Not at all e.g. <i>Unit summary is included</i>

How to structure exercise units

1/14/2022 • 3 minutes to read

In this article, you will:

- Organize the content of an exercise unit in an approved sequence

Overview

Exercises that are broken into chunks are easier to understand and easier to remember. They also help the learner see larger patterns in a solution instead of getting overwhelmed by the details.

For example, suppose you were writing an exercise to [Create a Windows Server virtual machine in the Azure Portal](#). From the options below, which looks more manageable?

OPTION 1	OPTION 2
<ol style="list-style-type: none">Open a web browser.In the search bar, type `www.azure.com`.Click Sign in.Enter your credentials.Click Portal.On the left navigation bar, click Create a resource.Click Compute.Click Virtual Machine.Select a valid subscription.Select or create a resource group.Enter a virtual machine name.Select a region.Select a virtual machine image.Select a size.Create a username and password.Click Review + create.Open Remote Desktop Connection on your computer.Enter the IP address of your virtual machine.Enter the username you created.Enter the password you created.Click Connect.	<p>Access the Azure Portal</p> <ol style="list-style-type: none">Open a web browser.In the search bar, type `www.azure.com`.Click Sign in.Enter your credentials.Click Portal. <p>Create a virtual machine using a template</p> <ol style="list-style-type: none">On the left navigation bar, click Create a resource.Click Compute.Click Virtual Machine. <p>Configure and deploy your virtual machine</p> <ol style="list-style-type: none">Select a valid subscription.Select or create a resource group.Enter a virtual machine name.Select a region.Select a virtual machine image.Select a size.Create a usernames and password.Click Review + create. <p>Verify that your virtual machine is properly running</p> <ol style="list-style-type: none">Open Remote Desktop Connection on your computer.Enter the IP address of your virtual machine.Enter the username you created.Enter the password you created.Click Connect.

As you can see, the second option has all related steps chunked into smaller sub-tasks. Instead of a 21-step task, you have four sub-tasks each with only five to eight steps. This not only makes the exercise feel more manageable, but it's easier to remember a four-step procedure and it helps the learner to focus on the larger pattern happening in the solution rather than the minutia of the steps.

This article describes how to structure an exercise unit into manageable sub-tasks.

Definition

- An **exercise unit** is a unit that contains procedural steps to help the learner apply what they've learned in a learning-content unit.

Importance

- **Learner:** Chunking an exercise into sub-tasks makes the content more manageable and easier to remember.
 - **Author:** The standard format simplifies exercise planning and helps the author build effective training content.
-

Video

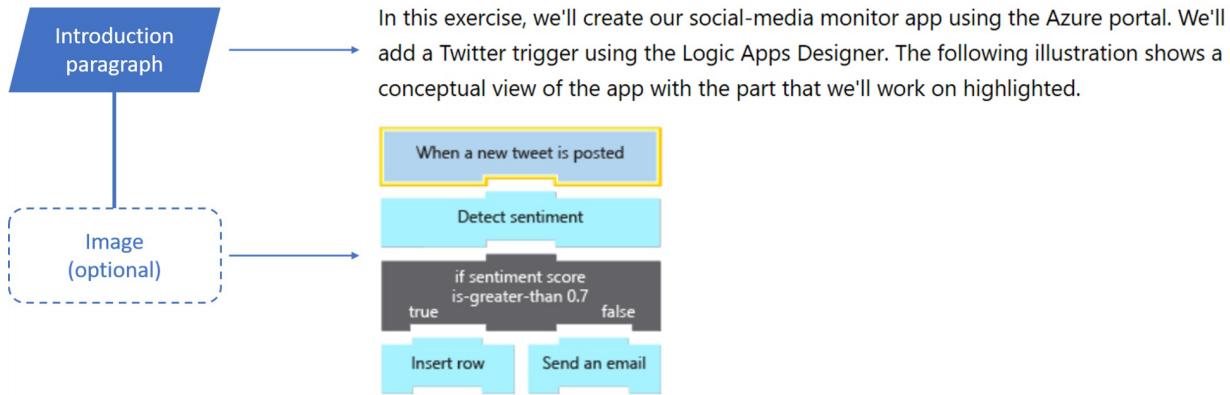
The following video summarizes this guideline.

Rule: Provide an introduction

Details

- An introduction includes:
 - A paragraph that orients the learner to the task they will complete
 - Optional image

Example



Rationale

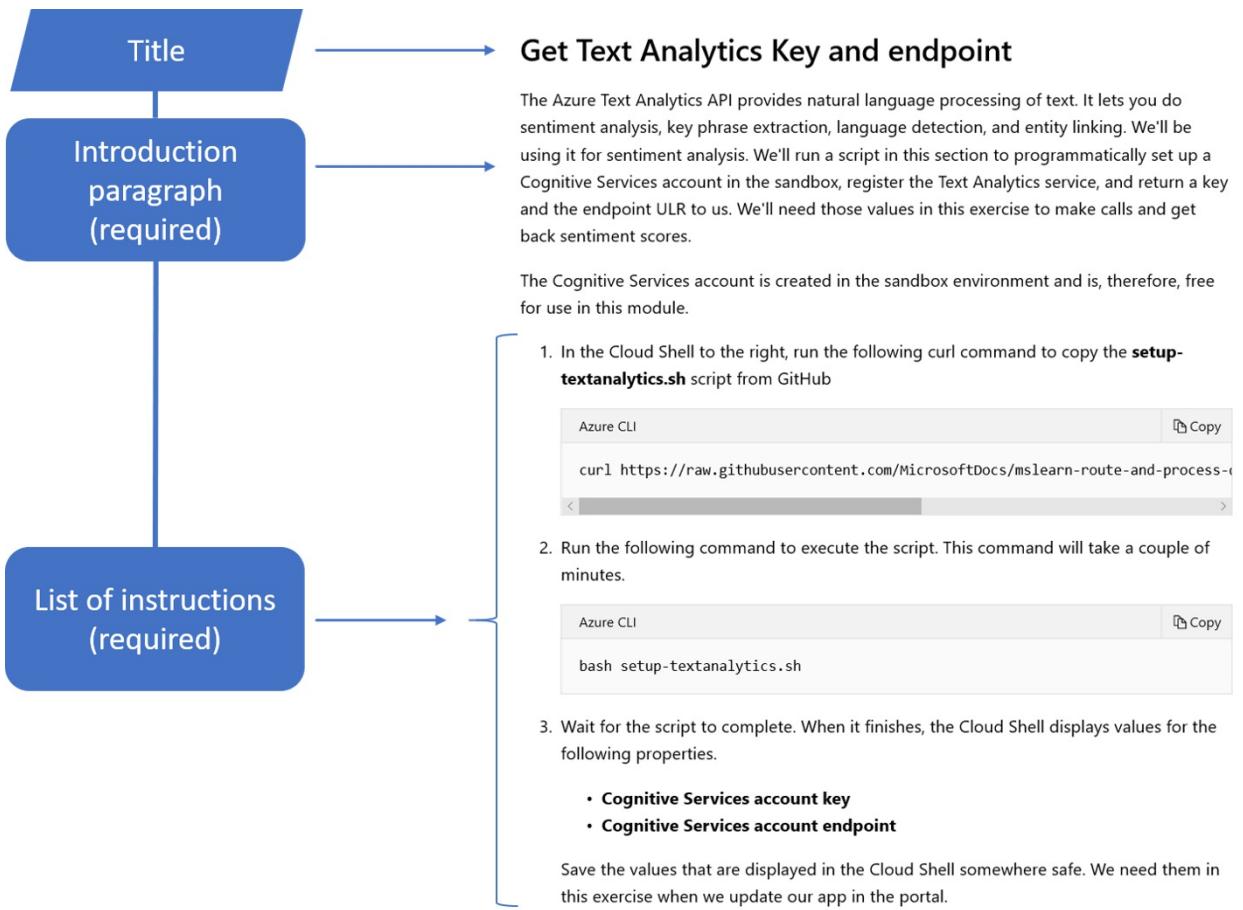
- Helps the learner determine the scope of the exercise.
-

Rule: Chunk exercise into completable sub-tasks

Details

- A sub-task includes:
 - Title
 - A brief paragraph that describes the goal of the sub-task
 - Numeric list of instructions

Example



Rationale

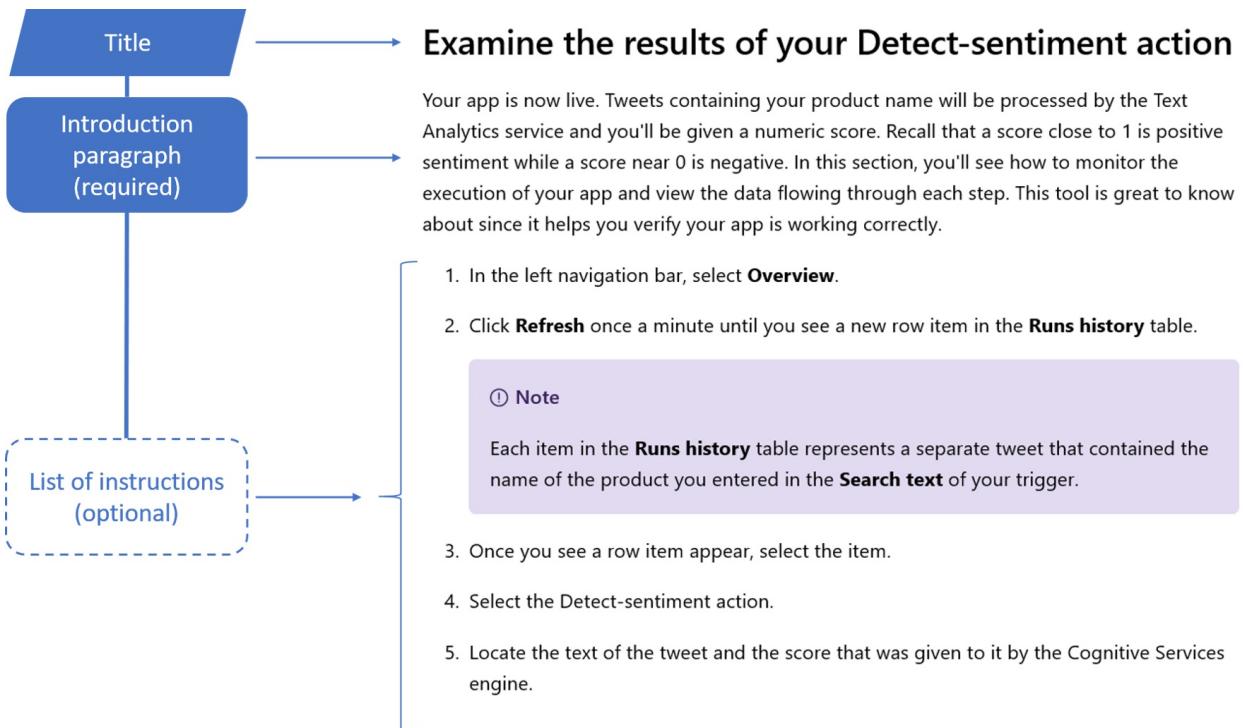
- Provides the learner with information to successfully complete the exercise.
- Presenting too much information at once can overwhelm the learner. Smaller sub-tasks can help the learner process the information more efficiently.
- Achieving success earlier and more frequently helps motivate the learner to finish the exercise.

Rule: Include a validation sub-task

Details

- A Validation sub-task includes:
 - Title
 - A brief paragraph that describes how they will validate if they've been successful
 - Optional numeric list of instructions (*when the learner needs to perform multiple steps to verify if they were successful*)

Example



Rationale

- Helps the learner to evaluate if they completed the exercise correctly.

Rubric

Use this rubric to evaluate your work. For each row, find the column that most-closely matches your content and note the number of points in the header of that column. Repeat the learning-unit row for each learning unit. Repeat the exercise-unit row for each exercise unit. Add up the points earned from the rows. The goal is a score of 15 or higher.

POINTS	3	2	1
Contains unit introduction	Fully e.g. <i>Unit begins with an introduction paragraph</i>	-	Not at all e.g. <i>Unit does not begin with an introduction paragraph</i>
Exercise split into sub-tasks	Fully e.g. <i>There are multiple sub-tasks each containing their own set of instructions</i>	-	Not at all e.g. <i>Only one task with a single set of instructions</i>
Sub-tasks contain a title	Fully e.g. <i>Each sub-task has a title</i>	Mostly e.g. <i>80% of sub-tasks have a title</i>	Not at all e.g. <i>Less than 80% of the sub-tasks have a title</i>
Sub-tasks contain an introduction paragraph	Fully e.g. <i>Each sub-task has an introduction paragraph</i>	Mostly e.g. <i>80% sub-tasks have an introduction paragraph</i>	Not at all e.g. <i>Less than 80% the sub-tasks have an introduction paragraph</i>
Includes a validation sub-task	Fully e.g. <i>Includes instructions to validate success at the end of the unit</i>	-	Not at all e.g. <i>No instructions to validate success included</i>

Detailed guidelines quick reference

1/14/2022 • 7 minutes to read

This article contains a condensed version of all the detailed guidelines for Learn modules.

Titles

- Write titles in *present-tense-verb > task > product* format
- Be specific
- Write from the learner's viewpoint

For more information, see the [Detailed guidance for writing titles](#) article.

Examples

TYPE OF MODULE	EXAMPLE
Introduction to...	Introduction to Azure Logic Apps
Choose...	Choose an Azure data storage approach for your relational data
Standard	Create an Azure Storage account by using the portal

Introductory summaries

- Include key technologies
- Include key sub-tasks
- Be brief (20-35 words)

For more information, see the [Detailed guidance for writing introductory summaries](#) article.

Example

Define a workflow using built-in triggers, actions, and conditional operators. Your workflow will run automatically when data is available, process the data in the cloud, record results in a database, and send email notifications.

Learning objectives

- Be specific and measurable
- Be achievable and time bound
- Use a verb that matches the level of Bloom's you want to measure
- Write from the learner's point of view
- Written to complete this sentence: Upon completion of this module/unit, the student will be able to:

For more information, see the [Detailed guidance about writing learning objectives](#) article.

Examples

LEVELS OF BLOOM'S	EXAMPLE OBJECTIVE
Create	Design a workflow for a Logic app
Evaluate	Choose the correct API key for the task
Analyze	Differentiate a 2D array from a 3D array
Apply	Launch a Logic App in response to an external event
Understand	Explain the use of a trigger in a Logic App
Remember	Define a control action

Prerequisites

- Don't reference Microsoft Learn modules or Docs content
- Be concrete, use clear and unambiguous language
- Use bulleted lists
- Quantify experience using levels
- Include software setup requirements

For more information, see the [Detailed guidance about writing prerequisites](#) article.

- Basic knowledge of the Azure Logic Apps concepts of connector, trigger, and action
- Familiarity with programming concepts like conditional statements and loops
- Experience creating and managing resources using the Azure portal at the beginner level

Scenarios

- Tell a specific story
- Make it anonymous
- Avoid gendered pronouns
- Keep it simple
- Match the problem
- Use one scenario per module

For more information, see the [Detailed guidance for writing scenarios](#) article.

Examples

TITLE	EXAMPLE SCENARIO
Module: "Route and process data automatically with Azure Logic Apps"	"Suppose you work at an athletic shoe company as the IT expert on the marketing team. You're launching a new basketball shoe created by a celebrity athlete. Previous launches resulted in a social media frenzy as people shared stories and photos of themselves wearing their new shoes. The marketing team wants to monitor hundreds of social media sites, blogs, and websites to determine consumer reaction. You expect there will be more data than you can process manually, and you don't have the server resources to build an automated solution in-house."

TITLE	EXAMPLE SCENARIO
Unit 1: "Launch your Logic App automatically when new data is available."	"Recall that your goal in the shoe company scenario is to monitor social media for content about your new product launch. A key part of your plan is to scan Twitter for tweets that mention your product. For each matching tweet, you need to extract the text to determine if it's positive or negative."

Module introduction format

- Write module introductions in *topic > scenario > TOC > goal* format

For more information, see the [Detailed guidance for writing module introductions](#) articles.

Examples

EXAMPLE	ROUTE AND PROCESS DATA USING AZURE LOGIC APPS
Topic sentence(s)	Azure Logic Apps let you automate your business processes by running your workflows in the cloud.
Scenario	Suppose you work at an athletic shoe company as the IT expert on the marketing team. You're launching a new basketball shoe created by a celebrity athlete. Previous launches resulted in a social media frenzy as people shared stories and photos of themselves wearing their new shoes. The marketing team wants to monitor hundreds of social media sites, blogs, and websites to determine consumer reaction. You expect there will be more data than you can process manually, and you don't have the server resources to build an automated solution in-house. You need a cloud-hosted system that integrates monitoring, text analytics, database storage, and email notifications.
Prose table-of-contents	In this module, you'll create a business workflow using Azure Logic Apps to automate the processing of news articles. Your workflow will trigger when a new article is available. It will use cloud services to determine if the article is positive or negative and branch based on the results. After processing, it will persist the data in a cloud-hosted database.
Goal (that is, terminal learning objective)	By the end of this module, you'll have learned how to create workflows which route and process data using Azure Logic Apps and its built-in connectors.

Learning-unit introduction format

- Write learning-unit introductions in *topic > scenario > TOC* format

For more information, see the [Detailed guidance for writing module summary units](#) article.

Examples

EXAMPLE	DECIDE HOW MANY STORAGE ACCOUNTS YOU NEED
Topic sentence	Organizations often have multiple storage accounts to let them implement different sets of requirements.

EXAMPLE	DECIDE HOW MANY STORAGE ACCOUNTS YOU NEED
Scenario sub-task	In the chocolate-manufacturer example, there would be one storage account for the private business data and one for the consumer-facing files.
Prose table-of-contents	Here, you'll learn the policy factors that are controlled by a storage account so you can decide how many accounts you need.

Exercise-unit introduction format

- Write exercise-unit introductions in *topic > scenario > task* format

For more information, see the [Detailed guidance for writing unit introductions](#) article.

Examples

EXAMPLE	CREATE A STORAGE ACCOUNT USING THE AZURE PORTAL
Topic sentence	A storage account represents a collection of settings that implement a business policy.
Scenario sub-task	Recall that in the chocolate-manufacturer example, there would be a separate storage account for the private business data. There were two key requirements for this account: geographically redundant storage because the data is business-critical and at least one location close to the main factory.
Task performed in the exercise	Here, you'll create a storage account with settings appropriate to hold this mission-critical business data.

Knowledge checks

- Write answers that are all about the same length
- Provide three plausible answers per question to choose from
- Don't write True/False questions
- Don't include "All of the above" or "None of the above" as answer choices
- Write questions as a complete sentence ending with a question mark (*closed* format)
- Avoid using the words "Not" and "Except" in questions
- If the answers are in numeric value, list them in sorted order
- Use third person, not "you"
- Provide a meaningful explanation for both correct and incorrect answers
- Align questions with learning objectives

For more information, see the [Detailed guidance about writing knowledge checks](#) article

Example

```
- content: "What needs to be installed to execute Azure PowerShell cmdlets locally?"  
  
choices:  
  
- content: "The Azure cloud shell"  
  
isCorrect: false  
  
explanation: "The cloud shell does execute Azure cmdlets; however, it's used in a browser."  
  
- content: "The base PowerShell product and the AzureRM module"  
  
isCorrect: true  
  
expalanation: "Both the base PowerShell product and the AzureRM module. The base product provides the shell itself, a few core commands, and programming constructs like loops, variables, etc. The AzureRM modules adds the cmdlets needed to work with Azure resources."  
  
- content: "The Azure CLI and Azure PowerShell"  
  
isCorrect: false  
  
explanation: "The Azure CLI is an alternate command-line and scripting tool. It is not needed if you are going to use Azure PowerShell."
```

Module summary

- Restate the scenario problem(s)
- Show how you solved the scenario problem(s)
- Describe the experience without the product
- Describe the business impact

For more information, see the [Detailed guidance about writing module summary units](#) article.

Example

Module: Route and process data automatically using Logic Apps

Our marketing team needed to gauge customer response to our new shoe. We wanted to monitor social media, determine customer reaction, and route the data to either a database or customer service based on sentiment.

Azure Logic Apps let us automate the process. Each step of our workflow was mapped to one of the built-in components. We used the Twitter trigger to detect mentions of our product and launch our app. The Azure Cognitive Services action let us analyze whether the tweets were positive or negative. A control action helped us decide where to route the tweet based on sentiment. Finally, we used actions to insert a row in SQL Server and send an Outlook email.

Imagine how much work it would be to build this app from scratch. We'd have to write code to access each of the services' APIs. We'd need a polling infrastructure to monitor Twitter and trigger our app when new tweets were available. Once all the code was ready, we'd need servers to host the app.

Logic Apps made it easy. The standard connectors did the hard work of integrating systems that were never designed to work together. Building the app took us less than an hour and we didn't have to write any code or set up any servers. We can now analyze every tweet about our product and our business analytics can use the data to shape future products.

How to write titles for learning paths, modules, and units

1/14/2022 • 5 minutes to read

In this article, you will:

- Create titles that help the learner decide whether to take the training
 - Structure titles in our task-oriented format
-

Overview

Learning Paths, Modules, and Units must have a task-oriented title. The title tells the learner what they will be able to do after taking the training. The same rules for writing titles apply to all three content types.

Definition

- A ***title*** is a concise statement that captures the terminal learning objective of the content.

Importance

- **Learner:** An effective title helps the learner evaluate whether the content will meet their needs. We don't want learners to begin a module and later realize it isn't what they expected and won't help them.
 - **Author:** A task-based title guides the author on what to include in the content. Authors should scope the content so it includes only information relevant for that task.
 - **Microsoft Learn:** The format helps us distinguish similar modules. Suppose there are two modules that cover the same basic task but for different audiences. It's likely that the primary learning goal will be slightly different in the two cases. Our title format incorporates this primary learning objective. The format is detailed enough that following the standards generally results in unique titles for the two modules. This allows content teams to publish the content that works for both parties – while ensuring the learner is not misled and has the best experience on the platform.
-

Video

The following video summarizes this guideline.

Rule: Use our task-oriented format

Details

- Write titles in ***present-tense-verb > task > product*** format.

Examples

	TITLE	ANALYSIS
Do	Automate Azure tasks by using PowerShell scripts	Follows the rule
Don't	Automating Azure tasks by using PowerShell scripts	Verb not in present tense
Don't	Use PowerShell scripts to automate Azure tasks	Product should not be listed first
Do	Create an Azure Storage account by using the portal	Follows the rule
Don't	Create an Azure Storage account	Does not include the product
Don't	Azure Storage Account options	Does not include a verb or task

Rationale

- A standard format gives all our titles the same structure. This makes it easier for learners to understand new titles as they encounter them.
 - Learners typically seek training when they have a specific problem to solve. Our titles start with a verb+task so the problem our content will solve is listed first. This helps the learner quickly decide whether the content is right for them.
-

Rule: Be specific

Details

- Make titles specific: use precise language and include brief details of what the learner will do and how they will do it.

Examples

	TITLE	ANALYSIS
Do	Automate Azure blob admin tasks by using PowerShell scripts	Follows the rule
Don't	Automate Azure tasks by using PowerShell scripts	Learner cannot determine if the tasks they need will be covered
Don't	Automate Azure tasks by using scripts	Learner cannot determine the tasks or the product
Do	Remove grass stains from clothes by using non-toxic household ingredients	Follows the rule
Don't	Remove stains by using non-toxic household ingredients	Learner cannot determine type of stain nor whether it applies to clothing, floors, carpet, etc.

	TITLE	ANALYSIS
Don't	Keep your clothes looking like new	Learner cannot determine whether this is about stain removal, ironing, storage, etc.

Rationale

- Make titles specific so they align with adult learning theory principles. Specific titles tend to be problem-centered which helps the learner see how the content is relevant to them.
 - Specific language tells the learner exactly what is in the content. This respects their time by helping them quickly choose the right module.
 - General language increases the risk of disappointment. The learner may assume the content will solve their problem, begin to take the training, and then be unhappy when they find that it doesn't meet their needs.
 - Specificity helps the author avoid bloating the content with extraneous detail. For example, consider a vague title like "Introduction to landscaping". As the author, should you include content on varieties of grass, types of trees, soil drainage, and sprinkler systems? The title gives you no guidance on whether to include these; in fact, the title implies 'yes' since these are all related to landscaping. By contrast, a title of "Choose the right grass for your climate" would make it obvious that you should omit the content on tree types.
-

Rule: Write from the learner's viewpoint

Details

- Write titles from the learner's view of the problem, not the author's view of the product.

Examples

	TITLE	ANALYSIS
Do	Route and process data by using Azure Logic Apps	Follows the rule
Don't	Orchestrate a serverless workflow by using Azure Logic Apps	Emphasizes the product, not the problem it solves; the learner may not know what a <i>workflow</i> is nor what <i>serverless</i> means
Do	Enable fast access to your data anywhere in the world by using Azure Cosmos DB	Follows the rule
Don't	Distribute your data globally by using Azure Cosmos DB	Learners may not know this includes synchronization nor that global distribution yields low latency

Rationale

- Titles written in terms of the problem help learners find the right content. Our training is often consumed just-in-time; that is, learners have a specific problem in mind that they need to solve. Learners often use search terms based on their *problem* and not a vendor's *product*. The learner may not know which product will help them. For example, suppose a learner needs to enable fast access to data for customers anywhere in

the world. What terms might they search for? Will they know to search for Cosmos DB? Will they know global distribution with replication is a good strategy? Or will they search for terms like "low latency" and "global fast access"?

Rubric

Use this rubric to evaluate your titles. For each row, find the column that most-closely matches your title and note the number of points in the header of that column. Sum the points earned from each row. The goal is a score of 6 or higher.

POINTS	DESCRIPTION	4	3	2	1
Content	Title conveys problem it will solve	Fully e.g. <i>Route and process data by using Azure Logic Apps</i>	Mostly e.g. <i>Create workflows by using Azure Logic Apps</i>	Partly e.g. <i>Orchestrate a serverless workflow by using Azure Logic Apps</i>	Not at all e.g. <i>Introduction to Azure Logic Apps</i>
Format	Contains all elements and in the right order	Fully e.g. <i>verb>task>product</i>	Mostly e.g. <i>product>verb>task</i>	Partly e.g. <i>verb>task</i> or <i>task>product</i>	Missing two or more elements

How to write introductory summaries for modules and learning paths

1/14/2022 • 3 minutes to read

In this article, you will:

- Write a summary that expands on the task and technology in the title
- Avoid extraneous content in your summary

Overview

Our *title*, *summary*, and *introduction* help customers decide if the content will be useful to them. They are typically read in that order: *title* > *summary* > *introduction* with each element giving more information than the last. This article discusses the *summary*.

Definition

The *summary* tells the learner what they will do and how they will do it; it expands on the *title* but doesn't duplicate it.

Importance

The summary helps the learners quickly decide whether the content meets their needs. This values the learner's time.

Video

The following video summarizes this guideline.

Rule: Include key technologies

Details

- Include the parts of the enabling technology that are covered in the content. To keep it brief, omit less-important parts or group a few low-level parts into one higher-level element.

Example

PART	DESCRIPTION
Title	Route and process data automatically using Azure Logic Apps
Technology	Azure Logic Apps

PART	DESCRIPTION
Part	Built-in triggers (<i>include</i>)
Part	Built-in actions (<i>include</i>)
Part	Built-in connectors (<i>do not include, including triggers and actions makes this unnecessary</i>)
Part	Conditional logic (<i>include</i>)
Part	Workflow designer (<i>do not include, this is interesting but would make the summary too long</i>)
Summary	Define a workflow using built-in triggers, actions, and conditional operators . Your workflow will run automatically when data is available, process the data in the cloud, record results in a database, and send email notifications.

Rationale

- This part of the summary is *technology-focused* and is useful for learners that are already familiar with the technology. Knowing which aspects of the technology are covered will help them decide whether the content will be useful in their own work.

Rule: Include key sub-tasks

Details

- Include sub-tasks of the content's main task. To keep it brief, omit less-important sub-tasks or group a few low-level sub-tasks into one higher-level element.

Example

SUB-TASK	DESCRIPTION
Title	Route and process data automatically using Azure Logic Apps
Task	Route and process data
Sub-task	Create a Logic App (<i>do not include, too obvious</i>)
Sub-task	Detect external event (<i>include</i>)
Sub-task	Execute actions (<i>include a few examples like database and email</i>)
Sub-task	Execute conditional logic (<i>do not include, too low-level for this section and is covered in the technologies section</i>)

SUB-TASK	DESCRIPTION
Summary	Define a workflow using built-in triggers, actions, and conditional operators. Your workflow will run automatically when data is available , process the data in the cloud, record results in a database, and send email notifications.

Rationale

- This part of the summary is *problem-focused* and is useful for learners that are unfamiliar with the technology. The sub-tasks tell the learner what problems they will learn to solve which helps them decide whether the content will be useful in their own work.
-

Rule: Be brief

Details

- Length should be 20-25 words; up to 35 is acceptable. To keep it brief, avoid teaching, omit motivation, and remove "learn", "in this module", and title words.

Examples

Module title: *Route and process data automatically using Azure Logic Apps*

	SUMMARY	ANALYSIS
Don't	Route and process your data in the cloud using the built-in triggers, actions, and conditional operators....	Repeats the title.
Don't	In this module, you will define a workflow using built-in triggers, actions, and conditional operators....	Includes "In this module".
Don't	Learn how to use Azure Logic Apps to automate your business processes....	Contains "learn".
Don't	Azure Logic Apps are a serverless technology that let you run your business processes in the cloud....	Contains teaching content.
Don't	Businesses often need to automate their processes to reduce costs and eliminate errors....	Contains motivation.

Rationale

- This rule ensures the summary stays short and focused. The summary should include only sub-tasks and key technologies.
-

Rubric

Use this rubric to evaluate your work. For each row, find the column that most-closely matches your content and note the number of points in the header of that column. Add up the points earned from the rows. The goal is a score of 21 or higher.

POINTS	4	3	2	1
Word count	20-25	26-35	Less than 20	More than 35
Repeats title	Not at all	Partly	Mostly	Verbatim
Contains "In this module" or "In this learning path"	No	-	-	Yes
Contains "learn" as a filler word, e.g. "You will learn"	No	-	-	Yes
Number of technology parts included	2-3	4	1	0 or more than 4
Number of sub-tasks included	2-3	4	1	0 or more than 4
Sub-tasks phrased as a task starting with present-tense verb	All are	1 is not	2 are not	3 or more are not

How to write learning objectives

1/14/2022 • 7 minutes to read

In this article, you will:

- Create effective learning objectives
 - Choose the right verbs for your learning objectives
 - Structure learning objectives in the standard format
-

Overview

Microsoft Learn (MSLearn) requires that modules and units contain learning objectives, the purpose of this article is to provide guidance on how to write them in the MSLearn format.

Definition

Learning objectives are **statements that define the expected goal of a course, in terms of demonstrable skills or knowledge that will be acquired by a student as a result of instruction**.

Modules and units require learning objectives. The difference between them is in scope.

- **Module level:** Terminal learning outcomes, objectives you want the student to meet by the end of the Module.
- **Unit level:** Learning objectives which ultimately enable the learner to meet the module level objectives.

Importance

Effective learning objectives accomplish several things:

- **Learner:** Direct learners' attention to the appropriate content and values the learner's time by letting them know exactly what they will learn
 - **Author:** Helps the author to plan and structure the content appropriately and helps to create knowledge checks
-

Video

The following video summarizes this guideline.

Rule: Be specific and measurable

Details

Learning objectives must be specific and measurable.

Good learning objectives align directly with how you plan to measure the learning. In other words, how will you measure what the learner is able to do or accomplish as a result of taking a module or unit?

How you would measure success with the following objectives?:

1. Upon completion of this unit, the learner will be able to **classify** services offered in Azure into one of the three service models
2. Upon completion of this unit, the learner will be able to **describe** the three models of service offered in Azure
3. Upon completion of this unit, the learner will be able to **list** the three models of service offered in Azure

LEARNING OBJECTIVE VERB	METHOD TO MEASURE
...the learner will be able to classify	Matching quiz
...the learner will be able to describe	Choose the correct description
...the learner will be able to list	Select the correct items from a list

Rationale

- Specific actions/verbs help the learner focus on relevant content.
- Measurable actions/verbs are important because they allow the learner to assess if they've met the objective of the module/unit.

Now consider how you would measure success with these seemingly similar objectives:

Upon completion of this unit, the learner will be able to:

1. **understand** the three models of service offered in Azure
2. **learn** the three models of service offered in Azure
3. **work with** the three models of service offered in Azure

LEARNING OBJECTIVE VERB	METHOD TO MEASURE
...the learner will be able to understand	Understand is not measurable
...the learner will be able to learn	Learn is not measurable
...the learner will be able to work with	Work with is not measurable

Rationale

- Non-specific action/verbs which makes it difficult for the learner to focus on what we expect them to learn.
- Ambiguous verbs make it difficult to measure success.
- Ambiguous verbs create an environment where the student can make assumptions about what the module will focus on, and ultimately walk away disappointed if the module doesn't meet that expectation.

Rule: Choose the right verb for your objective

Details

Learning objectives should directly align with how you plan to assess a learner's success.

In order to find the right verb for your learning objective you need to consider how deeply the learner needs to understand the content.

Let's take a look at our three objectives again:

Upon completion of this unit, the learner will be able to:

1. **classify** Azure services into one of the three service models
2. **describe** the three models of service offered in Azure
3. **list** the three models of service offered in Azure

All three examples teach *about* the same topic, but we are asking for differing levels of knowledge.

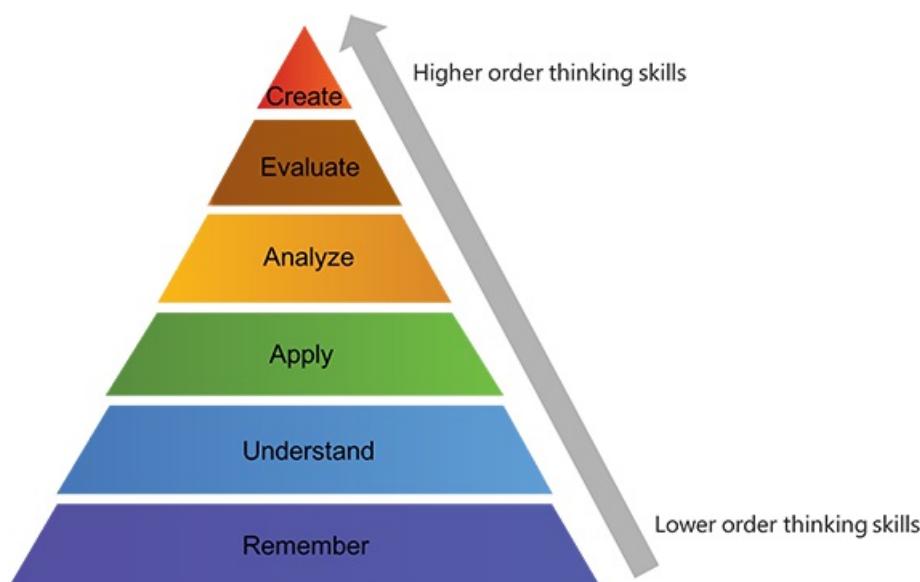
In order to achieve our first objective, **classify** services under different service models, the student would need to be able to:

1. *List* the service models.
2. *Describe* the differences between the service models.

The ability to *classify* information is a *higher level* learning task than listing or describing. A student will be unable to complete a higher level learning objective unless they've been taught or already know the content at the lower levels.

Bloom's Taxonomy

Bloom's Taxonomy is a framework which breaks up learning into levels of thinking (cognitive domains) required to learn. Once you decide which level of learning the student needs to do you can identify the specific and measurable verb. The higher up the chart you go, the more engaging the learning will be.



LEVEL OF LEARNING	DEFINITION	SAMPLE VERBS	SAMPLE ASSESSMENT
Create	Draw from ideas produce new or original work	Build, Change, Compile, Compose, Construct, Create, Design, Develop, Discuss, Elaborate, Estimate, Formulate, Happen, Imagine, Improve, Invent, Makeup, Maximize	Student will create a unique solution to a given problem

LEVEL OF LEARNING	DEFINITION	SAMPLE VERBS	SAMPLE ASSESSMENT
Evaluate	Make judgments about the value of ideas or materials	Assess, Choose, Conclude, Criticize, Decide, Determine, Estimate, Evaluate, Explain, Importance, Influence, Interpret, Judge, Justify, Measure, Prioritize	Have student choose which tools are best based on a scenario
Analyze	Separate a concept into ideas	Appraise, Compare, Contrast, Criticize, Differentiate, Discriminate, Distinguish, Examine, Experiment, Question, Test	Hands-on lab where student has to determine the correct steps or tools to use
Apply	Use a concept in a new situation	Choose, Demonstrate, Dramatize, Employ, Illustrate, Interpret, Operate, Sketch, Solve, Use, Schedule	Guided, hands-on lab
Understand	Comprehend the meaning of content and have the ability to put in one's own words	Classify, Compare, Contrast, Demonstrate, Explain, Extend, Illustrate, Infer, Interpret, Outline, Relate, Rephrase, Show, Summarize, Translate	Select the correct definition
Remember	Remember previously learned information	Define, Identify, Inventory, List, Locate, Name, Recall, Record, Repeat, Restate, State, Underline	Select the correct items from a list

Rationale

- Bloom's Taxonomy is a tool you can use to identify the level of learning you want the learner to achieve. Once identified, you can then choose the appropriate verb for your learning objective.

Rule: Use the standard format

Details

There are three rules to writing learning objectives in the standard format.

- Learning objectives should be written to complete this sentence: **Upon completion of this module/unit, the student will be able to:**
- Objectives should follow this pattern: **audience > behavior > condition > (standard)**.
- Verb should always be in the present tense, not present progressive.

	DEFINITION	EXAMPLE
Audience	Who the instruction is written for	The learner
Behavior	The action or task the audience will do (the verb)	e.g. Classify, list, build

	DEFINITION	EXAMPLE
Condition	The technology the student will perform the task with	e.g. Azure Containers, MS Word
Standard	By what standard do you expect the learner to achieve?	The student will be able to solve 9 out of 10 quadratic equations <i>without error</i> e.g. With minimal guidance, by rote

Examples

	OBJECTIVE	ANALYSIS
Do	Upon completion of this module the learner will be able to list the three service models offered in Azure.	Verb is in present tense and makes sense in the format/ follows the format
Don't	Working with the three service models in Azure.	Present progressive tense does not make sense in that format and makes the objective less clear to the learner / Does not follow the format

Rationale

- **Standardization:** A standard format gives all of our learning objectives the same structure. Learners will memorize the format which will make it easier for them to understand a title when they see it for the first time.

Rule: Write from the learner's point of view

Details

Learning objectives should be written from the student's point of view.

Examples

	OBJECTIVE	ANALYSIS
Do	Upon completion of this module/unit, the learner will be able to create a workflow to route and process data using Azure Logic Apps.	Written from the learner's point of view
Don't	This module will focus on building workflows.	Focuses on the module

Rationale

- **Clarity:** When written from the learner's point of view, the objective will help a learner focus their attention on right concepts.

Rubric

The following is a tool for you to rate your learning objectives. A score of 12 or higher meets MSLearn standards for quality.

CATEGORY	DESCRIPTION	3	2	1
Audience	Written from the learner's point of view	Fully: By the end of this module the learner will be able to:...	N/A	Not at all: In this module we will show you:...
Behavior	Verb/action which is fully measurable using an MSLearn assessment	Fully: List, describe, apply etc.	Partly: Objective is measurable but not in a way that can be done with existing tools.	Not at all: Understand, learn, etc.
Specific Condition	Describes how student will accomplish objective	Fully e.g. ...in PowerShell	N/A	Not at all: How task is accomplished is not included.
Standard	Clearly Based on a Standard	Fully e.g. learner will be able to solve a quadratic equation correctly 9 out of 10 times.	N/A	Not at all
Structure	audience > behavior > condition > standard	Fully	N/A	Not at all

How to write prerequisites

1/14/2022 • 5 minutes to read

In this article, you will:

- Write prerequisites in the standard format

Overview

Research shows that as much as 70% of learning depends on having the right prerequisite knowledge. This is because people learn by adding new information to what they already know. Accurate prerequisites help our learners decide whether a module is right for them and help ensure their success.

Definition

- A *prerequisite* is prior knowledge or skills a learner must have to succeed in a module

Importance

- **Learner:** Prerequisites allow the learner to evaluate if they're ready for a module.
- **Author:** Prerequisites help the author decide what content they can omit from the training content.

Video

The following video summarizes this guideline.

Rule: Do not reference Microsoft Learn modules or Docs content

Details

- Focus on the specific skills the learner needs rather than courses or documentation that include those skills

Examples

	PREREQUISITE	ANALYSIS
Do	Ability to navigate the Azure portal	Follows the rule
Don't	Take Tour Azure services and features	Requires Microsoft Learn module
Do	Ability to execute scripts with PowerShell	Follows the rule
Don't	Learner should have taken Automate tasks using scripts with PowerShell	Requires Microsoft Learn module

	PREREQUISITE	ANALYSIS
Don't	Learner should have read How to Write and Run Scripts in the Windows PowerShell ISE	Requires Docs content

Rationale

- A student can self-evaluate if a module is appropriate for them
 - Minimizes barriers to entry
 - Helps the modules be more loosely coupled which makes it easier to reuse a module in a different learning path
-

Rule: Be concrete

Details

- Use clear and unambiguous language to describe a prerequisite

Examples

	PREREQUISITE	ANALYSIS
Do	Ability to write HTML at a novice level	Follows the rule
Don't	Familiarity with HTML	'Familiarity' is ambiguous
Do	Ability to navigate the Azure portal	Follows the rule, e.g. Navigate is specific
Don't	Experience with the Azure cloud environment	Leaves room for interpretation, e.g. Navigate? Manage? Administer?
Do	Familiarity with programming concepts such as loops, conditionals and variables	Follows the rule, e.g. Specific skills indicate level of experience required
Don't	Familiarity with programming concepts	Leaves room for interpretation, e.g. What concepts?

Rationale

- Ambiguous language leaves room for the learner to make assumptions about what they need to know. For example, a prerequisite such as "familiarity with programming concepts" might lead an inexperienced programmer to assume they are ready for a module which requires deep knowledge they do not have. This can lead them down a path of failure which will lead to a bad experience.
-

Rule: Use bulleted lists

Details

- Make short, clear statements formatted as bullet points, don't use complete sentences with periods.

Examples

Do

- Experience navigating the Visual Studio user interface
- Familiarity with web app concepts such as http and rest
- Basic programming knowledge which includes use of variables, data types, methods, conditionals and loops
- Ability to write HTML at a novice level

Don't

This module assumes you have experience navigating the Visual Studio user interface, familiarity with web app concepts such as http and rest, basic programming knowledge which includes use of variables, data types, methods, conditionals and loops.

Rationale

- Respects the learner's time because bulleted lists are faster and easier to consume than long form sentences and paragraphs.
 - Each prerequisite is presented in isolation so the learner can evaluate them one at a time.
-

Rule: Quantify experience using levels

Details

- List skill first followed by a level such as beginner, intermediate or advanced. Avoid using time as a measurement.

Examples

	PREREQUISITE	ANALYSIS
Do	Experience writing C# at the beginner level	Follows the rule
Don't	Six months experience writing C#	Uses time as a measurement
Do	Ability to write HTML at an intermediate level	Follows the rule
Don't	Ability to write intermediate level HTML	Level in the middle of the statement

Rationale

- Using time as a measurement doesn't imply a specific skill set, what a person can achieve in a period of time won't always be consistent.
 - The standardization always places the level in same position. This lets the student quickly find the level or realize there isn't a qualifying level.
-

Rule: Include software setup requirements

Details

- List all necessary software requirements needed to complete the module exercises.
- Include download/install links as appropriate.

Examples

	PREREQUISITE	ANALYSIS
Do	Must have Visual Studio Community Edition with the .NET Core workload installed	Follows the rule
Don't	None	If the student needs software in order to complete an exercise, they need to be told in the prerequisites

Rationale

- Students will look to the prerequisites to ensure they are ready for a module, listing the software requirements up front will avoid unnecessary frustration.
- The ability to start a exercise without the distraction of software setup helps keep the learner focused on the learning content.

Rubric

Use this rubric to evaluate your titles. For each row, find the column that most-closely matches your title and note the number of points in the header of that column. Sum the points earned from each row. The goal is a score of 7 (9 if listing software).

RULE	GUIDANCE	2 POINTS	1 POINT
Do not reference other Microsoft Learn modules or Docs content	Prerequisite does not require another module or documentation	Fully e.g. <i>Ability to navigate the Azure the portal</i>	Not at all e.g. Take Tour Azure services and features
Be concrete	Use clear and unambiguous language to describe a prerequisite	Fully e.g. <i>Ability to navigate the Azure the portal</i>	Not at all e.g. <i>Experience with the Azure cloud environment</i>
Use bulleted lists	Make short, clear statements formatted as bullet points	Fully e.g. <i>Prerequisites formatted in a bulleted list</i>	Not at all e.g. <i>Prerequisites formatted as a sentence or paragraph</i>
Quantify experience using levels	List skill first followed by a level such as beginner, intermediate or advanced, Avoid time as a unit of measurement	Fully e.g. <i>Experience writing C# at the beginner level</i>	Not at all e.g. <i>Six months experience writing C#</i>
Include software setup requirements	When necessary, list any software setup needed to be successful in the module exercises	Fully e.g. <i>Must have Visual Studio Community Edition with the .NET Core workload installed</i>	Not at all e.g. <i>none</i>

How to write scenarios

1/14/2022 • 9 minutes to read

In this article, you will:

- Write a scenario that helps the learner apply the technology to their own work
 - Align your scenario with the key benefits of the technology
-

Overview

A story is more interesting and easier to remember than a list of facts. Think about the last time you read a good fiction book. How did the author present the background information for the main conflict? Did they dump it all on you at once or did they incorporate it into the story? Which style is more fun to read? Which is easier to remember?

Our scenarios use this idea: we take everything we want to teach and weave it into a story. This makes it easier to remember and shows the learner how all the pieces fit together to solve a problem. You create one scenario per module: describe the entire scenario at the module level and a sub-task of the scenario in each unit.

Definition

- A *scenario* is a fictionalized real-world job-task presented as a story that shows how the technology is used in practice.

Importance

- Adults prefer to study things that are immediately relevant to their work. Our scenarios help them make the connection between our technology and their job-tasks.
 - Humans find it easier to remember information when it's presented in context. Our scenarios provide that context by showing how to use the technology to solve a real-world problem.
-

Video

The following video summarizes this guideline.

Rule: Tell a specific story

Details

- Write your scenario as a fictionalized real-world story. Make it specific by including an *industry*, an *event*, and a *problem* to be solved. Do this even for "Intro to..." content that is inherently general.

Example

	SCENARIO	ANALYSIS
Do	"Suppose you work at an athletic shoe company as the IT expert on the marketing team. You are launching a new basketball shoe created by a celebrity athlete. Previous launches resulted in a social media frenzy as people shared stories and photos of themselves wearing their new shoes. The marketing team wants to monitor hundreds of social media sites, blogs, and websites to determine consumer reaction. You expect there will be more data than you can process manually, and you don't have the server resources to build an automated solution in-house. You need a cloud-hosted system that integrates monitoring, text analytics, database storage, and email notifications."	Follows the rule.
Don't	"This module shows you how to automate the processing of data in the cloud. You will create a workflow that runs automatically when data is available. You will send the data to an Azure service for processing and to a database for storage. To document the process, you will send notifications via email after each step."	Does not give intuition about why each step is needed, how they fit together, or the problem being solved.
Do	"You work on a personal-safety app that lets people send text messages and location information to their family members after an emergency or natural disaster. Demand is low most of the time but can spike to extremely high levels without warning. Your cloud architecture needs to scale quickly to handle this variable demand."	Follows the rule.
Don't	"It's common to have spikes in demand which means your cloud architecture needs to scale under load."	Does not describe industry or event.

Rationale

- Adults are motivated to learn when they have a problem they need to solve. Our scenarios align with this principle because they show how to apply our technology to solve a problem. This makes it easier for the learner to see how the training will help them with their own work.
- Specificity turns your *scenario* into a *story* which makes it easier for humans to remember.
- Specificity help learners see your scenario as a real-world problem.

NOTE

A common objection to specificity is that the learner might think "this doesn't match *my* job". We believe it's ok if the scenario doesn't match their job exactly. Learners will be able to translate the scenario to their own work. The specificity will make the key points memorable and relatable.

Rule: Make it anonymous

Details

- Do not use company, product, or employee names in your scenario. Avoid gendered pronouns.

Example

	SCENARIO	ANALYSIS
Do	"Suppose you work at an athletic shoe company as the IT expert on the marketing team. You are launching a new basketball shoe created by a celebrity athlete. Previous launches resulted in a social media frenzy as people shared stories and photos of themselves wearing their new shoes. The marketing team wants to monitor hundreds of social media sites, blogs, and websites to determine consumer reaction...."	Follows the rule.
Don't	"Suppose you work at Contoso Shoes as the IT expert on the marketing team. You are launching a new basketball shoe called the Fabrikam created by a celebrity athlete. Previous launches resulted in a social media frenzy as people shared stories and photos of themselves wearing their new shoes. Your boss, Sarah, has asked you to monitor hundreds of social media sites, blogs, and websites to determine consumer reaction...."	Includes company, product, and employee names.

Rationale

- Names add extraneous detail for the reader to remember. This distracts from the core points of the scenario.

Rule: Keep it simple

Details

- Choose scenarios that require only common knowledge to understand; minimize domain-specific information.

Examples

SCENARIO	EXAMPLE	ANALYSIS
Do	"... After you submit your expense report, it will be sent to your manager for approval ..."	Follows the rule.
Don't	"... After you submit your expense report, it must be approved by the closest manager above you in the org chart who has a safe-limit greater than the amount of the expense ..."	Includes company-specific detail that most learners will not understand.
Do	"Suppose you work at a phone company that bills customers by the minute for calls. You need to store phone-call metadata like the phone numbers involved and the call duration."	Follows the rule. The amount of data and the type of data stored are simple to understand without requiring domain-specific knowledge.
Don't	"Suppose you work at an oil company and your job is to locate new petroleum reservoirs underground. You do this using a 3D seismic survey that maps the composition and shape of underground rock formations since dome-like structures can trap oil underneath. To collect the data, you position thousands of geophone sensors across the area-of-interest. You then use specialized trucks to generate a seismic wave and your sensors take measurements of the reflections a thousand times per second."	Requires domain-specific information which makes the scenario longer and more difficult to understand. Even with the explanation, the data to be stored is not intuitive (e.g. what does a geophone measure?).

Rationale

- Scenarios should be quick and easy for the learner to understand. Learners are unlikely to have the time and patience to read a complicated scenario about a specialized application of the technology.
- Scenario should be easy to remember. The learner should be able to mentally refer to the scenario as they proceed through the training. This will help them put each piece of new information in context; that is, it will help them see how the technology is applied to the problem being solved.

Rule: Match the problem

Details

- Provide intuition for the key learning points.

Examples

	GOAL	SCENARIO	ANALYSIS

	GOAL	SCENARIO	ANALYSIS
Do	Motivate unexpected spikes in demand.	"You work on a personal-safety app that lets people send text messages and location information to their family members after an emergency or natural disaster."	Follows the rule.
Don't	Motivate unexpected spikes in demand.	"You write a political blog and demand increases sharply before every election."	The spike in demand is predictable.
Do	Motivate high availability.	"You work on the transaction-processing service for a major stock exchange."	Follows the rule.
Don't	Motivate high availability.	"You are building a web service to collect temperature data from a sensor network. Your service loads the temperature data into the database used by your weather app."	A short amount of downtime is not a critical failure here. Temperature changes slowly and the app could display older temperature data until new values become available.

Rationale

- The goal of a scenario is to show how our technology solves a real-world problem. A scenario that doesn't match the problem is misleading and might teach the wrong lesson. For example, consider a scenario about a "messaging app to connect with friends and family after an emergency". Would that work better for Azure Functions (which scale immediately to handle unexpected demand) or Azure Virtual Machines (which take a few minutes to scale)?

Rule: Use one scenario per module

Details

- Use one scenario for an entire module. The unit-level scenario will be a subtask of the module-level scenario.

Examples

TITLE	SCENARIO
Module: "Route and process data automatically with Azure Logic Apps"	"Suppose you work at an athletic shoe company as the IT expert on the marketing team. You are launching a new basketball shoe created by a celebrity athlete. Previous launches resulted in a social media frenzy as people shared stories and photos of themselves wearing their new shoes. The marketing team wants to monitor hundreds of social media sites, blogs, and websites to determine consumer reaction. You expect there will be more data than you can process manually, and you don't have the server resources to build an automated solution in-house."

TITLE	SCENARIO
Unit 1: "Launch your Logic App automatically when new data is available"	"Recall that your goal in the shoe-company scenario is to monitor social media for content about your new product launch. A key part of your plan is to scan Twitter for tweets that mention your product. For each matching tweet, you need to extract the text in order to determine if it is positive or negative."

Rationale

- Our modules are short, and it would be tedious and confusing for learners if we used a different scenario for each unit.
-

Rubric

Use this rubric to evaluate your work. For each row, find the column that most-closely matches your content and note the number of points in the header of that column. Add up the points earned from the rows. The goal is a score of 19 or higher.

	4 POINTS	3 POINTS	2 POINTS	1 POINT
Written as a concrete story containing the three standard elements: industry name, event, and problem to be solved	Includes all three standard elements	Includes two standard elements	Includes one standard element	Includes zero standard elements
Does not contain proper nouns for the company, products, or employees in the scenario	Includes no names	Contains one company or product name	Contains one or more employee names	Contains two or more types of names
Use of gendered pronouns	Does not contain gendered pronouns	Not applicable	Not applicable	Contains gendered pronouns
Requires only common knowledge to understand	Typical life experience or casual following of current events is enough to understand the scenario	Experience as a user or customer of the industry is needed to understand the scenario	A higher-than-usual interest in the industry is needed to understand the scenario (e.g. reads tech blogs, follows politics closely, knows all the teams in the sports league, etc.)	Work experience in the industry is needed to understand the scenario
Provides intuition for the key learning points	Every part of the scenario aligns with the solution presented	Not applicable	Not applicable	Any part of the scenario does not directly motivate the solution being taught

	4 POINTS	3 POINTS	2 POINTS	1 POINT
Uses one scenario for the module and units	Uses one scenario	Uses two scenarios	Uses three scenarios	Uses more than three scenarios

How to write introductions

1/14/2022 • 6 minutes to read

In this article, you will:

- Write module and unit introductions in the standard format

Overview

People say you can skip the first part of any content and not miss anything (I bet you were tempted to bypass this paragraph). Our introductions describe the real-world problem our content will solve. This lets customers decide whether our content is right for them. That same real-world problem is used as the basis for the exercises, so we're not wasting their time by asking them to read our introductions.

Definition

- An **introduction** tells the learner what's in the content and describes a problem it will solve.

Importance

- **Learner:** We want to value the learner's time by helping them quickly decide whether the content meets their needs.
- **Author:** A clear statement of a real-world problem helps the author keep the content focused. Only content needed to solve the problem should be included.

Video

The following video summarizes this guideline.

Rule: Use the standard module-introduction format

Details

- Write module introductions in **topic > scenario > TOC > goal** format. The scenario component must follow the **scenario** rules. Total length should be 200-300 words.

Examples

Module: *Route and process data using Azure Logic Apps*

AREA	INTRO
Topic sentence(s)	Azure Logic Apps let you automate your business processes by running your workflows in the cloud.

AREA	INTRO
Scenario	Suppose you work at an athletic shoe company as the IT expert on the marketing team. You are launching a new basketball shoe created by a celebrity athlete. Previous launches resulted in a social media frenzy as people shared stories and photos of themselves wearing their new shoes. The marketing team wants to monitor hundreds of social media sites, blogs, and websites to determine consumer reaction. You expect there will be more data than you can process manually, and you don't have the server resources to build an automated solution in-house. You need a cloud-hosted system that integrates monitoring, text analytics, database storage, and email notifications.
Prose table-of-contents	In this module, you will create a business workflow using Azure Logic Apps to automate the processing of news articles. Your workflow will trigger when a new article is available. It will use cloud services to determine if the article is positive or negative and branch based on the results. After processing, it will persist the data in a cloud-hosted database.
Goal (i.e. terminal learning objective)	By the end of this module, you will be able to create workflows which route and process data using Azure Logic Apps and its built-in connectors.

Module: *Create an Azure Storage account*

AREA	INTRO
Topic sentence(s)	Organizations often have diverse requirements for their cloud-hosted data. For example, they may need to store data in a specific region or get separate invoices for different data categories. Storage accounts let you formalize these types of policies and apply them to your Azure data.
Scenario	Suppose you work at a chocolate manufacturer that produces baking ingredients such as cocoa powder and chocolate chips. Your formulations and manufacturing processes are trade secrets. The spreadsheets, documents, and instructional videos that capture this information are critical to your business and require geographically-redundant storage. This data is primarily accessed from your main factory, so you would like to store it in a nearby datacenter. The expense for this storage needs to be billed to the manufacturing department. You also have a sales group that creates recipes and baking videos to promote your products to consumers. Your priority for this data is low cost, rather than redundancy or location. This storage must be billed to the sales team.
Prose table-of-contents	This module shows you how to analyze your data-storage requirements to determine how many storage accounts you need. You will then map those requirements to the corresponding storage-account settings. Once you know the right settings, you will use the Azure portal to create a matching storage account.

AREA	INTRO
Goal (i.e. terminal learning objective)	By the end of this module, you will be able to create multiple Azure storage accounts, each with the appropriate settings for the data it holds.

Rationale

- Uniformity values the learner's time. Learners will become familiar with the format, so they will be able to read it quickly.
- Every part of our introduction helps the learner decide if the content will solve their job-tasks. We start with a general sentence about the broad topic area that will be covered - if this doesn't match the learner's expectations, they can stop reading and reject the module immediately. We next present a real-world problem that the module will solve. This helps the learner decide how well the content will transfer to their own work. We then summarize the technical topics and sub-tasks that are covered in the module. We finish by reminding them of the job-task that the module addresses; this is often called the *terminal learning objective* and is typically formed by expanding the module title into a complete sentence with a bit of added detail.
- The scenario will be the basis of any exercise in the module. This means their study of the scenario gives them two benefits: it helps them decide if the module will solve their own problem and prepares them to do the exercise.

Rule: Use the standard learning-unit introduction format

Details

- Write learning-unit introductions in *topic > scenario > TOC* format. The scenario component must follow the [scenario rules](#).

Example

Module: *Create an Azure Storage account*

Learning unit: *Decide how many storage accounts you need*

AREA	INTRO
Topic sentence	Organizations often have multiple storage accounts to let them implement different sets of requirements.
Scenario sub-task	In the chocolate-manufacturer example, there would be one storage account for the private business data and one for the consumer-facing files.
Prose table-of-contents	Here, you will learn the policy factors that are controlled by a storage account so you can decide how many accounts you need.

Rationale

- Uniformity values the learner's time. Learners will become familiar with the format, so they will be able to read it quickly.
- Follows standard learning theory by stating what they will learn and putting it in a real-world context.

Rule: Use the standard exercise-unit introduction format

Details

- Write exercise-unit introductions in *topic > scenario > task* format.

Examples

Module: *Create an Azure Storage account*

Exercise unit: *Create a storage account using the Azure portal*

AREA	INTRO
Topic sentence	A storage account represents a collection of settings that implement a business policy.
Scenario sub-task	Recall that in the chocolate-manufacturer example, there would be a separate storage account for the private business data. There were two key requirements for this account: geographically-redundant storage because the data is business-critical and at least one location close to the main factory.
Task performed in the exercise	Here, you will create a storage account with settings appropriate to hold this mission-critical business data.

Rationale

- Uniformity values the learner's time. Learners will become familiar with the format, so they will be able to read it quickly.
- Clearly states the task at a high level. This helps the learner understand the purpose of each step as they work through the exercise.

Rubric

Use this rubric to evaluate your work. For each row, find the column that most-closely matches your content and note the number of points in the header of that column. Repeat the learning-unit row for each learning unit. Repeat the exercise-unit row for each exercise unit. Add up the points earned from the rows. The goal is a score of *(2 * number of learning units + 2 * number of exercise units + 6)* or higher.

POINTS	4	3	2	1
Module introduction word count	200-300 words	100-200 words	more than 300 words	Less than 100 words
Module introduction components	All 4	Scenario and any other 2	Scenario and any other 1	No scenario
Learning-unit introduction components	N/A	All 3	Scenario and any other 1	No scenario

POINTS	4	3	2	1
Exercise-unit introduction components	N/A	All 3	Scenario and any other 1	No scenario

How to write knowledge checks

1/14/2022 • 9 minutes to read

In this article, you will:

- Write knowledge checks that align to learning objectives
 - Write questions that support learning
 - Structure knowledge checks correctly
-

Overview

Knowledge checks can be difficult to write. Should you use True/False questions? How many answers should you have? Should you write in the second or third person? Writing good questions is a well-studied problem and we've summarized the most common guidelines in this document.

Definition

- A **knowledge check** is a [two to five question assessment](#) that measures if the learner has acquired the skills outlined in the learning objectives.

Importance

- **Learner:** Lets the learner assess if they are acquiring the skills described by the learning objectives.
 - **Author:** Assess if the instruction is effective.
-

Video

The following video summarizes this guideline.

Rule: Structure your knowledge checks

Details

1. Write answers that are all about the same length

- All answers in a multiple choice question should be about the same length, this avoids cueing the learner to select the longest or shortest answer in a group. The longest answer is typically the correct answer and many seasoned test takers are aware of that.

2. Provide three plausible answers per question to choose from

- To manage learner expectations MSLearn has standardized on three answers per question.
- Authors generally have an easy time creating two incorrect answers, but a difficult time creating three incorrect answers. This means that the third incorrect answer is often not plausible and is easily dismissed as incorrect by the learner.

3. Don't write True/False questions

- It's difficult to write a question that doesn't clue the learner to the answer.

4. Don't include "All of the above" and/or "None of the above" as answer choices

- All of the Above answers allow the learner to select a correct answer based on partial knowledge.
- None of the Above answers measure if a student can detect wrong answers, rather than the knowledge they have attained.
- Studies have shown that when these are shown in the answer list, it's highly probable they are the correct answer. Seasoned test takers are aware of this and may just select it without reading the question.

5. Write questions as a complete sentence ending with a question mark (*closed* format)

- Closed format questions are easier for the learner to understand than *open* format questions.

	QUESTION	FORMAT
Do	What year was Microsoft founded?	Closed Format
Don't	The year that Microsoft was founded:	Open Format

6. Avoid using the words "Not" and "Except" in questions

- Studies show that learners will fail to notice these words and therefore misinterpret the question.

7. If the answers are in numeric value, list them in sorted order

- Studies show that even when the learner knows the answer they will be unable to locate the correct answer in an unsorted list.

8. Use third person, not "you"

- The use of "you" becomes too personal to the learner and distracts them.

Rule: Write questions that support learning

Details

1. Provide a meaningful explanation for both correct and incorrect answers

- Meaningful feedback justifies the correct answer so that it reduces the chance the student will challenge the question.
- Studies show that immediate feedback is helpful in learning and performance.
- The learner may have guessed the correct answer by eliminating wrong answers, so it's best to reinforce the reasoning for the correct answer.

Example

```

- content: "What needs to be installed on your machine to let you execute Azure PowerShell cmdlets locally?"
```

choices:

```

- content: "The Azure cloud shell"
```

isCorrect: false

explanation: "The cloud shell does let you execute Azure cmdlets; however, you use it in a browser, so you are not running the commands on your local machine."

```

- content: "The base PowerShell product and the AzureRM module"
```

isCorrect: true

explanation: "You need both the base PowerShell product and the AzureRM module. The base product gives you the shell itself, a few core commands, and programming constructs like loops, variables, etc. The AzureRM modules adds the cmdlets you need to work with Azure resources."

```

- content: "The Azure CLI and Azure PowerShell"
```

isCorrect: false

explanation: "The Azure CLI is an alternate command-line and scripting tool. It is not needed if you are going to use Azure PowerShell."

2. Align questions with learning objectives

- Learning objectives are statements that define the demonstrable skills or knowledge that will be acquired by a student as a result of instruction. We use knowledge checks as a way for the learner to assess if they have grasped the material.

For each of your learning objectives find the row which most closely aligns with your objective's level of learning. The following chart provides examples of the types of questions you would write based on the learning level.

LEARNING LEVEL & EXAMPLE VERBS	TYPES OF QUESTIONS	EXAMPLE
Create Pull many concepts together to create original work. E.g. Student will be able to: - plan - produce - generate	<i>It is very difficult to write multiple choice questions at the create level.</i> You might get to this level with a homework assignment or challenge where the student would create a plan for a project, or design a website or app	<i>Design a dynamic website for a shoe company which requires separate authentication for employees and customers.</i>
Evaluate Test a solution based on specific criteria. E.g. Student will be able to: - critique - interpret - evaluate	Which would be a better solution to [scenario]? What could be done to integrate ___ and ___? A company is using ___ to store ___, what is an alternative solution that would reduce ___? How else could you accomplish ___? Examine the following scenario, how would you solve the issue they are experiencing?	<i>What are three Azure services you could use to create a website that stores user information, and allows registered users to upload and display images?</i>

LEARNING LEVEL & EXAMPLE VERBS	TYPES OF QUESTIONS	EXAMPLE
Analyze Break a concept into parts, see how parts work together. E.g Student will be able to: -compare - differentiate - discriminate	In what order would you perform the following steps to create ___? A pro of ___ is ___? A ___ is experiencing ___ problem, which is the most likely cause? What is the order of importance? What is the reason this is not an appropriate solution to ___?	<i>A company has recently decided to switch from MySQL to PostgreSQL, what is the advantage of that decision?</i>
Apply Use taught procedures or processes to complete a task. E.g Student will be able to: -demonstrate - employ -illustrate	What would you use ___ to do ___? Predict what might happen if ___? What is the next step in ___? How would you use ___? What would you need to know in order ___?	<i>Jim is adding a new secret to an existing Azure Key Vault. Using the CLI, which of the following commands will display the secret as plain text?</i>
Understand Demonstrate comprehension. E.g Student will be able to: -define - explain - rephrase	How would you describe ___? Identify the ___? How are these alike ___? What tool would one use to do ___? What is the difference between ___? Which does not belong?	<i>What are two similarities of MySQL and PostgreSQL?</i>
Remember Recall what they've been taught. E.g Student will be able to: - recall -repeat -identify	Which two services do/are ___? Where can you find ___? What is ___? Which is the best definition for ___? What are the characteristics of ___?	<i>Which of the following is a characteristic of a PostgreSQL database?</i>

3. No more than 1/3 of your questions should be at the **Remember or Understand** level.

- Questions at this level are less engaging and demonstrate a lower level of learning.

4. The majority of the questions should be at the **Apply, Analyze and Evaluate** level.

- Questions which measure higher levels of learning are more interesting and more demonstrative of gained skills.

Rubric

Use this rubric to evaluate your knowledge checks. For each row you'll find a rule for learning objectives, note the number of points your questions and/or answers receive from the header of that column. Sum the points earned from each row. The goal is a score of 32 or higher.

CRITERIA	EXPLANATION	3	2	1
Answers are the same length	All answers should all be within 10% of the same word count or sentence length	Fully e.g. <i>three 1-word answers</i>	Not Applicable	Not at all e.g. <i>answers vary in length</i>
Three answers provided	There are three possible answers to choose from	Fully e.g. <i>three answers provided</i>	Not Applicable	Not at all e.g. <i>More or less than three answers</i>
No True/False Questions	No True/False questions	Fully e.g. <i>No True/False questions</i>	Not Applicable	Not at all e.g. <i>TBD</i>

CRITERIA	EXPLANATION	3	2	1
No All of the Above/None of the Above	Answer choices do NOT include all of the above or none of the above	Fully e.g. A. Containers B. Blobs C. MySQL	Not Applicable	Not at all e.g. A. Containers B. Blobs C. All of the Above
Questions phrased in <i>closed</i> format	Question is a complete sentence with a questions mark	Fully e.g. Which is the best definition of a container in Azure?	Not Applicable	Not at all e.g. An Azure container is...
Question does not use the words "Not" and "Except" in questions	Question should ask for an answer that 'is', as opposed to what is 'not'	Fully e.g. _Which one is an example of...? _	Not Applicable	Not at all e.g. Which is NOT an example of...?
If the answers are in numeric value, list them in sorted order	If answers are numeric they are sorted in order	Fully e.g. A. 2 B. 3 C. 4	Not Applicable	Not at all e.g. A. 3 B. 4 C. 2
Question is in the third person	Question does not use the word "you"	Fully e.g. Jim is adding a secret to a Key Vault...	Not Applicable	Not at all e.g. You are adding a secret to the Key Vault...
Meaningful feedback provided	Feedback which explains why an answer is correct or best	Fully e.g. Visual Studio Code is cross-platform and supports Linux, macOS, and Windows.	Mostly e.g. Visual Studio is the correct answer	Not at all e.g. No feedback provided
Questions are aligned with learning objectives	Questions measure if student has achieved the learning objective	Fully e.g. _Knowledge check clearly measures a defined learning objective _	Mostly e.g. _Knowledge check measures a topic but not at the correct level (i.e. remember as opposed to apply) _	Not at all e.g. Knowledge check doesn't align to any learning objectives
No more than 1/3 of questions are at Remember/Understand	Avoid asking too many questions about facts and basic comprehension	Fully e.g. 1/3 or less at this level	Not Applicable	Not at all e.g. More than 1/3 are at this level
The majority of the questions should be at the Apply, Analyze and Evaluate level	Aim to test higher levels of learning	Fully e.g. 2/3 or more at this level	Not Applicable	Not at all e.g. Less than 1/3 are at this level

How to write module summaries

1/14/2022 • 14 minutes to read

In this article, you will:

- Write a module summary unit that aids in memory retention

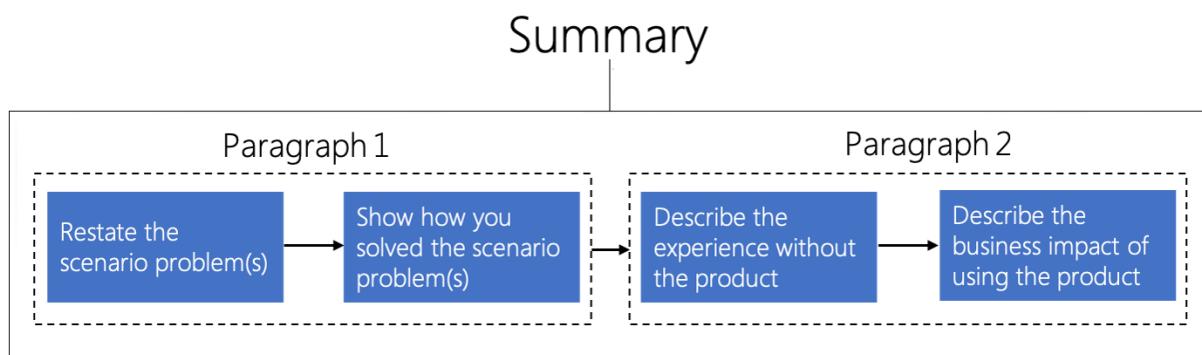
Overview

Authors often struggle to create a module summary. It's usually the last thing we write. We consider it an afterthought or less important than the substance of a module. The truth is, the summary is what learners are most likely to remember.

There is a common saying coined by Aristotle: "tell them what you're going to tell them, tell them, and then tell them what you told them". There is evidence to support this advice. In instructional design, we call this the *serial position effect*. Studies show that learners are most likely to remember the first and last things we tell them and forget what happens in the middle. There are two main reasons why this effect occurs, they are known as the *primacy effect* and *recency effect*.

- The *primacy effect* is the idea that when a learner begins a course their mind is clear and they have the cognitive resources to encode new information to long term memory. As a course progresses, more and more information is thrown at them and they have fewer cognitive resources to apply so they are less able to remember new content.
- The *recency effect* states that learners are most likely to remember the thing they heard most recently. The recency effect is the key thing that makes our summaries important. It's our opportunity to remind them of the most important things they learned in a module.

In this article, we'll describe how to write an effective summary that will help learners retain what they learned. We've broken the summary into four steps that are combined to create the summary. For a typical summary, you'll merge the four parts into two paragraphs as shown below. If you have a longer summary, you can place each of the four pieces in their own paragraph.



Definition

- A module *summary* is a unit that briefly reiterates the most salient points of a module in the context of the scenario.

Importance

- **Learner:** When learners remember the context of why and how to apply new knowledge, they will be able to transfer that to their own work.

- **Author:** Following a standard pattern will make writing effective summaries faster.
-

Video

The following video summarizes this guideline.

Rule: Restate the scenario problem(s)

Details

- Summarize the challenge(s) posed in the introduction scenario. Be brief (1-2 sentences).

Example 1

Module: Route and process data automatically using Logic Apps

Introduction scenario

Suppose you work at an athletic shoe company as the IT expert on the marketing team. You're launching a new basketball shoe created by a celebrity athlete. Previous launches resulted in a social media frenzy as people shared stories and photos of themselves wearing their new shoes. The marketing team wants to monitor hundreds of social media sites, blogs, and websites to determine consumer reaction. There will be more data than you can process manually. You would like a cloud-hosted system that automates monitoring, text analytics, database storage, and email notifications.

Summary with current rule emphasized

Our marketing team needed to gauge customer response to our new shoe. We wanted to monitor social media, determine customer reaction, and route the data to a database or to customer service based on sentiment.

Azure Logic Apps let us automate the process. Each step of our workflow was mapped to one of the built-in components. We used the Twitter trigger to detect mentions of our product and launch our app. The Azure Cognitive Services action let us analyze whether the tweets were positive or negative. A control action helped us decide where to route the tweet based on sentiment. Finally, we used actions to insert a row in SQL Server and send an Outlook email.

Imagine how much work it would be to build this app from scratch. We'd have to write code to access each of the services' APIs. We'd need a polling infrastructure to monitor Twitter and trigger our app when new tweets were available. Once all the code was ready, we'd need servers to host the app.

Logic Apps made it easy. The standard connectors did the hard work of integrating systems that were never designed to work together. Building the app took us less than an hour and we didn't have to write any code or set up any servers. We can now analyze every tweet about our product and our business analytics can use the data to shape future products.

Example 2

Module: How Plain English improves your writing

Introduction scenario

You work for a fire extinguisher manufacturer and are writing the operating instructions for a new model. This

extinguisher has a new one-handed activation method which is meant to be quicker and more effective. The instructions must be understood quickly by people of all reading levels. Feedback on your first draft indicates your instructions are too long, too descriptive, and contain too many technical terms.

Summary with current rule emphasized

You are writing the instruction manual for a new model fire extinguisher. The instructions must be quickly read and understood by a wide variety of people.

You did some research and found that Plain English is a good writing style for safety-critical communications. For your next draft, you applied several Plain English techniques to your instructions. You removed unnecessary words, which made your sentences easier to read even in a stressful situation like a fire. Then you made sure all sentences used the active voice, which made your content more direct. Next, you replaced fire-industry jargon with everyday words, which made the instructions accessible to a wider audience. Finally, you replaced a comma-delimited list with a bulleted list, which made the steps to activate the fire extinguisher easier to follow.

Fire extinguishers are critical safety equipment for both homes and businesses. Despite their importance, many customers don't read the instructions ahead of time. Confusing instructions could mean customers don't use the extinguisher correctly when they're needed. This can result in loss of property or life.

The test for effective instructions is whether customers can use your extinguishers correctly during an emergency. Users that fail might blame the instructions or the product. In either case, it's not good for business. On the other hand, successful customers are likely to share their stories and become advocates for your product.

Rationale

- Reminding the learner of the problems we solved will help them remember when they should use this product.
-

Rule: Show how you solved the scenario problem(s)

Details

- Describe how you used the product to solve the problem(s) posed in the introduction scenario.

Example 1

Module: Route and process data automatically using Logic Apps

Summary with current rule emphasized

Our marketing team needed to gauge customer response to our new shoe. We wanted to monitor social media, determine customer reaction, and route the data to a database or to customer service based on sentiment.

Azure Logic Apps let us automate the process. Each step of our workflow was mapped to one of the built-in components. We used the Twitter trigger to detect mentions of our product and launch our app. The Azure Cognitive Services action let us analyze whether the tweets were positive or negative. A control action helped us decide where to route the tweet based on sentiment. Finally, we used actions to insert a row in SQL Server and send an Outlook email.

Imagine how much work it would be to build this app from scratch. We'd have to write code to access each of the services' APIs. We'd need a polling infrastructure to monitor Twitter and trigger our app when new tweets were available. Once all the code was ready, we'd need servers to host the app.

Logic Apps made it easy. The standard connectors did the hard work of integrating systems that were never designed to work together. Building the app took us less than an hour and we didn't have to write any code or set up any servers. We can now analyze every tweet about our product and our business analytics can use the data to shape future products.

Example 2

Module: How Plain English improves your writing

Summary with current rule emphasized

You are writing the instruction manual for a new model fire extinguisher. The instructions must be quickly read and understood by a wide variety of people.

You did some research and found that Plain English is a good writing style for safety-critical communications. For your next draft, you applied several Plain English techniques to your instructions. You removed unnecessary words, which made your sentences easier to read even in a stressful situation like a fire. Then you made sure all sentences used the active voice, which made your content more direct. Next, you replaced fire-industry jargon with everyday words, which made the instructions accessible to a wider audience. Finally, you replaced a comma-delimited list with a bulleted list, which made the steps to activate the fire extinguisher easier to follow.

Fire extinguishers are critical safety equipment for both homes and businesses. Despite their importance, many customers don't read the instructions ahead of time. Confusing instructions could mean customers don't use the extinguisher correctly when they're needed. This can result in loss of property or life.

The test for effective instructions is whether customers can use your extinguishers correctly during an emergency. Users that fail might blame the instructions or the product. In either case, it's not good for business. On the other hand, successful customers are likely to share their stories and become advocates for your product.

Rationale

- It's difficult to remember facts without context. Describing how this product solved our problem provides that context. You're reminding them of what they did and why they did it which makes the substance of the module more memorable.
-

Rule: Describe the experience without the product

Details

- Describe what would be required to solve the problem without using the product. Be brief (1-2 sentences).

Example 1

Module: Route and process data automatically using Logic Apps

Summary with current rule emphasized

Our marketing team needed to gauge customer response to our new shoe. We wanted to monitor social media, determine customer reaction, and route the data to a database or to customer service based on sentiment.

Azure Logic Apps let us automate the process. Each step of our workflow was mapped to one of the built-in components. We used the Twitter trigger to detect mentions of our product and launch our app. The Azure Cognitive Services action let us analyze whether the tweets were positive or negative. A control action helped us decide where to route the tweet based on sentiment. Finally, we used actions to insert a row in SQL Server and send an Outlook email.

Imagine how much work it would be to build this app from scratch. We'd have to write code to access each of the services' APIs. We'd need a polling infrastructure to monitor Twitter and trigger our app when new tweets were available. Once all the code was ready, we'd need servers to host the app.

Logic Apps made it easy. The standard connectors did the hard work of integrating systems that were never designed to work together. Building the app took us less than an hour and we didn't have to write any code or set up any servers. We can now analyze every tweet about our product and our business analytics can use the

data to shape future products.

Example 2

Module: How Plain English improves your writing

Summary with current rule emphasized

You are writing the instruction manual for a new model fire extinguisher. The instructions must be quickly read and understood by a wide variety of people.

You did some research and found that Plain English is a good writing style for safety-critical communications. For your next draft, you applied several Plain English techniques to your instructions. You removed unnecessary words, which made your sentences easier to read even in a stressful situation like a fire. Then you made sure all sentences used the active voice, which made your content more direct. Next, you replaced fire-industry jargon with everyday words, which made the instructions accessible to a wider audience. Finally, you replaced a comma-delimited list with a bulleted list, which made the steps to activate the fire extinguisher easier to follow.

Fire extinguishers are critical safety equipment for both homes and businesses. Despite their importance, many customers don't read the instructions ahead of time. Confusing instructions could mean customers don't use the extinguisher correctly when they're needed. This can result in loss of property or life.

The test for effective instructions is whether customers can use your extinguishers correctly during an emergency. Users that fail might blame the instructions or the product. In either case, it's not good for business. On the other hand, successful customers are likely to share their stories and become advocates for your product.

Rationale

- Adults seek out training when they have a problem to solve. This section shows how the product will make it easier for the learner to solve their problem. It tells the learner why their projects will have an increased probability of success if they use the product.
-

Rule: Describe the business impact

Details

- Explain the business impact of using the product to solve the problem.

Example 1

Module: Route and process data automatically using Logic Apps

Summary with current rule emphasized

Our marketing team needed to gauge customer response to our new shoe. We wanted to monitor social media, determine customer reaction, and route the data to a database or to customer service based on sentiment.

Azure Logic Apps let us automate the process. Each step of our workflow was mapped to one of the built-in components. We used the Twitter trigger to detect mentions of our product and launch our app. The Azure Cognitive Services action let us analyze whether the tweets were positive or negative. A control action helped us decide where to route the tweet based on sentiment. Finally, we used actions to insert a row in SQL Server and send an Outlook email.

Imagine how much work it would be to build this app from scratch. We'd have to write code to access each of the services' APIs. We'd need a polling infrastructure to monitor Twitter and trigger our app when new tweets were available. Once all the code was ready, we'd need servers to host the app.

Logic Apps made it easy. The standard connectors did the hard work of integrating systems that

were never designed to work together. Building the app took us less than an hour and we didn't have to write any code or set up any servers. We can now analyze every tweet about our product and our business analytics can use the data to shape future products.

Example 2

Module: How Plain English improves your writing

Summary with current rule emphasized

You are writing the instruction manual for a new model fire extinguisher. The instructions must be quickly read and understood by a wide variety of people.

You did some research and found that Plain English is a good writing style for safety-critical communications. For your next draft, you applied several Plain English techniques to your instructions. You removed unnecessary words, which made your sentences easier to read even in a stressful situation like a fire. Then you made sure all sentences used the active voice, which made your content more direct. Next, you replaced fire-industry jargon with everyday words, which made the instructions accessible to a wider audience. Finally, you replaced a comma-delimited list with a bulleted list, which made the steps to activate the fire extinguisher easier to follow.

Fire extinguishers are critical safety equipment for both homes and businesses. Despite their importance, many customers don't read the instructions ahead of time. Confusing instructions could mean customers don't use the extinguisher correctly when they're needed. This can result in loss of property or life.

The test for effective instructions is whether customers can use your extinguishers correctly during an emergency. Users that fail might blame the instructions or the product. In either case, it's not good for business. On the other hand, successful customers are likely to share their stories and become advocates for your product.

Rationale

- It's important to adult learners that training has an immediate impact on their job and/or life. Our discussion of business value explicitly makes the connection between the module content and their job tasks.

Rubric

Use this rubric to evaluate your titles. For each row, find the column that most-closely matches your title and note the number of points in the header of that column. Sum the points earned from each row. The goal is a score of 10 or higher.

POINTS	3	2	1
Restate the scenario problem(s)	Fully e.g. <i>Brief description of the problems outlined in scenario</i>	Mostly e.g. <i>Description too elaborate/long, or does not restate all of the problems</i>	Not at all e.g. <i>Does not describe the business problems posed in the scenario</i>
Show how you solved the scenario problem(s)	Fully e.g. <i>Pulls all issues described in the scenario and how each was solved</i>	Mostly e.g. <i>Pulls some issues from scenario</i>	Not at all e.g. <i>Does not describe issues posed in scenario or describe how they were solved</i>
Describe the experience without the product	Fully e.g. <i>Brief description of solving the problem without the product</i>	Mostly e.g. <i>Description too elaborate/long, or does not state how difficult the problem would be to solve without the product</i>	Not at all e.g. <i>Does not describe the experience or is not included</i>

POINTS	3	2	1
Describe the business impact	Fully e.g. <i>Clearly describes the business impact of using the product</i>	Mostly e.g. <i>Alludes to the business impact of the product</i>	Not at all e.g. <i>Does not describe the business impact</i>

Add or edit Learn content

1/14/2022 • 6 minutes to read

After you fork/clone your repository and successfully set up the remotes (origin/upstream), you are ready to add content to your repo. The Learn repositories follow a strict folder organizational structure. Before, adding your content, identify the location and structure for your content. See the [Learn repo structure](#) article to understand how the Learn repos are organized and where your content fits in under your repo.

Identify the upstream branch to target

All published content resides in the *master* (source - already in production, or about to be there) and *live* (what's showing on production) branch. If you want to make edits to content that is already on the site, you should work off the *master* branch. However, sometimes, due to a number of reasons, you might have to work against a release branch created in your repo upstream. Release branches are mostly created to keep the master branch clean from day-to-day additions/updates that are not yet ready for production. Common reasons to use release branches are:

- New module development ([upstream release branch request form](#)).
- Development of embargoed or event content (need a place to submit content that is not ready to go live until a given date).

If you're unsure about which branch to use, talk to your content manager if you are required to work against a *release* branch or *master* branch.

Update your local repo with upstream

Before you start working on any new content, it is essential that you sync your local fork with the upstream master or release branch (whichever you are working against). This ensures that all changes made to the upstream remote branch by other users are downloaded and merged into your local copy.

1. Run the following command to sync your fork with the upstream master branch:

```
git pull upstream master
```

2. If you are working off a release branch, use the following command to sync your local fork with the upstream release branch:

```
git pull upstream <release-branch name>
```

Create a local branch

Whether you are working directly on master branch or on a release branch, you must always create a local branch to commit all your changes. When you are ready, create a pull request to submit your changes to either *master* or *release* upstream branches.

When you create a new local branch you must tell GitHub against which upstream branch you want it to be tracked to.

Run the following command to create a new local branch:

```
git checkout -B <new-branch-name> upstream/master or <release-branch-name>
```

Working with release branches

If you are working against the release branch, you first need to download the release branch to your local clone. Follow the steps listed below to work with a release branch.

1. Update your location clone with the new release branch information. Run the following commands:

```
git checkout master  
git fetch upstream
```

The fetch command lists the new branches with a label [new branch].

2. Run the following command to see a list of branches on your local machine:

```
git branch
```

Run the following command to see a list of remote branches:

```
git branch -r
```

3. Create a new local branch that tracks against the release branch.

```
git checkout -B <new-branch-name> upstream/<release-branch-name>
```

NOTE

The current active branch is listed in parenthesis with teal color in the Git bash prompt at any time. The default branch should be listed as (master) and changes to your local branch name when you use the `checkout` command. If there is no active branch shown when you navigate to the folder, you need to change directory into the repository sub-folder.

Add/update content in the local branch

When you add/update content, it is being done in your working directory, from where you need to first move it to the staging area and then commit them to your local repository.

After you finish add/update, follow the steps listed below to commit your changes:

1. Run the following command to add your changes to the staging area. `git add -A` moves all changes in the working directory to the staging area. If you want to move only specific files, use `git add <filename>`

```
git add -A
```

2. Run the following command to commit your changes. Always ensure that you provide a meaningful commit message. It makes it easier to identify the type of changes later.

```
git commit -m "<commit message>"
```

3. Run the following command to push your committed changes to your local repository. Because you are

working out of your fork, you must first push the changes to your fork (*origin*). From there, you need to create a pull request to push the changes upstream to *master* or *release* branch, as required.

```
git push origin <your-working-branch-name>
```

Create pull request from browser

When you push changes to your *origin* repository using the `git push origin` command, a URL is displayed for you to create a pull request. A **pull request (PR)** allows you to share your changes with others for discussion and review. Changes to the upstream repo can only be merged via pull request, so creating a PR is an essential step in getting your content in the upstream repo.

```
$ git push origin CG-Links-rework
Enumerating objects: 30, done.
Counting objects: 100% (30/30), done.
Delta compression using up to 4 threads.
Compressing objects: 100% (25/25), done.
Writing objects: 100% (25/25), 18.01 KiB | 1.20 MiB/s, done.
Total 25 (delta 13), reused 0 (delta 0)
remote: Resolving deltas: 100% (13/13), completed with 4 local objects.
remote: This repository moved. Please use the new location:
remote:   https://github.com/Kiranchandratrey/learn-docs.git
remote:
remote: Create a pull request for 'CG-links-rework' on GitHub by visiting:
remote:   https://github.com/Kiranchandratrey/learn-docs/pull/new/CG-Links-rework
remote:
To https://github.com/kiranchandratrey/learn-docs
 * [new branch]      CG-Links-rework -> CG-links-rework
```

Command run was "git push origin CG-links-rework" and output contains a pair of lines prefixed with "remote": the first line says "Create a pull request for 'CG-links-rework' on GitHub by visiting", and the second line is the github.com URL for the link that would start a pull request on GitHub.

After you create your PR, you can tag your team members for review and inputs. Once you are done with the changes, you submit your PR for the **PR review** process by commenting `#sign-off`. All Learn PRs must follow the **PR review guidelines**. See the **Pull request submission best practices** for more information.

Once you open a PR, you can keep making changes in your working directory and commit them to the opened PR. The pull request stays open until either it's *merged* or manually *closed*. You can make as many commits as you like to an open PR. All commits to a PR appear in a chronological order.

To create a new pull request:

1. Copy the url displayed after you ran `git push origin` command and paste it in a browser.
2. Verify that the correct upstream is selected.
3. Modify the PR title, if required. By default, the first commit message becomes the PR title.
4. Click **Create Pull request**.

Once you click the **Create pull request**, GitHub runs the validations configured for your repo and displays the build results on completion. If there are error or warnings, resolve them in your working directory and commit your changes again.

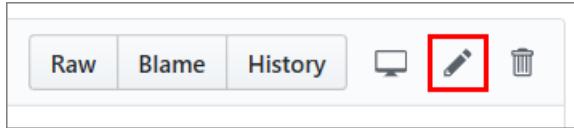
NOTE

You only need to click **Create pull request** button once when creating the PR the first time. For all the commits thereon, the build validations are run automatically as soon as you run the `git push origin` command.

Make quick edits from browser

The ideal way to edit your Learn content is to do it via local branch and then push upstream using pull request, as explained in the [Add/update content in the local branch](#) section. However, for minuscule changes such as one typo fix, or one formatting fix, you can make changes directly in GitHub via the browser.

1. Find the file (YAML or markdown) that you want to edit in your Learn repo and open it in the browser.
2. Click the **pencil icon** in the upper right.



3. Make your changes on the **Edit file** tab. Preview your changes on the **Preview changes** tab.
4. Scroll to the bottom of the page and click the green **Propose file change** button.
5. On the **Comparing changes** page, verify your changes and click **Create pull request** (at the top of the page) to submit your changes to the pull request queue.
6. Click the second **Create pull request** button to open the pull request.

The pull request will go through validation and say **All checks have passed** when the process is complete.

Track Learn content development in Azure Boards

1/14/2022 • 15 minutes to read

We track Learn content development in the Azure Boards service of our [Microsoft Learn Azure DevOps](#) project. There are hundreds of people (and growing!) contributing to Learn as a whole, via content development and supporting roles. The Learn Azure Boards are our holistic "source of truth" to provide the information that the different teams need to be successful:

- **Content teams** can see if there's other content that you could use in your learning paths and avoid extra effort or duplication of content.
- **Supporting partner teams**, like Design (achievement creation), Localization (DevRel, Microsoft 365, Dynamics, and so on), Labs, Instructional Design, and others can see the scope of what you're supporting for forecasting budget and resources.
- **Anyone** requesting work from the Learn team, like achievement images, instructional design reviews, lab setup, will have a closed feedback loop with Learn. They can easily check the status of their requests.

The Learn team has several standard queries to use to look for published content, in-progress content, or planned content. Here are a few of the most popular queries:

- [Published modules](#)
- [Completed last month/sprint](#)

Content teams are responsible for creating and updating their module and learning path work items up until the point of publication.

IMPORTANT

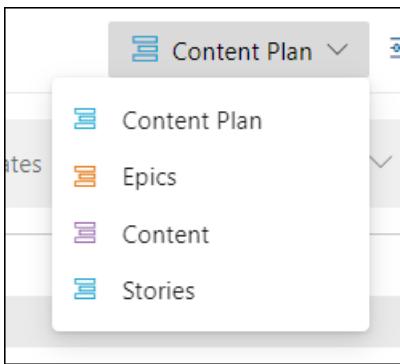
As soon as you start planning content and know roughly how many modules or learning paths you are going to create, start creating work items for them with as much information as you know. Update the work items as you finalize content.

If you can't access the [Learn Azure DevOps project](#), [request access to the CEAPEX DevOps board](#).

Planning content

Some teams prefer to create a comprehensive content plan that includes budgetary information before defining individual modules. We have a **Content Plan** work item type to collect the requirements and information into a single entry in the Azure Board. To create a content plan:

1. Go to the backlog appropriate for your content team:
 - Azure - [Azure Content](#)
 - Everyone else - [Partner Content](#)
2. In the upper right corner, select **Content Plan**. Don't select **Epics**, **Features**, or **Stories**.



3. Select **New Work Item** to create a new tracking work item.

The system should select **Content Plan** as the type automatically. If not, change it to **Content Plan**, then select **Add to top**.



4. Fill in the details in the presented form:

- The description should include overall information about the content you're creating.
- You're required to include the planned cost. It should be an estimate.
- You don't have to define modules at this level, but the expected module count is a required field.

5. Once you have all the details established, select **Save & Close**.

6. When you start defining modules for your plan, create relationships between the **Module** work items and the content plan. Use the **Related** link type. It's the most versatile approach.

Modules

Create **ModuleWorkItem** entries to represent *new* modules you plan to build. Create a single entry for each module and update the information in the work item as work progresses on the module. DevRel will automatically keep the Azure Boards representation synced with GitHub once you publish your content. Keep in mind that, while it's in-progress, you're responsible for all updates.

IMPORTANT

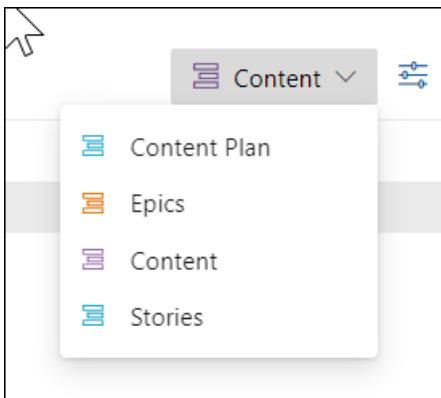
Do not create a new Module work item to represent an *update* to a published module. Instead, use a **Module Update** work item and associate it to the original work item with the **Related** link type. This lets us to maintain a history of prior work and versioning on modules in Azure Boards.

Create new module work items for new content

1. Go to the backlog appropriate for your content team:

- Azure - [Azure Content](#)
- Everyone else - [Partner Content](#)

2. In the upper right corner, select **Content**. Don't select **Epics**, **Content Plan**, or **Stories**.



3. Select **New Work Item**.
4. In the dialog box that appears, select **Module** from the drop-down list.

A screenshot of a 'New Work Item' dialog box. The 'Module' dropdown is open, showing several options: Feature, Learning Path, Module (selected and highlighted in grey), Module QA, and Module Update. The 'Module' option is currently selected. Other fields in the dialog include a search bar and an 'Add to top' button.

5. In the text box, enter the title of the module.

NOTE

It's ok if you don't know the exact title yet. Enter your best guess now and update it once you know more about it. For more information about good titles, see [How to write titles for learning paths, modules, and units](#).

6. Select **Add to Top**.
7. A new form will appear, prompting you to fill out the details for the module.

Description of the notable module work item fields

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Assignee	Optional - Highly recommended	Person creating the content, managing the content project, or point of contact for the content team.
Summary	Optional - Highly recommended	Summary of the module.
Design Notes	Optional	-
Learning Objectives	Optional - Highly recommended	Module learning goals. For more information about writing learning objectives, see How to write learning objectives .

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Prerequisites	Optional - Highly recommended, if known	Module prerequisites. For more information about writing prerequisites, see How to write prerequisites .
Justification	Optional - Highly recommended, if known	Justification for creating the module. Why do we need it on Learn?
Tech Reviewers	Optional - Highly recommended, if known	People slated for and capable of doing technical (content accuracy validation) reviews for this module.
Level	Required	Select a level of the module from the drop-down list: Beginner, Intermediate, Advanced.
Role	Required	Audiences that you're targeting this module for. Select from the drop-down list. You can choose more than one. If the audience that you need isn't listed, select Other and reach out to your content manager or Learn team point of contact for assistance. Role is a required field in the content itself and uses an allowlist. If we need to add an extra value, we'll need to work with Engineering to get it implemented.
Lab Enabled	Optional - Highly recommended	Toggle to True if you plan to implement an interactive lab with the module. (NOTE: <i>We have a limited set of interactive lab options at this time, so it only applies to Azure content.</i>)
Lab Type	Only required if Lab Enabled is True .	Select the type of lab you plan to implement or have implemented for the module from a pick list. If you don't know what type you need, select Unsure .
Vendor	Optional	If you're using a managed service vendor team, add their information here.
Content Team	Required	Select the name of your team. Content teams and supporting partner teams use the Content Team option to filter queries for their applicable content.
GitHub Repository	Optional - Highly recommended	Repository that the module will live in. If you're unsure - leave it blank or ask your content manager.

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Module UID	Optional - Highly recommended	UID for the module. The Module UID option is important. It helps us automate the updating of this work item for you. If you fill in the UID here and don't change it in the content, it's okay. We have tools to automate keeping the work item and the content synced. The title, summary, learning objectives, and so on, in this work item will remain synced with the content once it's published. You won't have to update it in two places.
Target Release Date	Required	Date that you're planning to release the module.
Published Date	Optional	Date that the module published to the live site. (NOTE: <i>We have tools to update it if you don't do it at the time of pushing your PR to the main branch.</i>)
Design Review Completed	Optional	Used to help identify if you've done a design review for the module. Using this option can help you query the status for this module across your content roadmap.
Technical Review Completed	Optional	Used to help identify if you've done a technical review for the module. Using this option can help you query the status for this module across your content roadmap.
Editorial Review Completed	Optional	Used to help identify if you've done an editorial review for the module. Using this option can help you query the status for this module across your content roadmap.
Related Work	Optional - Highly recommended if part of a learning path	Can link other work items to this one: achievement requests, video requests, learning paths, and so on. If the module is a part of one or more learning paths, it's highly recommended to link it to the learning path(s) so that we can see the higher-level plan. For information on how to link/relate work items, see the Link work items together section in this article, below.

8. Update the module state:

- **New** - Proposed item. If it hasn't been approved, you can also add a "Proposal" tag to the item.
- **Committed** - You're planning to do it, but haven't started the effort.

- **In Review** - In QA review phase.
- **In Progress** - In content development.
- **Blocked** - Content development is blocked.
- **Closed** - Module published to the live site.
- **Declined** - Decided not to move ahead with the proposed module.
- **Removed** - Module was published to the live site then deprecated.

9. Select **Save & Close**.

Update existing module work items

IMPORTANT

The instructions below are for tracking module creation progress while the content is being developed. If you are updating an *existing* published module, use the instructions in the next section.

1. Find the module work item that you want to update. There are multiple options in Azure Boards, but the most common are queries and backlogs:

Queries

- [Link to Learn Shared Queries](#)
- [Create your own query](#)

Backlogs

- Azure - [Azure Content](#)
- Everyone else - [Partner Content](#)

2. Select the module work item to open it.

3. Update the fields.

4. Select **Save and Close**.

Create a work item to update an existing published module

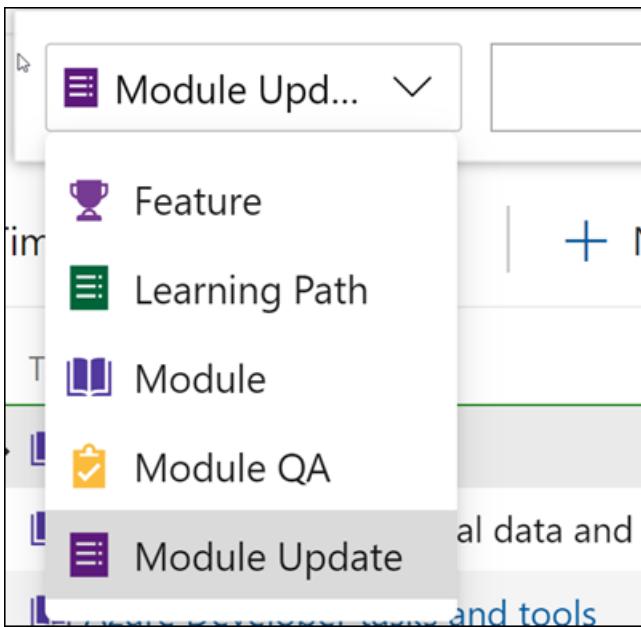
1. Go to the backlog appropriate for your content team:

- Azure - [Azure Content](#)
- Everyone else - [Partner Content](#)

2. In the upper right corner, select **Content**. Don't select **Epics**, **Content Plan**, or **Stories**.

3. Select **New Work Item**.

4. In the dialog box that appears, select **Module Update** from the drop-down list.



5. Fill in the Module GitHub repo and UID information from the existing, published module.
6. Add the details related to the change you're making. Most of these fields are the same or similar to creating a new module.
7. Select the **Content team** that owns the published module.
8. Add a link relationship to the existing **Module** work item that represents the published content. Use the **Related** link type.
9. Once you're done filling in the details, select **Save & Close**.

Learning Paths

Create new learning path work items

1. Navigate to the backlog appropriate for your content team:
 - Azure - [Azure Content](#)
 - Everyone else - [Partner Content](#)
2. In the upper right corner, select **Content**. Don't select **Epics**, **Content Plan**, or **Stories**.
3. Select **New Work Item**.
4. In the dialog box that appears, select **Learning Path** from the drop-down list.
5. In the text box, enter the title of the learning path.

NOTE

It's ok if you don't know the exact title yet. Enter your best guess now and update it once you know more about it. For more information about good titles, see [How to write titles for learning paths, modules, and units](#).

6. Select **Add to Top**.
7. A new form will appear, prompting you to fill out the details for the learning path.

Description of the learning path work item fields

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Assignee	Optional - Highly recommended	Person creating the content, managing the content project, or point of contact for the content team.
Summary	Optional - Highly recommended	Summary of the learning path.
Design Notes	Optional	-
Prerequisites	Optional - Highly recommended, if known	Learning path prerequisites. For more information about writing prerequisites, see How to write prerequisites .
Level	Required	Level of the module: Beginner, Intermediate, Advanced.
Role	Required	Audiences that you're targeting this module for. Select from the drop-down list. You can choose more than one. If the audience that you need isn't listed, select Other and reach out to your content manager or Learn team point of contact for assistance. Role is a required field in the content itself and uses an allowlist. If we need to add an extra value, we'll need to work with Engineering to get it implemented.
Content Team	Required	Select the name of your team. Content teams and supporting partner teams use the Content Team option to filter queries for their applicable content.
GitHub Repository	Optional - Highly recommended	Repository that the learning path will live in. If you're unsure - leave it blank or ask your content manager.
Module UID	Optional - Highly recommended	UID for the learning path. The Module UID option is important. It helps us automate the updating of this work item for you. If you fill in the UID here and don't change it in the content, it's okay. We have tools to automate keeping the work item and the content synced. The title, summary, learning objectives, and so on, in this work item will remain synced with the content once it's published. You won't have to update in two places.
Target Release Date	Required	Date that you're planning to release the learning path.

FIELD	REQUIRED/OPTIONAL	DESCRIPTION
Published Date	Optional	Date that the learning path published to the live site. (NOTE: <i>We have tools to update it if you don't do it when you push your PR to the main branch.</i>)
Related Work	Optional - Highly recommended	Can relate other work items to it (achievement requests, modules, and so on). For information on how to link related work items, see the Link work items together section in this article, below.

8. Select the drop-down arrow next to **Save and Close**.

NOTE

If you accidentally select **Save and Close**, it will save the work item in the new state. If you need to update the state, you can open up the work item again, update the state, and select **Save** or **Save and Close** again.

9. Update the learning path state:

- **New** - Proposed item.
- **Committed** - You're planning to do it, but haven't started the effort.
- **Published** - Learning path published to the live site.
- **In Progress** - In content development.
- **Blocked** - Content development is blocked.
- **In Review** - In QA review.
- **Rejected** - Decided not to move ahead with the proposed learning path.
- **Removed** - Learning path published to the live site then deprecated.

10. Select **Save**.

Update existing learning path work items

1. Find the learning path work item that you want to update. There are multiple options in Azure Boards, but the most common are queries and backlogs:

Queries

- [Link to Learn Shared Queries](#)
- [Create your own query](#)

Backlogs

- Azure - [Azure Content](#)
- Everyone else - [Partner Content](#)

2. Select the learning path work item to open it.

3. Update the fields.

4. Select **Save and Close**.

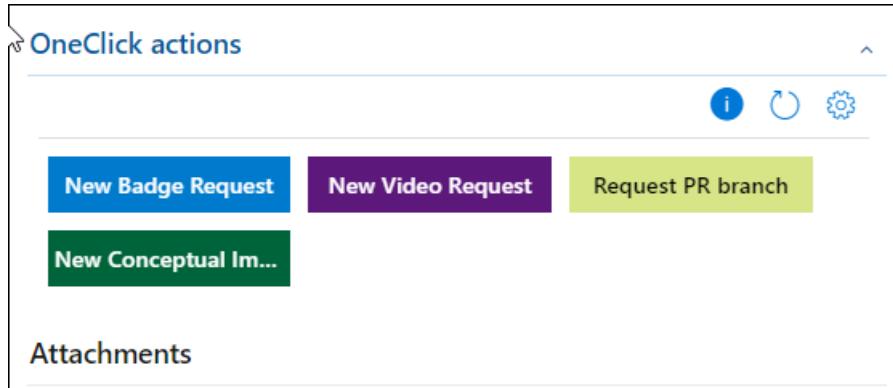
Other required work items

Create the following required work items at the same time you create a new module/learning path work item.

Badge or trophy

For a new module, you'll need to request a badge, and a new learning path needs a new trophy. Since the badge-design team is heavily backlogged and it may take some time to get your badge/trophy artwork completed, put in these requests as soon as you have the required details about the module.

Use the OneClick action within the module work item to open a badge request.



Release branch

Release branches are required for each new Learn module. Create the request as soon as you have finalized the module name and the repo location where the module should reside.

To request a new release branch, file a ticket with the [Content & Learning content production team](#). Provide the following details in the ticket:

- Check the boxes for **Microsoft Learn upstream module branch** and the sub-item for the desired Learn repo.
- Next to the Learn repo you checked, add the desired branch name prefixed with "NEW-" for new modules.
- Add any relevant aliases to the **FTE contacts** section.
- Set the **Requesting Group** to an appropriate value.

Conceptual artwork

For complex conceptual artwork, you can raise a request with the Content & Learning graphics artist (v-tstubbss@microsoft.com). Use the OneClick option within the module work item to open a request for a new conceptual image. Add the details of the required artwork and links to any existing images to help the artist. If you are creating this request on a contributor's behalf, tag them in the work item so they get notifications when the work item is updated.

Link work items together

There are a couple of ways to link work items together, depending on whether the work item exists already or not. Each is helpful for different scenarios.

If you created a one-off and forgot to link it, you may prefer the first option. Maybe you created a bunch that you need to sort out. You can use the second option. Are you the type of person who's organized ahead of time? You'd probably prefer option number three.

Link one work item to another

The first option for existing items is to link one work item to another.

1. Select the **work item** you want to relate another one to (could be Module or a Learning Path).
2. To the far right of the item, there's a section called **Related Work**. Select **Add link**.
3. Select **Existing Item**.

4. Follow prompts to link as a related item (**Link type = Related**).

NOTE

We don't link parent/child module and learning path work items since modules may appear in multiple learning paths.

5. Select **Save and Close**.

Update many work items to link to the same item

The second option for existing items is to update many work items to link them to the same item at the same time.

1. In the backlog view, **Ctrl+click** the items that you want to link. For example, select five module work items to link them to the same learning path work item.
2. **Right-click** the items you want to link.
3. Select **Add parent**.
4. Select **Existing Item**.
5. Follow prompts to link as a related item (**Link type = Related**).

NOTE

We don't link parent/child module and learning path work items since modules may appear in multiple learning paths.

6. Select **Save and Close**.

Creating new linked items associated with an existing work item

The third and final option is to create new linked items associated with an existing work item.

1. Select the work item you want link new items to.
2. To the far right of the item, there's a section called **Related Work**. Select **Add link**.
3. Select **New Item**.
4. Follow prompts to link as a related item (**Link type = Related**).

NOTE

We don't link parent/child module and learning path work items since modules may appear in multiple learning paths.

5. Select **Save and Close**.

Bulk creation tools

View the [Bulk add or modify work items with Excel](#) article on Docs. The process isn't simple, but we're constantly improving it. The process can save you significant time if you're creating more than 10-20 items.

Content queries

See the [View Learn content portfolio and roadmap](#) article.

Create compliant content

1/14/2022 • 3 minutes to read

All Microsoft Learn content must meet organizational level and Learn-specific compliance requirements. Any non-compliant content will be stopped from publishing. This article gives you an overview of compliance requirements for creating Learn content.

Your Learn content must be in compliance with the following to be eligible for publishing:

- [Organizational compliance requirements](#)
- [Style guides](#)
- [Learn-specific compliance requirements](#)

Organizational compliance requirements

- **Accessibility:** All Learn videos must be recorded in accordance to the [Microsoft Accessibility Standards \(MAS\)](#) and other accessibility guidelines on the [Enable website](#).
- **Closed captions:** All Learn videos must publish with closed captions (CCs); no exceptions.
- **Global readiness:** All Learn content must be appropriate for a global audience per the [Global Readiness](#) guidelines.
- **Licensing, third-party intellectual property (IP) and content:** All Learn content must meet the requirements under [Using Third-Party Content](#) and [copyright policy](#) on the [CELA](#) (Corporate, External, and Legal Affairs) website. IP includes:
 - Videos and text
 - Screenshots, iFrames, icons, photos, and artwork
 - Embedded materials
 - Names and quotes
 - Music
 - Studies, research, statistics, surveys, or software
- **Graphics:** Verify that your graphic images are compliant with [Microsoft Guidelines for Graphics](#), including the requirements for maps, flags, and symbols.
- **Branding:** All usage of Microsoft brands must adhere to the specifications on [Brand Central](#). Take care that you don't use brand names in an unapproved fashion, such as by combining two brand names together. If you're listing multiple brands, order them alphabetically.
- **PoliCheck:** PoliCheck must be run on all Learn content, no exceptions. Use your CorpNet credentials to download and run [PoliCheck](#).
- **Fictitious names:** Use CELA-approved fictitious names for people, companies, fictitious email addresses, and URLs that are found on [CELAWeb](#).
- **Code:** Ensure that all code owned by Microsoft is code signed. Also ensure that all open source or third-party code has been run through the Open Source Software (OSS) tool. For more information, see Microsoft's Open Source Code usage policy: [Open Source Software Approval Process](#).
- **Hidden data:** Remove all hidden data and personal information from text documents that you might have used as resources for your Learn module(including PowerPoint, Word, Excel, and PDF files). For

more information, see [Remove hidden data and personal information by inspecting documents](#).

- **General Data Protection Regulation (GDPR) references:** 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. 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. For detailed guidelines on including GDPR reference, see the [GDPR references in new or existing technical documentation articles](#) article in the Docs Contributor guide.

Style guides

There are several organizational-level style guides that all content on docs.microsoft.com must adhere to. For a list of style guides and the order to follow them in, refer to the [style guide hierarchy](#) article.

Also, your product might be required to follow a product-specific style guide. Refer to a [complete list](#) of style guides to see if there's one that is specific to your product/group. If you're unsure, contact the Learn CX team.

Learn-specific compliance requirements

Since Microsoft Learn provides a unique learner experience, there are certain guidelines that must be adhered to for a content to be eligible for Learn platform. Non-compliance of these guidelines may lead to your content being rejected for publishing.

Content requirements

Microsoft Learn hosts a vast range of content written by content authors from different teams across the organization. Learn aims at providing a consistent learner experience. To provide a consistent experience, we have defined few basic requirements that every content must meet before it can be published on Learn.

The requirements are divided into the following three categories:

- **Priority 0** - Content must meet all of these Priority 0 requirements before initial publication or it cannot be hosted on Learn. Content must also meet the Priority 1 requirements within 30 days of initial publication.
- **Priority 1** - Content must meet these requirements within 30 days of initial publication.
- **Priority 2** - We expect most content to follow these rules.

For detailed requirements, see the [Content requirement](#) article.

Instructional design guidelines

Learn has defined a set of guidelines to enable content authors to create effective and engaging learning content. The content helps learners achieve clearly defined goals and gain new skills. See our [Authoring guidelines](#) for details.

Video guidelines

Along with the organizational-level accessibility guidelines for videos, the Learn video team has defined several **Requirements and Best practices** for including videos in a Learn module. See the [Guidelines for adding video in Microsoft Learn](#) for details.

Create a Microsoft Learn module

1/14/2022 • 4 minutes to read

In this article you will:

- Learn about module content guidelines and how to appropriately leverage the modules feature
- Learn about the module folder structure and file types

Overview of Microsoft Learn training modules

Modules are the building blocks of the Microsoft Learn experience. A module contains a collection of related units that teach a concept using textual content, videos, and labs. They are:

- Atomic chunks of training, like a course.
- The smallest reusable piece in the Microsoft Learn content model and can be shared across multiple learning paths.

Learners should have gained a new skill upon completion of a Microsoft Learn module. Learners may choose any modules to take, no matter the order. Also, Microsoft Learn lets content developers string together modules into a recommended learning path. This structure helps guide learners who may not even know where to start or what they're looking for. Modules can be a part of a learning path or can be standalone, "specialty" content.

Example: A live example of a Microsoft Learn training module can be found [here](#). The source files are in the [learn-pr repo](#).

We have well-documented guidelines on how to write and structure different types of modules. For more information, see the following articles:

- [Standard modules](#)
- [Introduction modules](#)
- [Choose modules](#)

Implementation: Scaffold the module files

Modules are a package of files (YAML, MD, and others) that drive the content that the learner interacts with. When you scaffold a module, you create all of the files and folders that are required for the module to successfully build. There may be no actual learning content in it yet, but the structure, or shell, is built. This activity is done once before you submit your first pull request (PR).

Example: A module that has 5 units has at least 3 folders and 11 files in the total package.

- **Module folder:** Folder that contains almost all of the module's associated files.
- **Module 'includes' folder:** Folder that contains the core learning content markdown files.
- **Module 'media' folder:** Folder that contains all of the image files in the learning content.
- **Module index YAML file (`index.yml`):** A singular YAML file that defines the module metadata and the display of units.
- **Unit YAML files (`unit-name.yml`):** Multiple YAML files (1 per unit) that define the metadata, content, interactivity, and validation method of units.
- **Unit content markdown files (`unit-name.md`):** Multiple markdown files (1 per unit) that contain the actual learning content of the unit, except knowledge check questions. These content files are added as `includes` references in the unit files.

- **Media files:** Any images or videos included in the unit content files. These media files are added as `includes` references in the content files.

You can scaffold a module manually or using a template.

- For manual scaffolding, refer to the [Scaffold a module manually](#) article.
- For template scaffolding, select [Scaffold a module using template](#) article.

Implementation: Add an achievement to the module

There are several unique features of Microsoft Learn modules. One of those features is that learners can earn achievement badges upon completion of the modules and learning paths. These achievements show up on their Docs user profile and can be shared via social media and other channels.

To get the module to successfully build, you'll have to add an achievement to the content. For most projects, you'll have to use a placeholder image until the completed badge artwork is ready. You can't launch with placeholder image because they'll block your launch timeline. As such, you should request to have achievement artwork created at the beginning of the content development project. Then, the images can be ready by the time you're done developing and ready to launch. View the [Create an achievement](#) article for details about requesting achievement artwork.

1. In the module `index.yml` file, update the `iconURL` parameter to include the following value:

If you do NOT have the final achievement image:

```
iconUrl: http://via.placeholder.com/120x120
```

If you DO have the achievement image:

```
iconUrl: /learn/achievements/<module-folder-name>.svg
```

2. Update the badge uid to be globally unique across all Learn repos. The easiest way to make it unique and easy to manage is to use the module UID, and then append `.badge` to the end.

```
badge:  
  uid: <module-uid>.badge
```

TIP

Module achievements are always **badges**. If you try to create a trophy for a module, it will throw a build error.

IMPORTANT

The `achievements.yml` file still exists but is only for deprecated modules and learning paths. If a module or a learning path is not deprecated, there shouldn't be any associated achievement information in the `achievement.yml` file.

IMPORTANT

Do NOT change the uid for an achievement after it launches. This will deprecate the old achievement and cause a new achievement to be created in its place. This is a poor learner experience, as the users who earned the old one will not have credit for the new one.

Engineering Documentation

<https://review.docs.microsoft.com/new-hope/specs/triple-crown/modules-ignite>

<https://review.docs.microsoft.com/new-hope/specs/learn/achievement-authoring>

Related Documentation

[Unified content model](#)

Scaffold a module manually

1/14/2022 • 6 minutes to read

In this article, you will:

- Learn how to manually scaffold a module
- Learn about module metadata

Step 1: Create the module folder

Each module has its own folder under the relevant product folder in the repo. For example, an Azure-related module will have a folder under the `azure` folder in the `learn-pr` repo. The folder name drives the URL for the module under the base Learn URL.

Sample folder name

`secure-your-cloud-data`

Corresponding URL

<https://docs.microsoft.com/learn/modules/secure-your-cloud-data/>

Tips for naming the module folder

- Be specific with the folder name.
- Don't use product names in the folder name where it's obvious that it's a Microsoft product.
- If it's not obvious, make sure to include it. For example, `intro-to-security` should be `intro-to-azure-security`.
- Don't use the product brand name, if possible. Acceptable times to use the product in the name is when the name would conflict with another Microsoft product, such as `create-and-share-your-first-report`. There are multiple Microsoft products that allow you to name a module like this, so it should be more specific, like `create-and-share-your-first-power-bi-report`.
- Find a balance when using connecting words such as "and", "the", "a", and so on. There's no strict rule about including them or not, but remember learners will see the folder name. Also, the folder name will help you find your content in a large set of content. Many content developers find it helpful to make it a bit friendlier so that they can refer to the module by the folder name.
- Use lowercase only when naming a folder and hyphen '-' *not* spaces as separator. This guidance also applies when using product names.

Step 2: Create the `includes` folder

Each module folder contains a folder called `includes` that will store the Markdown files for all the learning content in the module.

Step 3: Create the module index file

The module index file is a YAML file that defines the metadata and structure for module. It's stored under the module folder.

Sample module index YAML file code

```

### YamlMime:Module
uid: learn.align-requirements-in-azure
metadata:
  title: Align requirements with cloud types and service models in Azure
  description: In this module, we'll discuss how to align requirements with cloud types and service models in Azure.
  ms.date: 09/24/2018
  author: markjulmar
  ms.author: smmark
  ms.topic: interactive-tutorial
  ms.prod: learning-azure
title: Align requirements with cloud types and service models in Azure
summary: >
  Azure supports three approaches to deploying cloud resources - public, private, and the hybrid cloud. Selecting between them will change several factors of the services you move into Azure including cost, maintenance requirements, and security. In this module, we will look at all three types and help you make an informed decision about which one to leverage for your services. We'll also quickly look at the service models supported by Azure which can help you determine the services you should start with when planning out an Azure deployment.
abstract: |
  In this module, you will:
  - Compare each cloud computing deployment model (public, private, and hybrid).
  - Understand the advantages of each cloud computing service model.
  - Decide which deployment and service model you should use for your services.
prerequisites: None
iconUrl: /learn/achievements/align-requirements-in-azure.svg
ratingEnabled: false
levels:
- beginner
roles:
- solutions-architect
products:
- Azure
units:
- learn.align-requirements-in-azure.1-introduction
- learn.align-requirements-in-azure.2-public-private-hybrid
- learn.align-requirements-in-azure.3-service-models
- learn.align-requirements-in-azure.4-iaas
- learn.align-requirements-in-azure.5-paas
- learn.align-requirements-in-azure.6-saas
- learn.align-requirements-in-azure.7-quiz
- learn.align-requirements-in-azure.8-summary
badge:
  uid: learn.align-requirements-in-azure.badge

```

Explanation of metadata

All metadata is required unless explicitly marked as optional. During scaffolding, populate as much information as you can.

If you don't have all of the following values in your module, your module will fail to publish or one of the following problems will occur:

- Reporting won't be precise.
- GitHub issues may not route appropriately.
- The modules may not publish correctly.
- RSS feeds or site search may not show your content as expected.
- Your modules may run worse for search engines.
- You may experience push back in PR reviews.

When you attribute your content correctly, site features will run as expected and you'll get to make the most of the built-in reports.

FIELD	VALUE	WHY?
<code>### YamlMime:</code>	Module	Specifies to the Docs platform the type of YAML file and metadata to expect. Must be the first line of the YAML file.
<code>uid</code>	Manually generated unique ID that follows the format: <code>(repo-name).(module-short-name)</code> . Use lowercase only, even for the product names.	This value is a unique ID to differentiate modules and reference them when building learning paths. For more information, see the Overview of uid article.
<code>title</code>	Module title	The title of the module that appears on the module card and landing page. Use sentence case only, except for the product name, which should be capitalized.
<code>summary</code>	A brief summary describing what the module is about.	Appears on the main module landing page.
<code>metadata</code>	Page-level metadata	These properties are helpful in search engine optimization. Review the SEO Basics article in the Contributors Guide for more information.
<code>metadata > title</code>	Module title	This value is the most important metadata for SEO. If possible use unbranded terms to help customers find the content and improve SEO.
<code>metadata > description</code>	A summary of the module	Used in site search. Sometimes used on a search engine results page for improved SEO.
<code>metadata > ms.date</code>	A date in the format MM/DD/YYYY.	Displayed on the published page to indicate the last time the module was substantially edited or guaranteed fresh. If this field is missing, the date of the last commit is displayed instead, which may be incorrect for freshness.
<code>metadata > author</code>	The author's GitHub alias.	Identifies the author by GitHub ID in case there are questions about or problems with the content. In some cases, the author might be notified via GitHub automation of activity involving the file.
<code>metadata > ms.author</code>	The author's Microsoft alias, without <code>@microsoft.com</code> .	Used for content reporting and BI.
<code>metadata > ms.topic</code>	Type of article, like <code>interactive-tutorial</code> .	The type of content.

FIELD	VALUE	WHY?
<code>metadata > ms.prod</code>	The type of product. Should be one of these values: <code>learning-azure</code> , <code>learning-powerapps</code> , <code>learning-powerbi</code> , <code>learning-flow</code> , <code>learning-d365</code> , <code>learning-<product></code> .	Used for issue triage and reporting.
<code>abstract</code>	Module learning goals	This value appears on the module tile. The abstract allows learners to make an informed decision about whether to go ahead with the module or not. It should follow the format "In this module, you will:" and then a bulleted list of action-oriented learning goals.
<code>prerequisites</code>	Bulleted list of module prerequisites or the word <code>None</code>	This value appears on the module tile so that learners can make an informed decision about taking the specific module.
<code>iconUrl</code>	The link to the module image shown on the module landing page.	This value is the same image as the badge/trophy awarded for completing the module. You won't have this value during the scaffolding stage, so we typically use a placeholder URL <code>http://via.placeholder.com/120x120</code>
<code>ratingEnabled</code>	By default, enter false. Only modules that are part of the module ratings feature need to have this set to true.	
<code>level</code> , <code>role</code> , <code>product</code>	Must be one or more of these values .	This value appears on the module tile to help learners decide whether the module is suitable for them or not. The value also drives discoverability through the Learn browse page filters.
<code>units</code>	Unique unit uids in the format <code>(module uid).(unit-short-name)</code> . Use lowercase only.	A list of module units in the order they should be displayed to the user.
<code>badge > uid</code>	A globally unique UID for the achievement badge in the format <code>(module uid).badge</code>	This badge appears on successful completion of the module.

Step 4: Create unit content files

All the content units need a YAML file and a Markdown file. Create the unit YAML files in the `module` folder and the Markdown files in the `includes` folder. The Markdown files will be blank at this stage. For file names, use lowercase with hyphen '-' separators.

For more information about how to scaffold unit content files, see the [Create a unit](#) article.

For more information about writing unit content, see the [Write learning content](#) article.

Scaffold a module using the Docs Scaffolding extension for VS Code

1/14/2022 • 5 minutes to read

In this article you will:

- Learn how to scaffold a module based on a standard pattern using the Docs Scaffolding extension
- Learn how to update the structure of a scaffolded module

For details about the different elements of module scaffolding, make yourself familiar with the file information provided in [Scaffold manually](#) and [Create a unit](#).

The Docs Authoring Pack for VS Code includes the Docs Scaffolding extension, which provides functionality to auto-generate (*scaffold*) a skeleton module based on a standard pattern defined by the Learn team. You can install the full Authoring Pack by clicking [here](#) or just Docs Scaffolding by clicking [here](#).

The scaffolding extension supports the following module patterns:

- [Standard](#)
- [Introduction](#)
- [Choose](#)

When you run the command to create a module and choose the pattern, the following are generated automatically:

1. A module folder in the selected parent folder.
2. An index.yml file for your module, with all the default metadata updated.
3. One YAML and one Markdown file for each unit in the module. These files contain template text to help author your module according to Learn best practices.
4. A media folder to store your module images and media files. This folder is empty during scaffolding.
5. An includes folder populated with a Markdown file for each unit you create.

NOTE

The scaffolding files are created with default values or blanks. Update all placeholder text in the files before submitting your PR. If you don't, your build will fail or your module will be incorrect.

Customize template settings

You can customize the template settings to provide the default values for some of the metadata for your module. If you don't set these defaults, modules will be scaffolded with default values or blanks for these, which can be manually updated later after the scaffolding files are built.

To customize template settings:

1. In VS Code, navigate to File > Preference > Settings.
2. In the User settings section, go to Extensions > Docs Scaffolding Extension Configuration.

Docs Scaffolding Extension Configuration

Docs > Scaffolding: Alias

Microsoft alias

Docs > Scaffolding: Githubid

GitHub ID

Docs > Scaffolding: Prefix

Default prefix e.g. learn

Docs > Scaffolding: Product

Default product e.g. azure

Docs > Scaffolding: Template_repo

Learn template repo

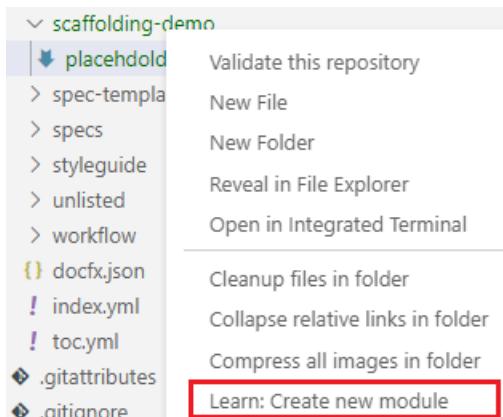
Set the following values:

- **Microsoft alias:** Enter your Microsoft alias without "@microsoft.com". This value is used for the `ms.author` metadata in index and unit YAML files.
- **GitHub ID:** Enter your GitHub alias. This is used for the `author` metadata. In some cases, the author might be notified via GitHub automation of activity involving the file.
- **Default prefix:** Defines the default prefix for the module and unit UID. This should map to the repo that contains your module. For example, for modules hosted in learn-pr enter "learn".
- **Default product:** Enter one or more valid value from the `Product` taxonomy to populate the `products` metadata. Use a comma-separated list if you specify more than one product.

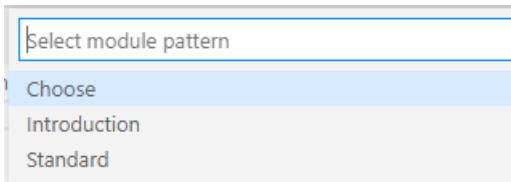
Select the appropriate module pattern

1. Right-click the parent folder where you want to add a new module.

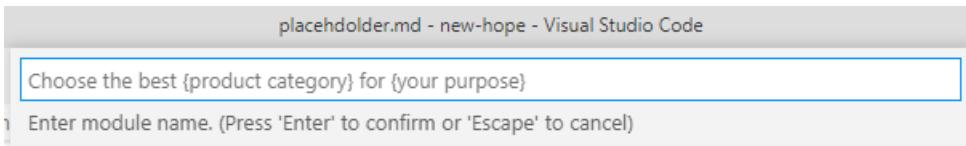
2. Select **Learn: Create new module**.



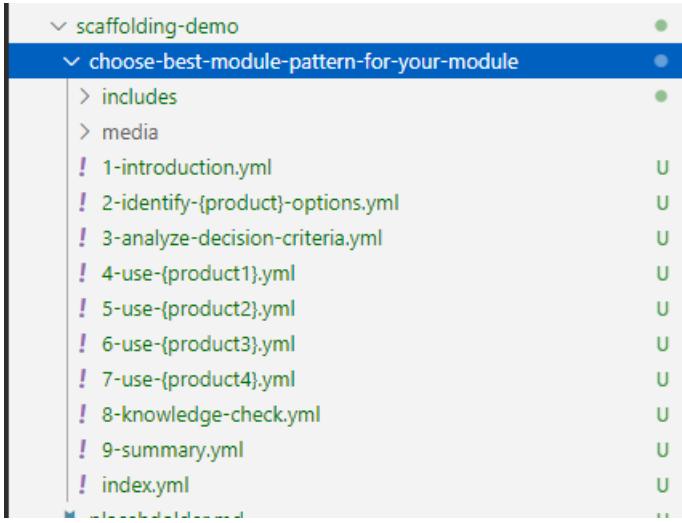
3. Select the appropriate pattern out of Standard, Introduction, and Choose.



4. Type a title for the module, using the placeholder text as a prompt.



5. The module is generated. Here's an example of a newly scaffolded choose module:



Before you begin authoring your module, browse the autogenerated files. You'll notice that some of the file names and UIDs contain placeholders you'll need to fill in, such as "{product}". Don't update these manually - the extension provides rename functionality, which we'll describe below.

You'll also notice that the YAML files stub some content and metadata based on your settings and on the type of module or unit, but some YAML fields contain placeholders you'll need to fill in. Placeholders you need to manually update are marked with hashes, as follows:

```

! index.yml ×
new-hope > scaffolding-demo > choose-best-module-pattern-for-your-module > ! index.yml > ...
1  ### YamIMime:Module
2  uid: learn.choose-the-best-module-pattern-for-your-module
3  metadata:
4    type: choose
5    title: # user input: module title for browser tab and search results
6    description: # user input: a description for site search and SEO
7    ms.date: 2/24/2021
8    author: meganbradley
9    ms.author: mbradley
10   ms.topic: interactive-tutorial
11   title: Choose the best module pattern for your module
12   summary: # user input: describe the module contents
13   abstract: | # user input: list the learning objectives
14     In this module, you will:
15     - # objective 1
16     - # objective 2
17     - # objective 3 (if needed)
18   prerequisites: | # user input: list the prerequisites
19     - # prerequisite 1
20     - # prerequisite 2
21     - # prerequisite 3
22   iconUrl: https://docs.microsoft.com/media/learn/module.svg
23   levels: # user input: add levels from level taxonomy (https://review.docs.microsoft.com/en-us/help/contribute/metadata-taxonomies?branch=master#level)
24     - # level
25   roles: # user input: add roles from role taxonomy (https://review.docs.microsoft.com/en-us/help/contribute/metadata-taxonomies?branch=master#role)
26     - # role 1
27     - # role 2
28   products:
29     - azure
30     - sql
31   units:
32     - learn.choose-the-best-module-pattern-for-your-module.introduction
33     - learn.choose-the-best-module-pattern-for-your-module.identify-{product}
           -options

```

In addition, the included Markdown files contain extensive guidance to help author each unit according to Learn best practices.

Update the metadata and be mindful of SEO

First, remove all comments from every placeholder metadata generated by the extension. Otherwise, your module will not be a valid module to be published. After this removal, update all placeholders to your titles and descriptions.

Be especially mindful about the `metadata.title`, `title` and `metadata.description` fields, as these are used to help users find your module in the MS Learn website. Best practices are:

- `title` and `metadata.title` should be the same.
- Both titles and `metadata.description` should follow the [SEO best practices](#) for a module.

The description field is the place where you should optimize your SEO. Be sure to include in it all the keywords and phrases the audience might use to search for your module.

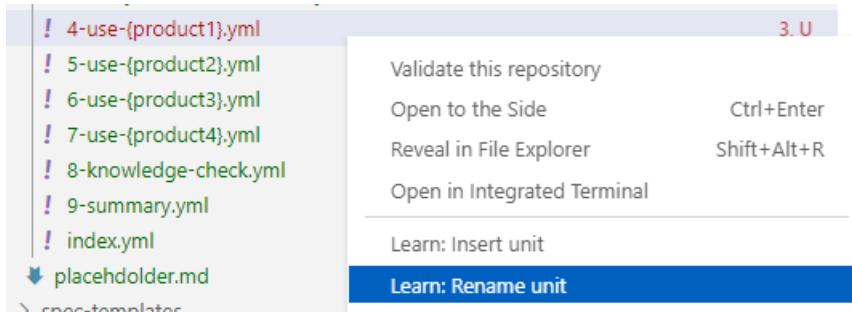
TIP

If still in doubt about how to best use SEO for your module, see [this tutorial on how to best optimize your SEO](#).

Update placeholder text

The Docs Scaffolding extension provides functionality to update file names, file references, and UIDs all at once. This means that if you update a unit YAML file name, you don't have to also update the corresponding Markdown file name, the INCLUDE reference, and the UID. Before you begin authoring your module, use the rename functionality to update placeholders, as follows:

1. For each unit you need to update, right-click the YAML file and select **Learn: Rename unit**.



2. Type the new file name, without a prefix or file name extension, in the pop-up box. For example, you might replace "use-{product1}" with "use-choose-module" if you're writing a module about how to choose between the standard module patterns. Press Enter.
3. In the next pop-up box, optionally update the unit title. For example, change "Use {product} to {solve problem}" to "Use the Choose module pattern to compare product choices". Press Enter.
4. The following changes are made automatically:

- The YAML file name is updated. For example, "4-use-{product1}.yml" becomes "4-use-choose-module.yml".
- The corresponding Markdown file name is updated, such as "4-use-choose-module.md".
- The INCLUDE in the `content` field of the YAML file is updated, such as
`[!include[]](includes/4-use-choose-module.md)`.
- The `uid` in the unit YAML file and in index.yml is updated, such as
`learn.choose-the-best-module-pattern-for-your-module.use-choose-module`.

NOTE

The UIDs are only updated if the unit hasn't yet been published live. This means you can automatically update UIDs with placeholders because they haven't been published yet, but you can rename published units without changing the UID, which isn't allowed.

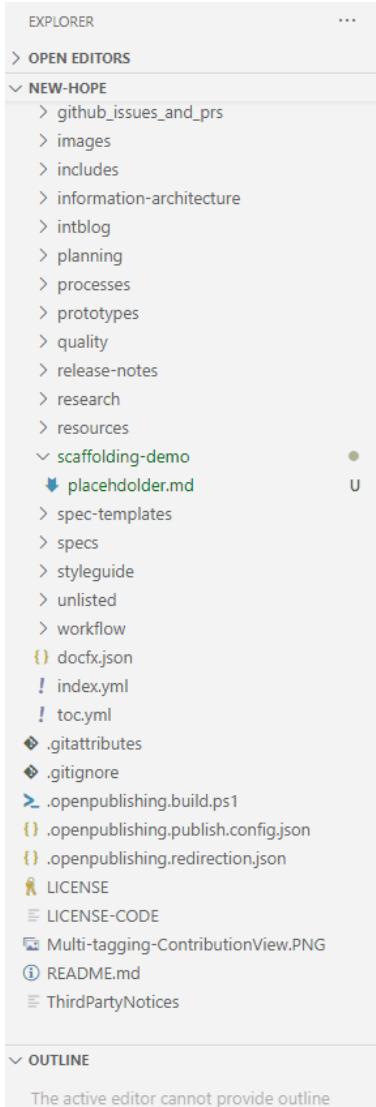
Add, remove, or reorder units

The extension also provides functionality to add, remove, and reorder units. Like the rename feature, you must initiate all structural updates in the unit YAML file. You can insert a new unit, delete an existing unit, or move an existing unit up or down. The YAML file names are updated to reflect the new order, and the Markdown file names and INCLUDE references are updated to match the new YAML file names. The UIDs are reordered in index.yml to reflect the new order.

For more information about these additional functions, see the extension readme.

Scaffolding demo

The following animated GIF shows the basics of creating a new module in a pattern, renaming a unit to update placeholders, and deleting a unit.



OUTLINE

The active editor cannot provide outline

Overview of Microsoft Learn units

1/14/2022 • 5 minutes to read

Units in Microsoft Learn are effectively the "chapters" of a module. Each one contains a chunk of information that supports the module. As a content developer, you can think of them as a page where you can present content, knowledge checks, and interactivity such as exercises.

This article focuses on the implementation details of creating the units. For example, the article shows how to configure the MD and YAML files. For information on writing good learning content, view the [Tips for writing learning content](#) article.

Content is written in standard Markdown with a few extensions. Review the following pages for more information on the basics of writing content for Docs and Learn.

- [Markdown Reference](#)
- [Markdown Extensions for Docs and Learn](#)

Example

- [A live example of a unit.](#)
- [The YAML metadata for this unit](#)
- [The Markdown source file for this unit.](#)

Learner experience and validation

When learners view a Microsoft Learn unit, the user experience is similar to a standard Docs article. The content contains text, video, and images, or whatever the content developer has chosen to use. A unit will be complete depending on the validation configuration in the unit's YAML file.

Validation options

The validation options for text and Cloud Shell units are:

- **Viewed content (no task validation):** Unit will be marked as complete when the user views the initial page.
- **Task validation:** Unit will be marked as complete once the system has confirmed that user has done the specified task required in the unit YAML file. Example: Created an Azure VM with the name "msft-learn".
- **Knowledge check:** If a unit has a knowledge check, the unit will be marked as complete only once the user has passed the knowledge check with 100% accuracy.

IMPORTANT

At least one unit in each module should have task validation or a knowledge check, so that users can't get marked complete for an entire module by only viewing the units.

Implementation

Units are made up of many files (YAML, Markdown, and others). Also, units are the core content that the learner interacts with. All units have the same base properties in the YAML file. But, you can add other options like task validation, knowledge checks, and embedded interactivity to this file as well.

- **Unit files (`unit-name.yaml`):** Multiple YAML files (1 per unit) that define the metadata, content, and unit

validation methods.

- **Content files** (`unit-name.md`): Multiple Markdown files (1 per unit) that contain the actual content of the unit. These content files are added as `includes` references in the unit files.
- **Media files**: Any images or videos included in the unit content files. These media files are added as `includes` references in the content files.

Reusing content

The files that make up a unit (YAML and Markdown) can only be referenced by a *single* Learn module. You can't reuse content at the unit level directly. However, you may have repetitive text in multiple units or modules that you don't want to duplicate in each usage spot. If you do, you can use the [Docs INCLUDE markdown extension](#). This extension allows you to place the repeated content into a single, shared Markdown file and then inject it into each usage.

The shared Markdown file can be placed at any level of your folder hierarchy, but must be in the same GitHub repository. There's currently no way to share content across repos. We recommend creating a specific folder for your shared assets. A specific folder helps ensure files don't get deleted accidentally if someone moves, removes, or changes a consuming module later on.

Hiding content

If you have some content that isn't ready for publishing, or needs revisions, and you want to hide it from the rendering engine, you can apply the `sxs-lookup` class to the content block to hide it. This application simply hides the content in the browser, it *doesn't* remove the content from the published HTML. To hide the content, move the content into a quote block and add a `div` header with the `sxs-lookup` class as shown below.

```
> [!div class="sxs-lookup"]
> This content will not be shown
>
> ![[alt text.]](image).
```

But the content will be shown since it's not in the quote block.

Create a basic unit YAML file

Each unit file is a singular YAML file that defines the module metadata and the display of units.

Sample unit YAML for a text-only unit

```
### YamlMime:ModuleUnit
uid: learn.add-and-size-disks-in-azure-virtual-machines.introduction
title: Introduction
metadata:
  title: Introduction
  description: Introduction
  ms.date: 09/24/2018
  author: markjulmar
  ms.author: smmark
  ms.topic: interactive-tutorial
  ms.prod: learning-azure
durationInMinutes: 3
content: |
  [!include[]](includes/1-introduction.md)
```

Explanation of metadata

FIELD	VALUE	WHY?
-------	-------	------

FIELD	VALUE	WHY?
### YamlMime:	ModuleUnit	Specifies to the Docs platform the type of YAML file and metadata to expect. Must be the first line of the YAML file.
<code>uid</code>	Manually generated unique ID that follows the format: (module uid).(unit-short-name). Should match exactly as defined in the index.yaml.	This value is a unique ID to differentiate units within a module. For more information, see the Overview of uid article.
<code>title</code>	Unit title	The title of the unit that appears in the header, TOC, and breadcrumbs.
<code>metadata</code>	Unit-level metadata	These properties are helpful in search engine optimization.
<code>metadata > title</code>	Unit title	This value is the most important metadata for SEO.
<code>metadata > description</code>	A summary of the unit. Add a meaningful description for your unit, or add this blurb "This content is a part of [module title]."	Used in site search. Sometimes used on a search engine results page for improved SEO.
<code>metadata > ms.date</code>	A date in the format MM/DD/YYYY.	Displayed on the published page to indicate the last time the module was substantially edited or guaranteed fresh. If this field is missing, the date of the last commit is displayed instead, which may be incorrect for freshness.
<code>metadata > author</code>	The author's GitHub alias.	Identifies the author by GitHub ID in case there are questions about or problems with the content. In some cases, the author might be notified via GitHub automation of activity involving the file.
<code>metadata > ms.author</code>	The author's Microsoft alias, without <code>@microsoft.com</code> .	Used for content reporting and BI.
<code>metadata > ms.topic</code>	<code>interactive-tutorial</code> .	The type of content.
<code>metadata > ms.prod</code>	The type of product. Should follow the format <code>learning-<product></code> , such as: <code>learning-azure</code> , <code>learning-powerapps</code> , <code>learning-powerbi</code> , <code>learning-flow</code> , <code>learning-d365</code> , and so on.	Used for issue triage and reporting.
<code>durationInMinutes</code>	The total duration to take the unit.	This value appears in each article under the title. The duration for each unit is added up automatically to define the duration for the entire module, which displays on the module tile.

FIELD	VALUE	WHY?
content	Reference to the unit markdown file within the include tag.	The markdown file contains actual content. At the scaffolding stage, add blank markdown files for each unit and refer the file here.

Add more functionality and media

Choose the links below to view information about adding more functionality and media:

- [Add a Learn Lab](#)
- [Add task validation](#)
- [Add a knowledge check](#)
- [Add video](#)
- [Add artwork](#)
- [Add code samples](#)

Related Documentation

[Unified content model](#)

Write learning content

1/14/2022 • 13 minutes to read

Learning content design principles

Microsoft Learn content is different from docs content, and is built using the following principles:

- **The learner comes first:** Who is the learner? What do they need to know? What do they currently know? How can you design content that bridges that gap?
- **Write concise learning objectives:** Learning objectives should describe knowledge, skills, and attitudes that are clearly demonstrated through performance once the learning exercises are completed. An example for structuring learning objectives can be built using [Mager's framework](#) of *audience, behavior, conditions, and degree*.

Example

Let's say that your goal is to teach the learner how to deploy a simple Cosmos DB. Here is a sample objective, which could be used for this module:

The novice developer will be able to create a Cosmos DB, given access to an Azure test environment, with 75% accuracy.

Things to consider

- **Determine how they'll learn:** Certain types of lessons will be better absorbed in different ways. For your Microsoft Learn module, does that mean that the learner is reading text, or performing hands-on labs, or watching video, etc.?
- **Make learning simple:** Make what you're building fast and easy to learn. If it isn't, how could you restructure the way they're going down this learning path? Do the learning objectives need to be rewritten?
- **Evaluate the learnings:** Make sure that your learners are able to consistently complete the tasks in the module.
- **Make learning fun:** If you're bored, they're bored. Bored people don't retain information.

Docs platform guidelines and writing resources

Guidelines for Microsoft Learn are supplemental to what is already in place for the Docs team (meaning, you need to follow Docs guidelines and Microsoft Learn guidelines when writing Microsoft Learn content).

The following resources and sites provide official guidance from various groups about writing content. These should be used in tandem with the guidelines supplied for Microsoft Learn.

SITE	PURPOSE
Microsoft Docs Contributor's Guide	The manual for working within the Microsoft Docs publishing system.
Style guides for Learn and Docs content	A list of all the style guides available, and what order to use the style guides in.
Microsoft Open Publishing System	Learn more about the Microsoft Open Publishing System.

SITE	PURPOSE
DocFX Markdown Engine	The underlying generation engine used by OPS.
Microsoft Brand Central	Home for the Microsoft brand + guidelines.
Art and multimedia gallery guidance	Resources for official art and video for Learn and Docs content.

Tips for writing unit content

Units in Microsoft Learn are effectively the "chapters" of a module. Each one contains a chunk of information that supports the module. As a content developer, you can think of them as a page that you can present content, quizzes, and/or interactivity with.

- **Don't re-invent the wheel:** Microsoft has invested a lot of resources into studying what works and what doesn't work for writing Docs content. Our best practices and standards are stored in the [style guides for Learn and Docs content](#) and the [Microsoft Docs Contributor's Guide](#). Review these to write better content.
- **Keep it friendly:** The tone of an interactive tutorial should be conversational and casual, like a friend taking you on a journey through the product/service and patiently explaining how to use it. Review the [5 voice principles](#) for help with writing for this.
- **Build confidence:** *Training* has a different quality bar than other kinds of learning - the learner wants not only to become knowledgeable, but to become *confident* in actively applying their learning, because they know that the real world will test them with situations that they can't anticipate. The hands-on nature of tutorials builds confidence through experience, but tutorial activities can't possibly cover the full depth and breadth of the experience of developing/architecting/administering software for the cloud. In a training context, marketing-style content is distracting and reduces confidence that the content is grounded in truth. Focus on facts and concrete statements. Consider using the following content strategies for building confidence:
- **Definitions are critical.** Before describing what a product or service can *do*, define what it *is* in unambiguous terms with at least one noun, even if the definition needs to be unpacked and further explained. Define Azure-specific terms, use them consistently, and proactively call out if terms have different meanings in different contexts.
- **Focus on technical outcomes.** When explaining what a product or service does, avoid vague descriptions and promises. Adjectives like "intelligent" and "smart" and broad, aspirational phrases like "empower every person to achieve more" don't add any value for an audience of technical learners.
- **Don't make unqualified value judgments.** Broad comparisons and adjectives like "fast", "powerful" or "versatile" without concrete metrics or specific points don't add any value to training content.
- **Create scenarios around problems, not products.** "Product tours" that guide readers through long, intricate happy-path scenarios feel like advertising. They're generally created to showcase a product as the best solution to a specific problem. When building tasks and scenarios, focus on a problem statement first before using a product to find a solution.
- **Enable good decision-making:** A significant source of uncertainty and apprehension related to Azure is *choosing*: selecting between strategies, services, implementations, performance tiers, architectures, etc. Beginners are best served by quickly giving them enough knowledge and context to understand what kinds of decisions need to be made and the ability to confidently make them.
- **Examine alternative steps/tasks and tools.** Don't deep-dive on details, but help the learner understand what the relevant options are and how they compare in a basic way. Explain relevant variations on problems and scenarios and how they might be handled differently.
- **Call out common decision points and criteria.** What are the characteristics of other scenarios that would call for a different selection?

- **Compare instead of recommend.** Don't directly recommend certain products or services over others - focus on technical and conceptual comparisons and guiding the learner to make decisions based on their goals. Don't alter or reframe standard scenarios or problem statements with the goal of making a product you want to promote look like a best fit.
- **Simplify explanations of tradeoffs.** When exploring alternatives within a task, call out whether or not the process of deciding between them is nuanced, but leave out the specifics if it is. "Rules of thumb" are a great way to promote confidence and enable decision-making for beginners.
- **Present the real-world.** Build your scenarios around realistic problems that learners may relate to in their day-to-day activities.
- **Acknowledge similarities/differences to real-world scenarios.** It's expected and understood that tutorial tasks don't have the same limitations and freedoms as real-world scenarios, and that they take shortcuts to save time and complexity, but be explicit about where, why and how. "We're putting your connection string straight into your application code here for simplicity. In a real application, you would put your connection string in a configuration file, because...".
- **Illustrate best/common practices.** For example, in Azure, resources and resource groups can be organized and named however you like. But how do experienced people *actually* do it, and why do they do it that way? What's an example of a naming strategy that would get a beginner laughed out of a design meeting with more experienced people, and why?
- **Provide context.** Discuss other tasks that may come before or after the tutorial task in a real-world scenario, or the kinds of applications that would require such tasks. Consider discussing historical context of recommendations and design decisions when appropriate.
- **Empathize with the learner:** Showing empathy is one of the [core principles](#) of the Docs Style Guide. A beginner needs more than technical information, they need assurance and support.
- **Describe subjective experience.** Understanding how something subjectively *feels* to other people before trying it boosts confidence. Subjective experience is different for everyone, but try to capture the universal aspects. "Designing objects is a creative activity with lots of room for different strategies"; "Architecting a cloud service from a blank slate can be an overwhelming experience at first."
- **Be honest, especially about bad subjective experiences.** Tutorials guarantee forward progress, correct answers and a tangible result, and many people find them inherently fun. Sometimes the real-world version of the same task is just a chore, or is universally frustrating - don't pretend otherwise! "Sometimes creating a configuration that does what you want can take a lot of searching and experimentation, and sometimes you find out it's not possible."
- **Be up front about expected difficulty.** If a concept or task workflow is especially difficult/complicated or is supposed to be especially easy, particularly in a real-world context, call it out. If it's supposed to be easy, what might cause a learner to find it difficult? If the tutorial makes a task appear simple but it's often difficult in a real scenario, why?
- **Don't assume clarity or ease, and don't be dismissive.** Avoid words like "obviously" and "clearly". Be careful about labeling tasks or concepts as easy or simple without a qualifier - consider comparing them to another relevant task that is more complicated or difficult to make the point.
- **Anticipate questions and answer them.** The [Curse of Knowledge](#) makes this difficult. Try explaining the concept to a beginner and see what confuses them the most; browse existing documentation and user experiences like the Azure portal to find places where other authors and designers have called out concepts they found to be confusing; identify places where you yourself have questions like "why does it work that way?"
- **Use media wisely:** Using a combination of videos, images and text in combination, can create a compelling experience. When using videos in your articles, start with an overview that's a continuation from the previous in-category tutorial.

Tips for writing lab content

Lab units will likely be the longest units in a Triple Crown module and they are the learner's opportunity to get

experience with the Microsoft products. So, it is important to keep the learner engaged. Here are tips for writing good lab content:

- **Help learners achieve success early and often.** Help the learner achieve small victories throughout the lab unit, with no more than 5-10 minutes of work in between obvious progress.
- **Make each lab unit interactive, engaging, and fun.** Labs should make the learner think and explore the technology. Try to put some personality into the lab – for example, Xamarin University often includes monkeys (due to the Xamarin = Tamarin tie) or adds a bit of whimsy into the lab that is often missing from the Enterprise developer's day job. It's not necessary to create silly labs, but the unit should lean towards more engaging data and concepts vs. standardized ("dry") content.
- **Always demonstrate best practices.** Any source code presented to the learner should be reusable by the learner where possible. It's common for learners to take the code and use it in their own projects. If the code omits required production elements for simplicity, it should be called out in the comments to indicate that the code needs additional work to be production ready.
- **Keep the lab units relatable (as much as possible).** They should have a real-world feel to it (no "foo" labs). Labs should build something that people would want to create - not a documentation "sample" meant to show off a small feature.
- **Keep labs as simple as possible.** When simplicity interferes with best practices, defer to best practices. Always ensure the instructions and associated code are leveled to the expected skill and experience of the average target audience.
- **Focus the exercise and cover primary scenarios (not edge cases).** The activity should cover the specific content explained in the learning materials; extraneous elements should be avoided. If it is necessary to use something core to the technology to build the lab that is not covered in the associated module (or prior modules), then revise the learning materials to make sure it's covered. Do not cover a new topic in a lab – labs are not for teaching concepts, they are for experiencing topics that have already been explained.
- **Keep a consistent structure.** The unit content itself should be composed of an **introduction**, one or more **sections**, and a **summary**.

Structure of lab content

The unit content itself should be composed of a **starting point** (if applicable), an **introduction**, one or more **sections**, and a **summary**.

Starting point

Many labs will need a *starting point*, which might be an existing project, or a template to launch a VM. Starting points should be called out directly as the first lab unit section – prompting the learner to download the associated project and /or template. Source code should be in GitHub, under the MIT license, with the project zipped up and stored in the <docs.microsoft.com> area used for samples.

Introduction

The introduction of the unit should have several elements to it:

- Goals for the lab unit
- Expected time to complete the unit
- Challenge steps

Goals for the lab unit

The unit should have clearly defined goals that are stated at the beginning of the content. For example: "In this lab you will learn how to ..."

The motivation should be clear:

- *Why* are you doing this lab.
- *What* are you going to build

- *What tools will you be using, etc.*

Challenge Steps

All labs should have some challenging aspects that stretch the learner. Challenges can be placed at the end as optional tasks, or in the middle, with a way for the learner to solve the challenge without being frustrated (for example, code hints, etc.).

Also as a part of the introduction, list high-level steps that the learners will do which link to the specific explanation sections. Linking allows learners who are already familiar with the technology to "jump ahead", without having to read through the full description of the steps.

Ideally, labs should also allow for experimentation. One of the goals for learners is to allow for exploration of the technology in question - so learners should be able to go sideways and pursue a different trail on their own for a little while and then come back in line with the main lab unit goals.

NOTE

This could be accomplished by providing directives around constrained opportunities for creativity or extensibility of the project: "Explore this on your own by ...".

Lab Sections

Triple Crown labs should be built in small, logical sections. Each section should have **no more than 10 steps to complete**. If oriented around a programming task, the project should be buildable and/or runnable with changed behavior when the section is finished to validate syntax. This will allow learners to more quickly identify errors and fix them before too many things have changed. Generally, each subsequent section should add to the lab without requiring removal or modification of work done in a previous section.

The steps should be written to the expected level and experience of the learners – for example, provide explanations where the learner might not be familiar with a step, or a link to external documentation where necessary.

Screenshots and videos can be useful in the sections, and where applicable, show the output from the section (for example, app running, output from the console, etc.). However, be cautious about capturing static images of content that changes rapidly to avoid constant refresh cycles – in these cases, a more generic textual description may be a better approach.

NOTE

When code examples are used, make sure to follow the [Code Samples Guidance](#). If possible, an integrated CLI experience or VM is ideal for these.

Cleanup

The final lab unit section should detail the cleanup / shutdown steps required for the lab. This would involve removing any Azure resources, deallocated VMs, uninstall apps, etc.

Unit Summary

Finally, each lab unit should be terminated with a summary that reminds the learner what they have accomplished ("you learned how to..."), provides optional "exploration" ideas to think through and work on their own, and a link to a completed, working code project if the lab unit is associated to a programming task in case they had trouble finishing the lab.

Ideally, include an interrogation point here to ask how difficult the learner found the lab and success rate. It would also be nice to collect analytics on how long the learner took to go through the content – this could be used to baseline the timing, or even determine whether they attempted the lab unit.

Add a knowledge check to a unit

1/14/2022 • 6 minutes to read

Overview

A knowledge check can be standalone as the only content in a unit. It can also exist after the learning content in a unit at the bottom of the page. The [expected question count](#) is based on whether the knowledge check is standalone.

Learners must pass units with a knowledge check to earn credit for completion. The reason isn't to require compliance. Rather, the purpose is to allow the learner to check whether they kept the information or to reinforce the knowledge to the learner.

IMPORTANT

We refer to this type of interactivity as a "knowledge check" and not "quiz" to the learner (even though the property in the file is called `quiz`). The term "knowledge check" helps avoid the heavy compliance connotation of the word "quiz". Any use of the word quiz that is shown to learners will block a PR from being merged.

Learner experience and validation

When learners view a text with knowledge check unit, they don't have to sign in to access Microsoft Learn content. But they do need to sign in to save their progress. Learners receive experience points for successfully completing each knowledge check unit. They need to sign in to save the points earned. However, they can choose to continue without signing on. Without signing in, the module won't save their points earned and they won't earn a badge for completing a module.

Learner Experience

- User is consuming Microsoft Learn content (authenticated or non-authenticated).
- User gets to knowledge check (authenticated or non-authenticated).
- User answers multiple-choice questions (authenticated or non-authenticated).
- User submits answers for validation (authenticated or non-authenticated).
- Correct and incorrect answers will be shown to the user. For example, the answers shown are Question 1 = Wrong, Question 2 = Correct, and so on, with explanations if the content author wrote explanations.
 - If the user got the question incorrect, they'll only see that it's incorrect and not what the correct answer is.
- User can retake the knowledge check as many times as they want until they get 100% correct AND the answer selection doesn't reset each time. They only need to fix the incorrect answers, not all of the answers each time.

Knowledge check validation - correct and incorrect answers

Check your knowledge

1. Suppose you work at a startup with limited funding. Why might you prefer Azure data storage over an on-premises solution?

- To ensure you run on a specific brand of hardware which will let you form a marketing partnership with that hardware vendor.

- The Azure pay-as-you-go billing model lets you avoid buying expensive hardware.

There are no large, up-front capital expenditures (CapEx) with Azure. You pay monthly for only the services you use (OpEx).

- To get exact control over the location of your data store.

2. Which of the following situations would yield the most benefits from relocating an on-premises data store to Azure?

- Unpredictable storage demand that increases and decreases multiple times throughout the year.

- Long-term, steady growth in storage demand.

Azure data storage is flexible. You can quickly and easily add or remove capacity. You can increase performance to handle spikes in load or decrease performance to reduce costs. In all cases, you pay for only what you use.

- Consistent, unchanging storage demand.

Knowledge check complete



Congratulations!

You earned experience points for completing the knowledge check.



[Sign in to save progress >](#)

[Continue without saving progress](#)

Validation method

- **Knowledge checks 100% correct:** Learners will complete the unit once they complete the knowledge check with 100% skill. They have unlimited attempts and they don't have to reanswer questions that have already been marked as correct.

Tips for writing knowledge check questions

For guidance on writing knowledge checks, refer to the [How to write a knowledge check](#) article.

Resources:

- <https://community.articulate.com/articles/20-tips-for-writing-great-quiz-questions-and-response-options>
- <https://timslade.com/blog/tips-for-writing-better-e-learning-quiz-questions/>

Implementation

Units that have knowledge checks have the knowledge check questions embedded directly in the unit YAML file.

IMPORTANT

If you are writing a knowledge check as part of a module that will use Lab OnDemand, the knowledge check is required to be in its own unit and be the very last unit in the module.

Unit files (`unit-name.yml`)

Multiple YAML files (1 per unit) that define the metadata, content, and validation method of units.

Sample unit YAML for a unit with a knowledge check

The `content` property contains an includes file referencing Markdown for the learning content. This property is optional. If your unit doesn't have any learning content and is a knowledge-check only, then this parameter would be left blank. The content property is required, but the value is optional.

```
### YamlMime:ModuleUnit
uid: learn.azure-cli-2-0
title: Create virtual machines with Azure CLI 2.0
metadata:
  title: Create virtual machines with Azure CLI 2.0
  description: Get started with Azure CLI by creating virtual machines.
  ms.date: 11/27/2018
  author: barlan
  ms.author: barlan
  ms.topic: interactive-tutorial
  ms.prod: learning-azure
durationInMinutes: 3
content: |
  [!include[]](includes/resource-group.md)
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.
```

Explanation of metadata

METADATA	REQUIRED	DESCRIPTION
### YamlMime:ModuleUnit	Yes	Specifies to the Docs platform the type of YAML file and metadata to expect.

METADATA	REQUIRED	DESCRIPTION
<code>uid</code>	Yes	A human generated unique ID that follows the format: (repo-name).(module-short-name).(unit-short-name).
<code>title</code>	Yes	Unit title. Displayed in the header, ToC, and breadcrumbs. This property is the most important metadata for SEO.
<code>metadata > title</code>	Yes	A summary of the unit. During scaffolding, the description can be same as the title. Used in site search. Sometimes used on a search engine results page for improved SEO.
<code>metadata > description</code>	Yes	A summary of the unit. During scaffolding, the description can be same as the title. Used in site search. Sometimes used on a search engine results page for improved SEO.
<code>metadata > ms.date</code>	Yes	A date in the format MM/DD/YYYY. Displayed on the published page to indicate the last time the module was substantially edited or guaranteed fresh. If the field is missing, the date of the last commit is displayed instead, which may be incorrect for freshness.
<code>metadata > author</code>	Yes	The author's GitHub alias. Identifies the author by GitHub ID in case there are questions about or problems with the content. In some cases, the author might be notified via GitHub automation of activity involving the file.
<code>metadata > ms.author</code>	Yes	The author's Microsoft alias, without <code>@microsoft.com</code> . Used for content reporting and BI.
<code>metadata > ms.topic</code>		The type of content. For example, <code>interactive-tutorial</code> .
<code>metadata > ms.prod</code>		The type of product. Should be one of these values: <code>learning-azure</code> , <code>learning-powerapps</code> , <code>learning-powerbi</code> , <code>learning-flow</code> , <code>learning-d365</code> , <code>learning-<product></code> . Used for issue triage and reporting.
<code>durationInMinutes</code>	Yes	The amount of time that the unit takes to complete, in minutes. It's displayed to the learner.

METADATA	REQUIRED	DESCRIPTION
<code>content</code>	Yes	Source of the content. You can write Markdown directly in the YAML here, but it isn't advised. Rather than putting all your Markdown inline, you can use a Markdown <code>include</code> file. Using an <code>includes</code> file allows the Markdown for each step to be in a dedicated Markdown file, providing a better editing and preview experience.
<code>quiz</code>	No (Yes if including a knowledge check)	The beginning of the knowledge check content.
<code>quiz > title</code>	No (Yes if including a knowledge check)	Title of the knowledge check.
<code>quiz > questions</code>	No (Yes if including a knowledge check)	The beginning of the knowledge check question content.
<code>quiz > questions > content</code>	No (Yes if including a knowledge check)	Knowledge check question.
<code>quiz > questions > choices</code>	No (Yes if including a knowledge check)	The beginning of the knowledge check answer options.
<code>quiz > questions > choices > content</code>	No (Yes if including a knowledge check)	Knowledge check answer option.
<code>quiz > questions > choices > isCorrect</code>	No (Yes if including a knowledge check)	Boolean value if the answer is correct (true) or not (false).
<code>quiz > questions > choices > explanation</code>	No (Yes if including a knowledge check)	Hint/Explanation for the user why this answer is the correct or incorrect answer.

Template knowledge check YAML

The following example is a template with values easily replaceable for you to drop into your YAML.

```
quiz:  
  title: Check your knowledge  
  questions:  
  
    - content: 'Question 1'  
      choices:  
        - content: 'Answer 1'  
          isCorrect: false  
          explanation: 'Explanation 1'  
        - content: 'Answer 2'  
          isCorrect: false  
          explanation: 'Explanation 2'  
        - content: 'Answer 3'  
          isCorrect: false  
          explanation: 'Explanation 3'  
        - content: 'Answer 4'  
          isCorrect: true  
          explanation: 'Explanation 4'  
  
    - content: 'Question 2'  
      choices:  
        - content: 'Answer 1'  
          isCorrect: false  
          explanation: 'Explanation 1'  
        - content: 'Answer 2'  
          isCorrect: false  
          explanation: 'Explanation 2'  
        - content: 'Answer 3'  
          isCorrect: false  
          explanation: 'Explanation 3'  
        - content: 'Answer 4'  
          isCorrect: true  
          explanation: 'Explanation 4'
```

What is interactivity in Learn?

1/14/2022 • 10 minutes to read

Interactivity in Microsoft Learn context is defined as instructional elements that are performed by the learner with the product they are learning about as they go through the content (sometimes commonly referred to as exercises). We require that every Learn module includes interactivity so that the learner can *do* the things they are learning about in a controlled environment that helps them be successful.

Exceptions are allowed if there's a compelling reason to exclude interactivity. An example of this might be a module where the purpose is to teach the learner how to determine which Microsoft product to use to build their solution when many could be used (a [choose module](#)). In that case, we aren't teaching about a specific task with a service but are instead trying to inform the learner which ones they should use, or the pros and cons of each.

Modules that include interactivity give the learner the opportunity to apply their new knowledge and gain hands-on experience. This active learning experience helps to cement the new concepts, fail and recover in a guided fashion, and improve comprehension and retention. Having at least one opportunity to use experience the product or service being taught is considered a best practice for Learn modules.

What is a lab on MS Learn?

A lab consists of a set of exercise instructions provided in one or more units of a module, combined with access to a Microsoft-provided lab environment which the learner uses to complete the exercises. Labs allow learners to simulate the hands-on practical application of the concepts taught in a given module in a realistic fashion at no cost to the learner.

IMPORTANT

A lab can span multiple units within a single Learn module, but can't be reused across multiple modules.

Creating a Lab is part of the [module planning & creation process](#). Content Authors should plan for their lab experiences when developing their overall Learn content plans.

Learn Lab Guidelines for content authors

When planning a lab, there are several guidelines to keep in mind.

- Learn modules should take a learner no more than 60 minutes to complete, including a lab experience. Factor your lab experience into your module design.
- The lab should teach the learner to do something useful, not just demo features.
- Leverage the conceptual content as part of the exercise - have the reader recall the details vs. just showing a step-by-step tutorial.
- Respect the learner's time. Consider whether the lab experience meaningfully contributes to the module and use other formats, such as screenshot walkthroughs or a short video, to demonstrate the most simple tasks.
- Labs should demonstrate methods that are consistent with best practices and standards.
- Content authors are accountable for maintaining their labs. Learners should not have to worry about the freshness, accuracy, or relevancy of the content on Learn. Learn will remove content from the catalog when needed.
- Take advantage of MS Learn functionality to enhance your learner's experience. For example, use the [Zone](#)

[Pivot functionality](#) to provide different versions of lab exercises within the same unit. For example:

- When a lab exercise can be completed from either the Cloud Shell or the Azure Portal GUI, consider creating separate versions of the lab exercises for each and allowing the Learner to select their preferred approach.
- Create "beginner" and "advanced" versions of the lab exercises, allowing more in-depth explanation and granular instructions for beginners while allowing advanced learners to focus on information more relevant to them.

For more tips, see [Guidance for creating technical exercise content](#)

Lab environments

There are several hosted lab environments that can be used to execute a lab in Microsoft Learn:

1. Azure sandbox
2. Dynamics 365 sandbox for Sales
3. Jupyter Notebooks sandbox
4. Labs on Demand (VM)
5. Try.NET C# REPL

All of these environments allow learners to try Microsoft products for free.

Azure sandbox

The Azure sandbox is the *preferred* environment for Azure-based labs. Learners sign into the Azure Sandbox environment with their Microsoft account and access a Microsoft-owned Azure subscription. There is no need to enter any personal information/credit card to complete the exercise. The lab can use an integrated Cloud Shell (a browser-based command-line for Azure) or the Azure portal to create and manipulate Azure resources such as databases, virtual machines, and websites.

Pros

- Free for the learner with no requirement to have an existing Azure account.
- The learner doesn't have to give Microsoft any personal information beyond email and name (taken from the logged on Microsoft account).
- Azure cost is billed at internal rates (AIRS).
- Can decide the resources the learner can create.
- Can validate what the learner has created.
- Can control how long the environment exists (1-4 hours).
- Each module gets a unique (clean) environment.

Cons

- Exercises cannot span modules as each module is given a new environment.
- Learners cannot (easily) save their work.
- Only have access to a single Azure Resource Group (RG) which is created as part of the environment. services that require multiple resource groups are currently not supported.
- When the Sandbox times-out, all resources are deleted and all work is lost.
- Learners can only activate 10 Sandboxes per day.
- Cannot use or create resources that require a service principal (a user in Active Directory).
- Cannot create or use resources that are not controlled by Azure Policy or don't exist in a resource group.

Read the [Add an Azure sandbox-based lab](#) article for more detail.

Dynamics 365 sandbox for Sales

MS Learn has an integrated Dynamics 365 experience for Sales and Customer Engagement products. This

environment is similar to the Azure sandbox. The user is placed into a D365 business group and has access to the environment with a pre-loaded set of sales data for 1-4 hours.

Read the [Add a Dynamics 365 Sales sandbox-based lab](#) article for more detail.

Jupyter Notebooks

Jupyter Notebooks has become a primary data science tool used to analyze, test, and make predictions from data sets. Jupyter is now a mainstream technology in academia, leveraged by many universities as a primary vehicle for data science training. MS Learn has an embedded Notebooks experience that enables exercise units to be written using a Jupyter Notebook.

This format dictates a slightly different flow for coding exercises which involves Markdown content and code cells be interleaved within the single unit. Currently, the only language available is Python - however the experience does support different environments:

- Python 3.6
- Python 3.6 with TensorFlow
- Python 3.6 with PyTorch
- Python 3.7
- Python 3.7 with TensorFlow
- Python 3.7 with PyTorch

Read the [Add a Jupyter Notebook sandbox-based lab](#) article for more detail.

Labs On Demand

Labs on Demand (LOD) is a third-party vendor that supplies a virtual machine environment over the Internet. Learn modules can use this approach when the service being explored is not supported in the Azure Sandbox environment. This includes labs for non-Azure products, labs that require a pre-installed environment, labs that require elevated Azure permissions, or labs that need a more sophisticated setup such as Azure Stack.

Choose a VM Lab if Learn's [Azure Sandbox](#) can't support your lab experience. VM Labs are typically required when:

- An Azure lab experience requires elevated permissions (higher than "contributor") to an Azure resource group, or access to more than one resource group.
- A lab experience is required for Microsoft products other than Azure (Microsoft 365, Business Applications, etc.)
- The lab experience requires pre-installed software, or a pre-configured environment.
- The lab experience requires the learner to interact with another technology stack which must also be virtualized.

LOD supports two interactivity options: Virtual Machine (VM) or Cloud Slice.

OPTION	DESCRIPTION
Virtual Machine	The VM-based lab enables lab scenarios that require a Windows or Linux environment with a set of pre-installed tools. For example, a Visual Studio Enterprise installation. In this model, the content team builds a VM image used as the basis for all lab environments. Activating a LOD VM-based experience opens a new browser tab with a prompt to login to the LOD environment with a set of provided credentials. Once signed in, the learner is presented with a desktop view of a remote computer along with the lab instructions on the right side of the screen. This approach uses Azure subscriptions that are owned by Microsoft.

OPTION	DESCRIPTION
Cloud Slice	The Cloud Slice environment provides access to a dynamically created Azure subscription using a set of supplied credentials. This approach leverages LOD's capability as an Azure EA reseller and enables lab scenarios that require subscription-level privileges. Activating a Cloud Slice experience opens a new browser tab that includes the username/password to log into Azure. There is no VM in this case - instead, the learner signs into the Azure portal using the provided credentials that provides access to an LOD-owned Azure subscription.

Pros

- Free for the learner with no requirement to have an existing Azure account.
- VM environment can provide installed tools such as SQL Server Management Studio or Visual Studio.
- Can control VM CPU, memory, disk space, etc.
- Enforces time limits to control costs (1-4 hours).
- Fraud detection and reporting monitored by LOD.
- Cloud Slice allows subscription-level privileges that enables support for AAD, etc.

Cons

- Exercises cannot span modules as each module is given a new environment.
- Experience isn't as integrated. Requires a new browser tab and unique login to their environment. Learner must switch back to the original tab to continue the module.
- Cloud Slice is priced at retail Azure costs + 20% markup.
- Learners cannot (easily) save their work.
- By default, only 30 concurrent users can launch VM-based labs for Azure. Teams can pay an additional cost to up this limit. Dynamics and PowerApps allow 100 concurrent users.

Check the [Add a Learn Lab to a unit using a VM lab experience](#) article for more detail on creating a VM-based lab.

Contact [Mark Smith](#) on the Learn product team for information on how to use Cloud Slice.

C# REPL

The Try.NET experience is supported in Learn content. Check the [Add a C# based lab to a unit](#) for more details.

What if I can't use any of the hosted environments?

In some cases, interactivity is desired, but cannot be enabled using the supported managed environments. In these instances, there are two available options:

1. Use an interactive "click-through" video
2. Use a personal subscription/account.

Interactive video

A second option is to record an interactive video and embed it into the content. This allows the learner to see how something works and potentially interact with the product in a simulation without requiring any installs or registration. We are currently exploring options for this type of interactivity. Please contact the Learn team if you are interested in exploring this option.

Personal subscriptions

Labs can direct the learners to install the product or use trials to learn the product. In these cases, care should be taken to ensure that the exercises always provide a no-cost route so we don't violate our 'Free Learning' tenant

of Microsoft Learn. This is not a preferred option as it adds more complexity to the learning, as well as a burden on the learner to install software or create an account.

IMPORTANT

Always make sure you have no other options before using a personal subscription. Having users create Azure accounts requires entering personal information - including credit cards, which can limit the audience significantly worldwide. In addition, it adds friction to the experience and can have a cost on the user which violates the "free learning" tenant of MS Learn.

Pros

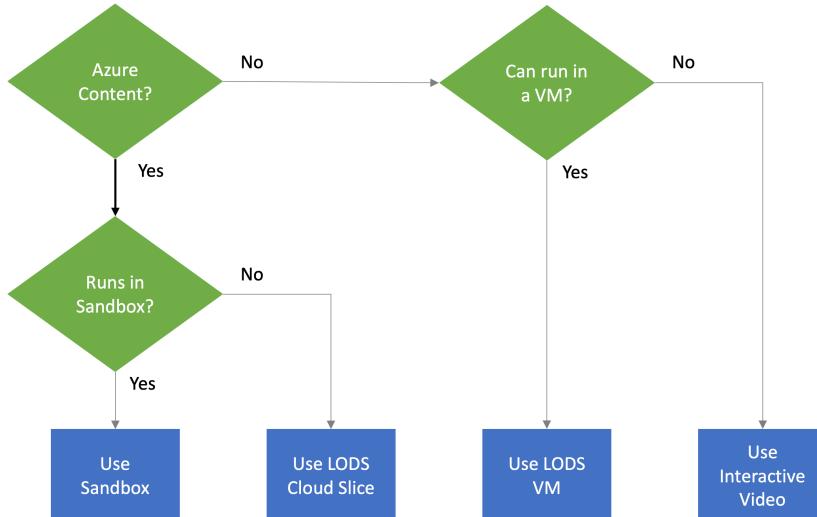
- Free to Microsoft.
- The work is all done in the learner subscription. They can do whatever they want with it and keep it as long as they want.
- Instructions are completely real – exactly what they'd have to do on their own.
- Easiest to write for since you can test using your own account.
- Work can carry forward from one module to the next since the learner controls the lifetime.
- You can do (almost) anything.

Cons

- It's not free. Might even have unexpected costs if the learner doesn't delete the resources.
- Misleading marketing messaging to learners – we promote that Microsoft Learn is free all over the site and marketing, but personal subscriptions and software are not.
- We can't (currently) use the embedded Cloud Shell environment because it's tied to the Sandbox today. That means the learner has to open a new tab to do exercise work or open a command-line prompt on their local computer.
- There are a few things that are limited to domain owners – particularly with Azure Active Directory.

Choosing the interactivity type to use

When authors are designing their content, they scope out what they want the learner to learn/do and design the best exercise/interactivity to support that. Next, they must confirm which interactivity option they can use to deliver this experience. They would follow a logic tree that is more expansive, but looks like this:



Need Help or have a question?

If you run into problems or have questions about labs, you can post them in the [Docs Support channel in Teams](#).

Learn Lab funding model

1/14/2022 • 2 minutes to read

The Azure and Dynamics sandbox environments are funded by DevRel and BAG. LOD based labs are funded by the business group sponsoring the content.

The basic cost breakdown is as follows.

	LEARN SANDBOX LABS	LODS VM LAB	LODS VM LAB (VM CREATED BY CONTENT TEAM)	LODS CLOUD SLICE
Setup Cost	No setup fee	- ~\$5,000-\$10,000 Setup fee per lab - Cost varies by length of lab & complexity of lab environment	- No charge from LODS when performed by content author. - \$250 LODS security & abuse QA required to publish any lab profile not created directly by LODS	No setup fee
Ongoing Hosting Costs	Azure Consumption	- \$650 per lab with max 30 concurrent users, per year - Additional monthly maintenance costs for large, complex, or resource-intensive labs environments - Azure Consumption if using LODS subs. If using Learn's subs, Learn may cross-charge for high-consumption/high-cost scenarios	- \$650 per lab with max 30 concurrent users, per year - Additional monthly maintenance costs for large, complex, or resource-intensive labs environments - Azure Consumption if using LODS subs. If using Learn's subs, Learn may cross-charge for high-consumption/high-cost scenarios	Azure retail rates + 20% markup.
Lab Changes & Updates	No cost, Sandbox is maintained by Learn & Lab unit edits are performed by content owner in their Learn repo	- \$150/hour, lab environment changes performed by LODS Support - \$0 Lab unit edits are performed by content owner in their Learn repo	- \$0/hour, lab environment changes performed by content author - \$150/hour, lab environment changes performed by LODS Support - \$0 Lab unit edits are performed by content owner in their Learn repo	No cost.
Concurrent Users	No predefined limit	- 30 concurrent users by default - Additional \$650 annual hosting fee per lab for each additional 30 users	- 30 concurrent users by default - Additional \$650 annual hosting fee per lab for each additional 30 users	Negotiated with LOD.

	LEARN SANDBOX LABS	LODS VM LAB	LODS VM LAB (VM CREATED BY CONTENT TEAM)	LODS CLOUD SLICE
Time to Create & Deploy	<ul style="list-style-type: none"> - Minimal setup time required - Lab available as soon as module published to Learn - Localized module content live within ~1-2 weeks after module published in English 	<ul style="list-style-type: none"> - ~2-3 weeks to return a functional lab for QA by content author (assumes content author hands off fully-defined lab experience and lab exercise units to LODS) - Additional time may be required for complex lab environments - Additional security QA required for labs at risk for abuse - Must coordinate lab availability on LODS & Learn module publish date - Localized module content live within ~1-2 weeks after module published in English 	<ul style="list-style-type: none"> - Timeline determined by content author - Allow 1-2 weeks for security & abuse QA by LODS - Additional security QA required for labs at risk for abuse - Must coordinate lab availability on LODS & Learn module publish date - Localized module content live within ~1-2 weeks after module published in English 	Depends on the complexity of the lab.
Billing	Generally, funded by Learn. Azure consumption may be cross-charged in high-consumption or high-cost scenarios	<ul style="list-style-type: none"> - LODS will invoice content author/business group directly for lab creation, consumption, & maintenance fees - Azure consumption to Learn's subs may be cross-charged in high-consumption or high-cost scenarios 	<ul style="list-style-type: none"> - LODS will invoice content author/business group directly for hosting, consumption, and maintenance fees - Azure consumption to Learn's subs may be cross-charged in high-consumption or high-cost scenarios - \$250 for security & abuse QA. 	Requires a PO with Learn on Demand Systems.

Lab security and preventing abuse

1/14/2022 • 2 minutes to read

Overview

Labs are a perpetual target for abuse and misuse for a variety of purposes, including crypto currency mining, bot networks, piracy, identity theft, denial of service attacks, etc. Such abuse can rapidly escalate to tens or hundreds of thousands of dollars in Azure consumption in less than a day. To continue to offer our customers premium lab experiences for free, Learn has taken a cautious approach to lab development and publishing.

Shared security model

Content authors and Learn share responsibility for preventing lab abuse.

The Learn team and our vendor partners continuously monitor resources used by labs for evidence of abuse. When abuse is identified, Learn takes action to mitigate it as quickly as possible. This may include banning individual users, temporarily restricting the number of concurrent users, temporarily taking the lab offline, or completely removing the Learn module from the platform until the root cause can be identified and a fix implemented.

The number one vector for lab abuse are labs with missing or poorly configured permissions & access controls. If a learner has free reign to create virtual machines in a lab, those VM's can be used for virtually any purpose. Content authors are responsible for taking reasonable steps to properly configure & secure their labs against abuse. They should proactively review the the lab experience they intend to enable and define the minimum resources & permissions required to deliver that experience.

Modules with VM labs must have the lab environment QA'd by LODS prior to going live on Learn. Modules at high risk for abuse may go through additional QA prior to being allowed to go live.

Lab security guidelines

In general, the following guidelines should be applied:

1. Labs should follow the principle of "least possible privilege," meaning that the lab should be configured to allow the minimum permissions & resources required to complete the lab exercises as-written.
2. Labs requiring contributor-level access and above must have robust restriction policies.
3. Labs that allow creation of virtual machines must be limited to no more than 2 hours and use policies to restrict learners to creating the smallest, least powerful VM needed to complete the lab while still allowing for a good learner experience.

When lab abuse happens

If a lab becomes the target for abuse, Learn will take appropriate actions to minimize the impact to the platform and for learners. Learn will make a reasonable attempt to notify content authors of the abuse & actions required, however, abuse mitigation will take priority.

Learn may employ any of the following:

- Temporarily disabling the lab(s) pending review & fix
- Temporarily disabling the module(s) pending review & fix
- Permanently removing the module(s) pending reauthoring of content and/or lab environment by content

author.

- Temporarily removing content author/vendor access to create VM labs.
- Permanently revoking content author/vendor access to create VM labs.

The Learn Azure sandbox

1/14/2022 • 7 minutes to read

Using the Azure sandbox in a Learn module requires a few planning steps along with some administration. It's recommended to plan out your exercise completely - know what resources you'll need to create and how long it takes to go through the entire exercise from start to finish. If it takes more than an hour to go through the module, consider breaking the module up to make it easier to get through in a short amount of time.

Once you know what the lab will need, there are five steps that must be done to use the Azure sandbox.

1. Verify your lab scenario can run in the Azure sandbox.
2. Submit a registration request.
3. Add unit metadata.
4. Add task validation.
5. Test and publish the module.

Step 1: Verify your lab can run in the sandbox

The Azure sandbox has limitations enforced to protect privacy and control costs. Before you start planning, check the [known restrictions](#) to see if your specific lab requires elevated privileges, more than one resource group, or needs to create unsupported services.

Try to create your lab scenario in PPE

If you aren't sure whether your scenario is supported, try creating it in the pre-production environment (PPE). This sandbox has no Azure policy restrictions applied but grants the same contributor-access resource group as a production Azure sandbox.

1. Go to the '[no policy](#)' test page.
2. Activate the sandbox.
3. Go through all the steps in your lab exercise.
4. If you are able to complete the exercise, then the lab itself *can possibly* be supported depending on predicted costs and a security evaluation - move onto step #2.

IMPORTANT

If you receive an error at some point, then you are running into one of the technical restrictions of the sandbox environment and you won't be able to use this approach. Instead, consider using an [LODS VM](#) or [CloudSlice environment](#).

Step 2: Submit a registration request

Once you've determined the lab scenario is supportable, you must submit a request to register your module in the Learn sandbox database. This will link your module to a specific set of Azure Policy rules that will guard against fraudulent activity and put restrictions on what students can create or use while using the Azure sandbox with your module.

1. Make sure you've performed all the steps for your lab and created all the services in the PPE Azure sandbox environment.
2. Identify the Azure resources your module creates or requires. You can use the Azure Cloud Shell to list the resources with the following command:

```
az resource list --resource-group {resource-group-name} --query [].type --output tsv | uniq
```

3. Fill out a [Sandbox Module Onboarding Request Form](#) in SiteHelp with the following information in your request:

- Select **MS Learn** as the choice for where the change is to be made (1st field on the form).
- Select **Learn module Azure sandbox onboarding** for the Service Category (2nd field on the form).
- A link to the **ModuleWorkItem** in Azure DevOps.
- A link to your pull-request to the associated Learn repo.
- The UID of the module (from `index.yml`).
- The *full list of resources* the module will need to create, listed in `{resource-provider}/{resource-type}` format along with the minimum SKUs required for each service if relevant.
- Estimated time needed to complete the module.
- Any downloads users are expected to execute in the lab, including download links.
- Specific network ports that must be accessible, or other networking requirements.

The form will create a ticket in Azure DevOps and you will receive an email response with a link to that work item. Your lab will then be registered in the database by a member of the Learn team. It will take 1-2 days to complete and the team will reach out if there are any questions.

Step 3: Add unit metadata

Once the module has been registered with the Learn sandbox database, you can add support to each unit that will be using the sandbox with the `sandbox: true` metadata key/value. This needs to be added to each exercise **unit** YAML file as shown below:

```
### YamlMime:ModuleUnit
uid: learn.azure.xyz
title: Some title here
sandbox: true
metadata:
    title: {SEO title}
...
```

Adding this key with a value of **true** to the unit will present a new HTML header on the top of the unit page that allows the user to either activate the sandbox, or see the current time remaining. Here's an example of the prompt before the sandbox has been activated:

This module requires a sandbox to complete. You have used 1 of 20 sandboxes for today. More sandboxes will be available tomorrow.

[Activate sandbox](#)

The sandbox only needs to be activated once in a module session. Once that's done, the header provided on each unit will display the remaining time for the sandbox before it times out and releases all the created resources.

Sandbox activated! Time remaining: 2 hr 34 min

You have used 3 of 20 sandboxes for today. More sandboxes will be available tomorrow.

Identify the sandbox resource group

A single resource group with the name `Learn-{guid}` is created as part of the sandbox activation process. This resource group will be visible in the Azure portal, or through CLI commands such as `az group list`.

You can reference the name of the resource group in exercise instructions using the placeholder `<rgn>[sandbox resource group name]</rgn>`. The `[sandbox resource group name]` text will be shown in the instructions until the sandbox is activated. Once the learner activates a session, the value will be replaced in the page and all subsequent pages in the module with the name of the resource group.

For example, in [learn-pr/azure/design-ip-addressing-for-azure/includes/5-exercise-implement-vnets.md](#) the resource group is used in a code block:

```
35 1. In Azure Cloud Shell, run the following command to create the **CoreServicesVnet**  
   virtual network:  
36  
37     ````azrecli  
38     az network vnet create \  
39         --resource-group <rgn>[sandbox resource group name]</rgn> \  
40         --name CoreServicesVnet \  
41         --address-prefix 10.20.0.0/16 \  
42
```

Use the Azure portal

The Azure sandbox environment provides a real Azure subscription for the learner to work in. The lab instructions can use CLI commands or the Azure portal to perform tasks in Azure. There is no direct integration of the Azure portal in the content, therefore the instructions should ask the learner to open the Azure portal in a new tab or window of the browser once the sandbox is activated.

The screenshot shows a web browser window with the Microsoft Azure portal loaded. The URL in the address bar is <https://docs.microsoft.com/en-us/learn/modules/tour-azure-portal/4-exercise-work-with-blades>. The page title is "Exercise - Work with blades". It indicates "10 minutes" of time required. A note states: "This module requires a sandbox to complete. A sandbox gives you access to Azure resources. Your Azure subscription will not be charged." A button labeled "Sign in to activate sandbox" is present. Below this, a section titled "Activate the Azure sandbox" contains two steps: "1. Start by activating the Azure sandbox above." and "2. Once it's activated, sign into the [Azure portal for sandbox](#). Make sure to use the same account you activated the sandbox with." The entire "Activate the Azure sandbox" section is highlighted with a green border.

To ensure the link is opened in a new tab/window, include the query string `?azure-portal=true` on the URL. In addition, to make sure the student is placed into the proper subscription, add `/learn.docs.microsoft.com` to the end of the URL. For example:

1. Sign into the [Azure portal](<https://portal.azure.com/learn.docs.microsoft.com?azure-portal=true>).

The query string value will be stripped by the rendering engine and is only present to indicate that the URL should be opened in a new tab. You can combine the value with other parameters and it can be in any location of the URL.

Use the embedded Azure Cloud Shell

If the lab exercise needs to perform command-line actions, the unit can activate an embedded [Azure Cloud Shell](#). This option allows learners to use commands and tools as if they were running them in the terminal through [Bash](#) or [PowerShell](#). The embedded shell is the same Cloud Shell environment used in the Azure Portal and includes all the [same tools](#).

The Cloud Shell visibility in the unit is controlled by the `interactive` field, defined in a unit's YAML file:

```
### YamlMime:ModuleUnit
uid: learn.azure.xyz
title: Some title here
sandbox: true
interactive: bash
metadata:
  title: {SEO title}
...
```

The `interactive` field supports three values:

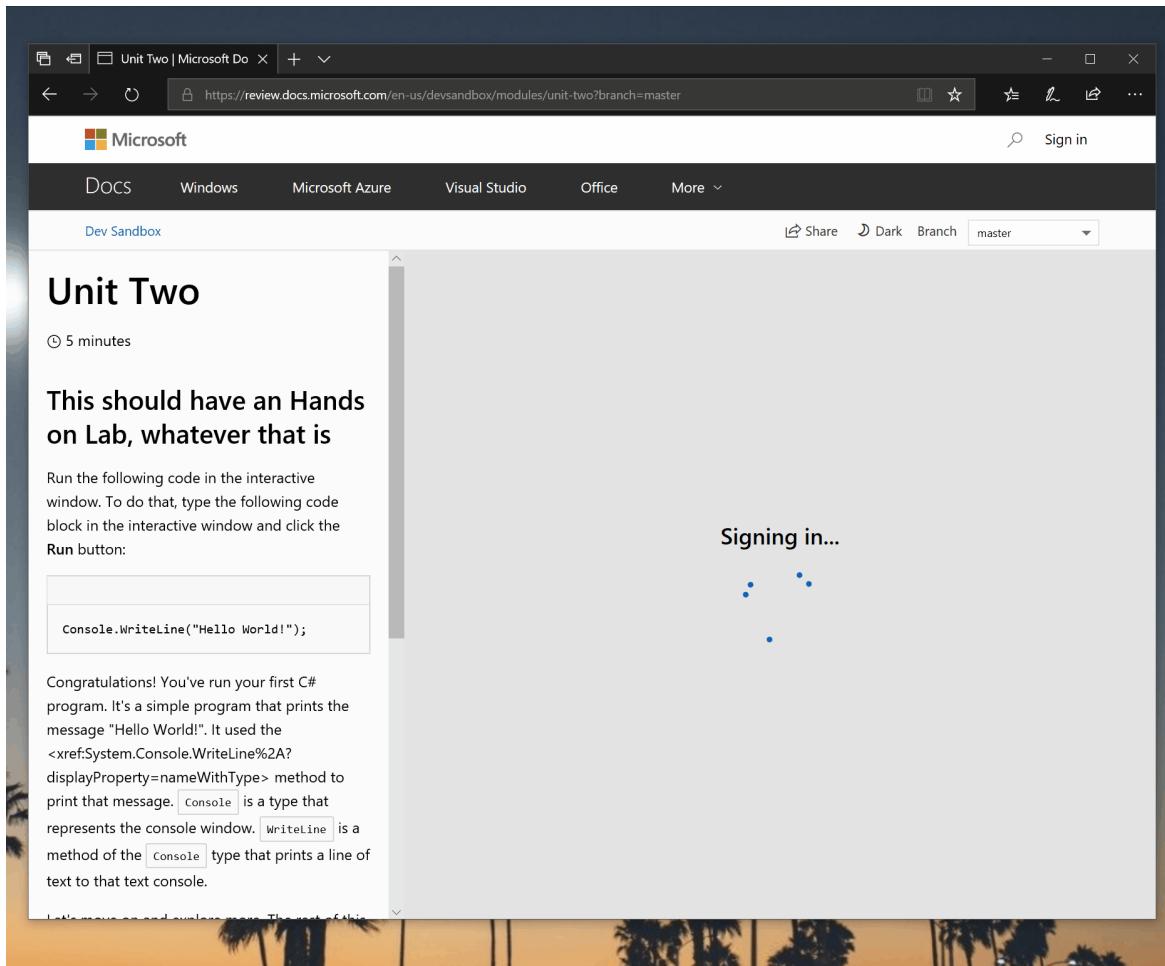
1. `bash` or `powershell` can be used when the learner needs to have focus both on the content and the shell in equal proportion - e.g. you are guiding them through creating Azure resources via the CLI. This splits the screen equally - 50% is given reserved for the shell, and 50% for the instructions.

The screenshot shows a Microsoft Docs page titled "Unit One". On the right side, there is an "Interactive" panel containing an "Azure Cloud Shell" terminal window. The terminal shows the output of a command: "Requesting a Cloud Shell. Succeeded. Connecting terminal... den@Azure:~\$". Below the terminal, there is a code editor window with the following C# code:

```
Console.WriteLine("Hello World!");
```

Below the code editor, there is explanatory text about the program's purpose and how it uses the `Console.WriteLine` method. At the bottom of the page, there is a note about the text being from classical Latin literature.

2. `bash-editor` or `powershell-editor` provides more space to the Cloud Shell editor area.



Step 4: Add Task Validation (optional)

To ensure learners have done the exercise properly, you can add support for [task validation](#). This is optional but highly recommended as it also forces the reader to *complete* the exercise in order to earn the badge.

Step 5: Test and Publish

- Once the team notifies you that the module has been registered and which policy it has been mapped to, you might be asked to test the lab scenario again with the policy in place. As before, make sure to execute all steps for your lab exercise to make sure it's working properly. Here are the specific links to each policy type you can test against:
 - [Web policy](#)
 - [Database policy](#)
 - [Virtual Machine policy](#)
 - [IoT policy](#)
 - [Container policy](#)
- Once you've verified that your labs works with the policy in place, the team will merge any required policy changes into production and register your module with the production database. This will then allow you to test on the formal review site and publish your module to production.

TIP

If you see the message **Module not associated with an account ID**, it means the module UID in `index.yml` doesn't match the one registered in the Learn sandbox database. Check to make sure you are using the correct module UID and if so, email LearnLabsAzSandbox@microsoft.com to fix the mismatch.

Once the module has been published to master, **make sure to test your exercise end-to-end**. This will ensure it's all registered properly and the environment activates.

Understand the Azure sandbox limitations

1/14/2022 • 3 minutes to read

The Azure sandbox was created to help new users get into Azure and allow them to be able do something very quickly with no setup. It's not designed to enable every Azure service or scenario. Its primary use case is for beginner-level scenarios - content that embraces that will likely work fine. There are several limitations on the Azure environment enabled by the Azure sandbox. These restrictions are put in place to control costs and limit fraud, but also impact the types of solutions you can create in a sandbox.

Here are the primary limitations that will potentially affect your lab:

Limited to a single pre-created Azure Resource Group

The sandbox creates a single resource group when it's activated and grants the learner **contributor** access. This resource group is deleted (along with all associated resources) when the sandbox is destroyed. The Azure sandbox cannot support lab experiences that require access beyond the contributor role, multiple resource groups or subscriptions, or objects that can't exist inside resource groups (such as Azure AD tenants).

Limitations on the number of resources created

Learn uses an external service to monitor the resources created by the learner. If the number of resources exceed a threshold, the sandbox is deleted without warning or learner notification. At this time, learners can only create three VMs or containers at a time, and are blocked from creating more than six resources of the same type.

Strictly enforced time limit

Modules are time-boxed from 1-4 hours based on the resources needed. Learners must finish the module before the sandbox times out. Once it times out, all progress is lost. The timer starts when the sandbox is activated and the current remaining time is always shown on units where the `sandbox: true` metadata key/value.

Each module gets a unique sandbox environment

Sandboxes are always destroyed at the end of a module and cannot be reused across modules. To work around this limitation, modules that have a starting point should include a script or starting solution that sets up the expected environment.

Limitation on the number of sandboxes per day

A given MSA can only activate 10 sandbox environments in a 24-hour period. This effectively means a learner can only go through 10 sandbox-based modules per day. The number of sandboxes is cumulative across all sandbox-hosted environments (Azure, D365, Notebooks, etc.)

NOTE

Microsoft employees can activate 20 sandbox environments per day.

Azure App Service limited to Free, Dynamic & Shared tiers

Azure App Service is currently restricted to the Free, Dynamic & Shared service tiers. In addition, there are limits on the number of App Service Linux instances that can be created (this is a restriction of Azure App Service, not the sandbox).

Embedded Cloud shell can timeout

The integrated Cloud Shell will time out after 20 minutes. The sandbox will still be available, and the Cloud Shell can be reactivated but any command-line or environment variables used will be lost. Try to make sure the content doesn't have long units between your interactivity instructions.

Azure Cache for Redis restrictions

The Azure Cache for Redis service is limited to Basic (C0), Standard (C0, C1), or Premium (P1) SKUs.

Virtual Machine restrictions

VMs are the resource that has the most fraud potential, and as such, have the most restrictions. Any module using a VM is always time-limited to a single hour.

Available SKUs

The following VM sizes can be created:

- Standard_A1_v2
- Standard_A2_v2
- Standard_A1
- Standard_A2
- Standard_B1ls
- Standard_B1s
- Standard_B2s
- Standard_B1ms
- Standard_B2ms
- Standard_DS1_v2
- Standard_DS2_v2
- Standard_D2s_v3

VM networking restrictions

All created VM's have network security groups (NSG) applied to control inbound and outbound traffic. When VMs are involved, the NSGs cannot be modified or removed.

Allowed Inbound Traffic

SOURCE	DESTINATION PORT	PROTOCOL	COMMENTS
Any	80, 443, 8080	Any	
Any	22, 3389	TCP	

Allowed Outbound Traffic

DESTINATION	DESTINATION PORT	PROTOCOL	COMMENTS
Any Azure Services except (AzureCloud)	Any	Any	

DESTINATION	DESTINATION PORT	PROTOCOL	COMMENTS
AzureCloud	80, 443	TCP	
GitHub	22, 80, 443, 9418	TCP	
Fastly Edge	80, 443	TCP	Any domain that uses Fastly Edge (including raw.githubusercontent.com) - there might be other blocks that weren't added
Ubuntu	80, 443	TCP	security / keyserver under the ubuntu.com domain
Microsoft CDN Edge (Verizon)	80, 443	TCP	Any domain that uses Microsoft CDN (including api.nuget.org)
NPM JS	80, 443	TCP	

Blocked MSA domains

To prevent abuse, we block sandbox activation by learners whose user accounts are from known disposable email providers. Additionally, we block activation from domains that end with:

- .onmicrosoft.com
- @pokemail.net
- @Invoke.org
- @2mailnext.com
- @for4mail.com
- @student.humg.edu.vn and any other domain ending in .edu.vn

Cannot use the Azure Portal Cloud Shell

The portal-based Cloud Shell tries to create a new resource group for the storage account which is not allowed. Instead, use the embedded Azure Cloud Shell in your unit.

Add task validation to an Azure Sandbox lab (Optional)

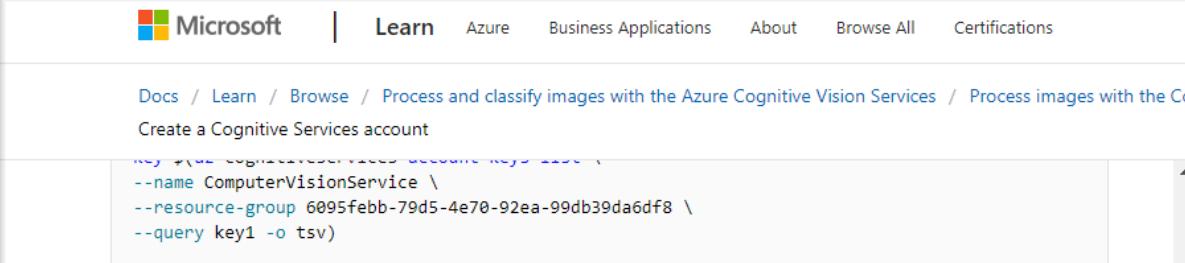
1/14/2022 • 5 minutes to read

Content authors can add task validation to Learn modules that feature an Azure Sandbox lab experience. Learn will check the learner's work in the lab against the expected results defined by the content author. Learn then displays feedback to the learner. This step is optional, but recommended. The step ensures that the learner has completed all the steps requested in the exercise.

IMPORTANT

Adding task validation forces the learner to pass the validation check to receive credit for the module. Omitting this step allows the student to bypass the exercise and still obtain the badge which is undesired.

A unit with task validation includes a **Check Your Work** button at the bottom of the page. Learners use the Azure Sandbox to complete the lab exercises defined in the unit. They then select the button to begin task validation.



The screenshot shows a Microsoft Learn module for "Process and classify images with the Azure Cognitive Vision Services". The URL is [https://docs.microsoft.com/learn/modules/process-images-with-the-azure-cognitive-vision-service/](#). The page displays an Azure CLI command to get keys:

```
az cognitiveservices account key list --name ComputerVisionService --resource-group 6095febb-79d5-4e70-92ea-99db39da6df8 --query key1 -o tsv
```

The Azure CLI 2.0 uses the `--query` argument to execute a JMESPath query on the results of commands. JMESPath is a query language for JSON, giving you the ability to select and present data from CLI output. These queries are executed on the JSON output before any display formatting. The `--query` argument is supported by all commands in the Azure CLI.

In our example, we query the list of keys for an entry named "key1" and output the result to tsv format. This format removes quotations around the string value. We assign the result to a variable key.

Important

We're going to use this key throughout the module, so saving it in a variable is a good idea. If you lose the value or the variable becomes unset, run the command again to set it.

3. To see the value of our key, execute the following command in Azure Cloud Shell:

```
Azure CLI
```

```
echo $key
```

Now that we have an account and a key, it's time to make some calls to the API.

Check your work

A failed validation displays hints to the learner, suggesting how to properly pass the validation.

Adding support for task validation

Task validation steps are added to the unit YAML file as a `tasks` array. This array is typically added to the end of the definition as shown below:

```
### YamlMime:Tutorial
uid: learn.azure.xyz
title: Some title here
...
tasks:
- action: exists
  environment: azure
  azure:
    resourceGroup: TutorialResourceGroupName
    tags:
      - name: msft-learn
        value: true
      - name: unitTitle
        value: Creating resource groups
      - name: unitTask
        value: createresourcegroup
    any_resource_attribute: expected_value_for_that_attribute
    locationid: centralus
  hint: "Make sure the Resource Group name matches 'TutorialResourceGroupName' and it is deployed to the central us datacenter"
- action: not-exists
  environment: azure
  optional: true
  azure:
    resourceGroup: TutorialResourceGroupToDelete
    hint: "The Resource Group 'TutorialResourceGroupToDelete' still exists, this value will need to be removed"
- action: exists
  environment: azure
  azure:
    resourceGroup: "TutorialResourceGroupName"
    resource:
      type: "Microsoft.Storage/storageAccounts"
      name: TutorialResourceTEST
      any_resource_attribute: expected_value_for_that_attribute
  hint: |
    [!include[](includes/resource-group.md)]
```

Explanation of metadata

METADATA	REQUIRED	DESCRIPTION
<code>tasks</code>	No (Yes, if using task validation)	Lab configuration. Each <code>action</code> in the collection can be used to confirm that an Azure Resource <code>exists</code> or does <code>not-exist</code> per the Task Validation - Azure Resource Manager specs.
<code>tasks > action</code>	No (Yes, if using task validation)	Required field; Allowed values: <code>exists</code> : Used to confirm a resource exists. <code>not-exists</code> : Used to confirm when a resource DOES NOT exist.
<code>tasks > environment</code>	No (Yes, if using task validation)	Currently only <code>Azure</code> is supported. No validation of how a resource is managed, only the end state of the resource.

METADATA	REQUIRED	DESCRIPTION
<code>tasks > azure</code>	No (Yes, if using task validation)	Contains Azure resource information for validation.
<code>tasks > azure > resourceGroup</code>	No	Resource group name that contains the resources for validation. If you want to validate a resource group, then this field is required and <code>resource</code> node should not be provided. If the <code>resourceGroup</code> field is not provided, the validation tool would seek the target <code>resource</code> in resource groups.
<code>tasks > azure > tags</code>	No	An array of tags that sits under the conditions node, and is used to validate against the tags associated with a resource: <code>name:</code> The name of the tag. <code>value:</code> Value that is expected.
<code>tasks > azure > tags > name</code>		
<code>tasks > azure > tags > value</code>		
<code>tasks > azure > any_resource_attribute</code>	False	The attribute and expected value Example: <code>locationid: centralus</code> . Validatable resources can be viewed on Azure Resource Manager .
<code>tasks > azure > location_id</code>		
<code>tasks > type</code>	No (Yes, if using task validation)	Required field; A resource type that exists within the environment specified above.
<code>tasks > name</code>	No	Used to confirm the <code>name</code> attribute for the resource that is being validated.
<code>tasks > optional</code>	No	Defines if this task is required to complete the tutorial module. Default is <code>false</code>
<code>tasks > hint</code>	No (Yes, if using task validation)	<p>Response returned to the learner if the validation does not pass, giving advice on what might have gone wrong or what the learner should double check prior to requesting a validation again</p> <p>The hint can either be a string OR an included Markdown file (to allow authors to reuse content) that will be returned to learner.</p>
<code>tasks > conditions</code>	No	Ability to perform additional requirements for validation <code>any_resource_attribute: expected_value_for_that_attribute</code> example: <code>locationid: centralus</code>

METADATA	REQUIRED	DESCRIPTION
<code>tasks > tags</code>	No	An array of tags that sits under the conditions node, and is used to validate against the tags associated with a resource <code>name:</code> The name of the tag. <code>value:</code> Value that is expected.
<code>resource</code>	No	Property set for target resource for validation. If not provided, the system would validate resource group only.
<code>resource > name_and_resource > type</code>	No (Yes, if using task validation)	<p>Name and type of the target resource.</p> <p>Example: <code>name: testSqlServer, type: Microsoft.Sql/servers</code> to validate sql server with name <code>testSqlServer</code>.</p> <p>Example: <code>name: testSqlServer/testDatabase, type: Microsoft.Sql/servers/databases</code> to validate sql server database with name <code>testDatabase</code> on <code>testSqlServer</code>.</p> <p>You can see the name and type are combined to validate a resource. Azure resource ID template is: <code>https://management.azure.com/subscriptions/{subscriptionId}/resourcegroups/{resourceGroupName}/providers/{resourceProviderNamespace}/{parentResourcePath}/{[resourceType]}/{resourceName}</code>. The <code>resourceType/resourceName</code> combination can be recursive.</p> <p>For example, <code>providers/Microsoft.Web/sites/op-dhs-prod-jobs/slots/archive/webjobs/ReportDataGenerator</code></p> <p>This is a webjob named <code>ReportDataGenerator</code> that exists in the <code>archive</code> slot of the <code>ops-dhs-prod-jobs</code> website. For task validation, writers should provide <code>name: op-dhs-prod-jobs/archive/ReportDataGenerator, type: Microsoft.web/sites/slots/webjobs</code></p>

Writing task validation

Pull template of resource properties

Sign into [Azure Resource Manager](#). You can get samples for resources. Usually, you should expand

`subscriptions -> {subscription} -> resourceGroups -> {resourceGroup} -> providers -> {provider type}`. Then, you can preview your resources of that provider.

For example, you can find below json from Azure Resource Manager for resource:

```
/resourceGroups/myResourceGroup/providers/Microsoft.Sql/servers/testSqlServer?api-version=2015-05-01-preview .
```

```
{  
    "kind": "v12.0",  
    "properties": {  
        "administratorLogin": "admin",  
        "version": "12.0",  
        "state": "Ready",  
        "fullyQualifiedDomainName": "testSqlServer.database.windows.net"  
    },  
    "location": "northcentralus",  
    "id":  
    "/subscriptions/XXXXXXXXXXXX/resourceGroups/myResourceGroup/providers/Microsoft.Sql/servers/testSqlServer",  
    "name": "testSqlServer",  
    "type": "Microsoft.Sql/servers"  
}
```

Then, you can write the yaml for validation as:

```
...  
tasks:  
- action: exists  
  environment: Azure  
  azure:  
    resourceGroup: myResourceGroup  
    resource:  
      type: Microsoft.Sql/servers  
      name: testSqlServer  
      kind: 12.0  
      properties:  
        version: 12.0  
        state: ready  
      location: northcentralus  
  hint: |  
    [!include[]](includes/resource-group.md)  
...
```

Basically, you can put all kinds of sections into a validation yaml except ID, name, and type. The system validates resources with provided sections `existing` or `not existing`.

Testing task validation

We're still developing the front end for task validation. Follow the method below to test task validation in the meantime:

1. Create a module unit in your Learn GitHub repo with validation tasks. You can create a branch for this work.
2. Create relative resources for validation on [Azure](#) under a subscription that is owned by a Microsoft organization
3. Modify the subscription to grant read permission for `TripleCrownSmokeTestApp`. The Application ID is `4d59a4c9-2da3-4885-a775-612703674e87`.
 - a. Go to https://ms.portal.azure.com/#blade/Microsoft_Azure_Billing/SubscriptionsBlade.
 - b. Select the subscription you're testing with.
 - c. Select **Access Control (IAM)**.
 - d. Select **+ Add**.
 - a. For **Role**, select `Reader`.
 - b. For **Assign access to**, don't change this value.

- c. Select `4d59a4c9-2da3-4885-a775-612703674e87`.
 - d. Select **Save**.
4. Go to [docs learning backend ppe site](#)
 5. Expand the `/units/{unitId}/progress` section.
 6. Select **Try it out**.
 7. Fill in the unit ID, branch, and payload. The payload format would be similar to the following example:

```
[  
 {  
   "azureSubscription": "<learner azure subscription id. The subscription contains all the resources  
   for validation>",  
   "type": "FreeAzureSubscription"  
 }  
]
```

8. Select **Execute** and verify the output in the **Response** section.

Add a Lab to a unit using a hosted VM lab experience

1/14/2022 • 6 minutes to read

Modules with a VM lab give the learner access to a secure, pre-configured virtual environment to complete Learn Lab exercises. VM labs are hosted by our VM lab partner, Learn on Demand Systems (LODS).

IMPORTANT

Learn on Demand Systems (LODS) is the only lab vendor authorized to host VM labs for MS Learn.

Content authors can embed access to a VM lab environment within one or more units of a Learn module. To launch the VM, the learner must authenticate using an MSA account. Task validation is currently not supported for VM labs.

When the VM is activated, Learn launches a new browser window that displays a LODS VM. The exercise unit containing the lab instructions and other lab resources is displayed in an iframe side-by-side with the VM in the same browser window.

Once the learner completes the module (or after the lab's time limit has elapsed), LODS automatically cleans up the resources and removes the learner's access.

Who can create a VM Lab for use on Learn?

Learn content authors (or their approved vendors) can request to create their own VM labs directly on LODS's platform, or they can hire LODS to create VM labs on their behalf.

- Read [Creating your own LODS VM Learn Labs](#) to learn more about creating your own VM labs for your Learn content.
- Read [Working with LODS to create a VM Learn Lab](#) to learn more about working with LODS to create VM labs for your Learn content.

Adding a LOD based lab experience to a Learn module

Once you've created a lab experience (VM or Cloud Slice), you will get an assigned *lab id*. This is a numeric identifier used to interact with the lab on the LOD admin portal. Each unit that will leverage the lab environment must have a `labId` identifier set in the unit YAML:

```
### YamlMime:ModuleUnit
uid: learn.{product}.{folder}
title: 'Exercise: TBD'
labId: {lab-id-goes-here}
durationInMinutes: 10
```

Adding this metadata will cause Learn to add a new header onto the unit with a button to launch the VM experience.

This unit requires a VM to complete.

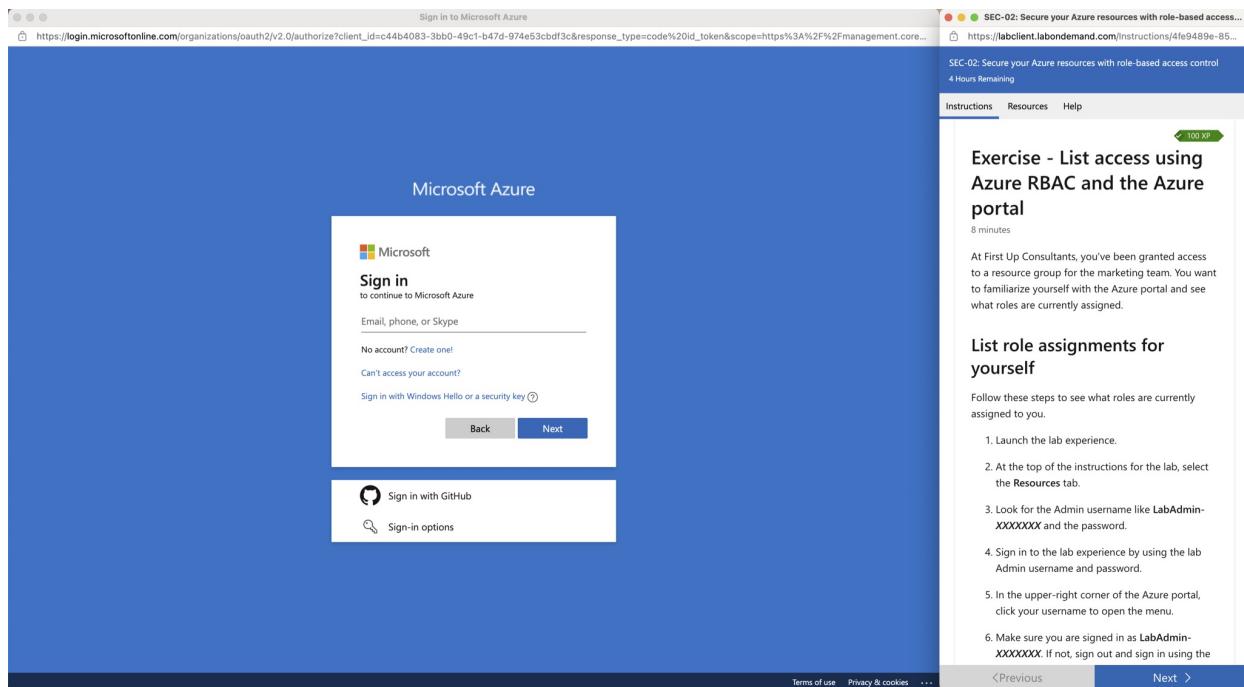
VM Mode provides a free, web-based virtual machine environment to complete the steps in this unit.

[Launch VM mode](#)

Clicking the button will, by default, launch the VM *in-place*, replacing the current unit's contents with the VM tab. You can adjust that behavior by adding a `labModal` flag into the unit YAML.

```
### YamlMime:ModuleUnit
uid: learn.{product}.{folder}
title: 'Exercise: TBD'
labId: {lab-id-goes-here}
labModal: true
```

This flag will change the behavior to launch the VM or Cloud Slice lab into a separate window. In both cases, the exercise instructions will be displayed on the side of the VM experience. Here is the full-screen (separate window) experience.



The LOD environment also provides sign-in credentials. Unlike the Azure Sandbox, the currently logged on user's MSA is *not* used in this environment. Instead, fabricated credentials are provided that are setup on the VM itself. You can see the credentials on the **Resources** tab:



SEC-02: Secure your Azure resources with role-based access control

3 Hr 59 Min Remaining

Instructions

Resources

Help

Azure Portal

URL <https://portal.azure.com/#home>

Subscription 92a5590d-8d7f-44d1-9a46-4e7fab67784e

Username LabUser-
16773158@triplecrownlabsoutlook.onmicrosoft.com

Password G*5e!pM0Dj

Username LabAdmin-
16773158@triplecrownlabsoutlook.onmicrosoft.com

Password G*5e!pM0Dj

Resource Group

FirstUpConsultantsRG1-lod16773158

VM lab costs

Review the [Learn Lab funding model](#) for details.

VM lab security & abuse prevention

VM Labs are a perpetual target for abuse and misuse for a variety of purposes, including crypto currency mining, bot networks, piracy, identity theft, denial of service attacks, etc. Such abuse can rapidly escalate to tens or hundreds of thousands of dollars in Azure consumption in less than a day. The root cause of this abuse is often misconfiguration or misapplication of security policies and restrictions.

Content authors and Learn share responsibility for preventing lab abuse. Please review Learn's [Shared Security Model](#) for Learn Labs.

Modules with VM labs must have the lab environment QA'd by LODS prior to going live on Learn. Modules at high risk for abuse may require additional QA by Learn prior to being allowed to go live.

LODS VM lab content guidelines and limitations

LODS's lab platform is very flexible and can support a wide variety of lab experiences for Microsoft products. Please see below for specific content guidelines & limitations for Learn.

Can LODS support my product or lab experience?

Need a consultation call with LODS to scope & plan your lab experience? Click [here](#) to contact Quinn Lamb, Learn's LODS Account Manager, to request a discovery meeting.

Documentation for LODS's [LabOnDemand platform](#) is available on their website.

Lab unit instructions will be pulled directly from Learn

IMPORTANT

Do not write lab content directly in LabOnDemand's IDLx platform.

In a VM Lab, Learn presents the contents of your Learn module in an iframed window, side-by-side with the VM environment. The lab exercise units from the Learn module are served directly from Learn, and learners can progress from one lab exercise unit to the next in the lab.

LODS's LabOnDemand platform includes a lab instruction authoring tool called "IDLx." Do not write lab content or exercises directly in IDLx.

One VM lab per Learn module

Multiple modules cannot share the same VM lab, a VM lab can only be used for a single module.

Using a VM Lab across multiple units in the same module

A VM Lab can be used across multiple consecutive units in the same module, but there are some limitations.

1. The lab units must be in consecutive order, and not interspersed with other unit types. For example, a module with a lab in units 2-5 can share the same VM lab across all 5 units. A module with a lab in units 2,3, and 5 with a knowledge check in unit 4, is not supported.
2. Include the `labId:` metadata tag in the YML file for every unit that is part of the lab experience, or the learner will not be able to navigate from one unit to the next in the lab experience. The `labId:` tag is case sensitive.

Time limits

The default time limit is 4 hours. This time limit can be configured for each lab. If you change the duration, make sure lab duration time is longer than the expected module duration time.

We do not currently allow learners to extend the time limit to help control lab abuse.

Task validation is not supported

Learn does not currently support [task validation](#) for Learn VM labs hosted by LODS. However, content authors can validate learning via a quiz.

Because lab units must be in consecutive order, add the quiz to a **separate** unit following a lab exercise units. **Do not include a quiz in a lab exercise unit**, it may render incorrectly.

Basic Troubleshooting

- "Lab not published to API" error message displayed when trying to launch lab in a live (published) module
 - Usually a result of embedding the lab ID for the lab's Dev profile in the `labId:` tag in the unit YML.

- Confirm correct lab ID in the unit YML, replace with lab profile's Production lab ID
- If LODS built your lab for you, work with your LODS contact to fix.
- If you or another vendor created the lab, follow the guidance [here](#)
- Lab instructions don't render correctly (or at all) on Learn's Review site
 - This is normal behavior. The lab instructions won't display correctly until the module is published & live on Learn.
 - The lab instructions are pulled directly from the published module on the Learn live site and rendered in the lab environment. Until the module is published & live on Learn, there is nothing that can be rendered in the lab environment.
- Lab instructions don't render correctly (or at all) when a lab is launched from a live (published) module.
 - Usually a result of a missing, or incorrectly configured, URL in the "Resources" tab of the lab profile.
 - If LODS built your lab for you, work with your LODS contact to fix.
 - If you or another vendor created the lab, follow the guidance from the "[Resources Tab](#)" section of the [Creating your own LODS VM Learn Labs](#) article in Learn Guide to configure correctly.
- "Lab not available" error message on when launching a lab from a live (published) module.
 - LODS API is returning a failure message to Learn
 - Typically, this is because the lab profile has been removed or disabled in Learn's Production tenant.
 - Could also indicate a general problem/outage of LODS's platform
 - [Contact LODS Support](#)

Need Help from LODS?

Need help with your LODS VM lab?

1. Open a [LODS Support Ticket](#)
 - Enter Organization Name as "Microsoft Learn- <your product name here>"
2. Email LODS support: support@learnondemandsystems.com
 - Include your ticket number

Work with LODS to create a hosted VM Lab

1/14/2022 • 4 minutes to read

Learn on Demand Systems (LODS) offers Microsoft Learn content authors a full-service lab creation program on a fee-for-service basis. If you'd like to employ LODS for VM lab creation, contact [Quinn Lamb](#), Learn's LODS Account Manager, directly for consultation and a quote.

NOTE

You must [submit and track](#) all VM labs alongside their respective Learn modules in Azure DevOps. Plan your lab as part of your overall Learn content planning and proposal process.

VM lab costs

Review the [Learn Lab funding model](#) for details.

Overview

You'll experience five major phases in the LODS VM lab creation process. The timeline and steps to create a new VM lab will vary depending on:

- The complexity of the lab environment.
- The length of the lab experience.
- How much interaction is required between LODS and the content author to produce the final production version.

Phase 1 - Discovery

Can LODS support my lab?

In Phase 1, as the content author, you have a high-level idea of the module you want to write and the lab experience you need to enable. Before you begin more detailed module design and planning, you can engage with LODS to confirm that Learn can support your lab experience.

TIP

Need a discovery meeting with LODS? Email [Quinn Lamb](#).

Phase 2 - Get Started

Learn about planning, development timeline, cost estimation, and billing setup.

In Phase 2, you write and submit two artifacts:

- Your module design document.
- The initial design for the lab exercises you expect the learner to finish in the module.

Engage with LODS to plan the creation of your lab experience. In the process, you'll understand the costs to create the lab and the estimated Azure consumption. You'll also create an SOW for the work.

Then, you'll send LODS a copy of your completed module design document and a completed [VM Learn Lab Intake Form](#).

Now you can write the content for your Learn module.

Phase 3 - Lab development

In Phase 3, you work with LODS to create, test, and sign off on the lab experience for your Learn module. In this phase, you hand off the written Learn module content, including the lab exercises, to LODS. LODS begins developing the lab experience.

Allow at least two to three weeks to finish this phase. Allow more time if your lab environment is complex.

LODS will build a functional prototype of the lab and review it with you. Together, you'll confirm that you've met all of the functional requirements. LODS then incorporates your feedback to build the final release-ready version of the lab and sends you a launch link for final QA, sign off, and testing.

NOTE

The lab exercises and instructions will not display correctly when the VM lab launches. This is the expected behavior until your module is live on Learn. The lab renders the Learn website to display the exercises in the lab environment.

Phase 4 - Embed your lab in your module and prepare to publish

In Phase 4, LODS has:

- Delivered the final release-ready version of the lab.
- Cloned the lab profile into Learn's Production tenant.
- Sent you the production launch URL for the lab.

You're ready to publish the lab with its Learn module. You embed the VM lab in your Learn module, and prepare to publish the new module and lab to Learn.

Next, embed your lab in your Learn module. All you have to do is add a metadata key value pair at the top of the YAML file for each unit in the module that has lab exercises. For example, if your module has three units of lab exercises, edit the YAML files for all three units.

1. Find the Lab ID in the production lab's launch URL. It's the five-digit code at the end of the URL.



`https://labondemand.com/LTI/Launch/12345`

2. Open the YAML file for the first unit page.

3. Find the metadata at the top of the page.

4. Add the `labId:` metadata tag, followed by the five-digit Lab ID.

The `labId:` tag is case-sensitive.

5. Save the file and repeat for each unit.

Here's an example of the YAML metadata for a unit with lab exercises. The Production Lab ID is `45597`.

```
### YamlMime:ModuleUnit
uid: learn-sandbox.sample-module-3.unit-2
title: Sample Unit 2 with lab url
metadata:
  description: Sample Unit 2 Description
labId: 45597
durationInMinutes: 12
content: |
  [!include[](includes/unit-1.md)]
```

NOTE

Because your module is not live on Learn yet, the lab exercises and instructions will not display correctly when the VM lab launches. This is the expected behavior.

Phase 5 - Publish to Learn and final test

In Phase 5, you:

- Confirm LODS has published your Learn module and its VM lab experience to Learn.
- Do one final test to confirm:
 - The VM lab launches as expected.
 - The lab experience correctly displays the lab exercises as you expect.

IMPORTANT

Before publishing your module to Learn, confirm that:

- LODS has pushed the lab to Learn's production environment.
- You added the lab id to the YAML files for each unit that contains lab exercises.

1. Navigate to your new Learn module, then to the first unit with the VM lab embedded.
2. Select **Launch VM** and wait for the LODS VM to load.
3. Confirm that the lab loads.
4. Confirm that the lab exercises display correctly in the iframe window.
5. If there's more than one unit of lab exercises, scroll to the bottom of the first page, select **Next** and confirm that the content for the next Learn unit loads as expected.

TIP

If a unit page fails to load, check that the `labId:` metadata tag is present in that unit's YAML file, and includes the Production Lab ID. Remember, the `labId:` is case-sensitive.

DEV-03: Add Machine Learning to your UWP Apps
3 Hr 59 Min Remaining

Instructions Resources Help

Store.

- Splash screen: SplashScreen, appears when your app starts.
- Lock screen: LockScreenLogo, appears on your app's lock screen.



We reviewed our UWP app template. Now let's use the Windows ML API and built-in Visual Studio automation to add a pretrained machine learning model to our app.

<Previous Next >

6. Select **Previous** and confirm that the content from the previous Learn unit loads as expected.

Congrats! Your new Learn module and VM lab are now live!

Contact LODS

Need help with your LODS VM lab?

- Create a [LODS support ticket](#).

In the ticket, set **Enter Organization Name** to *Microsoft Learn - <your product name here>*

- Email [LODS support](#).

Include your LODS support ticket number

Create your own hosted VM Labs with LODS Lab on Demand

1/14/2022 • 14 minutes to read

Overview

Microsoft Learn lets authorized content authors and their selected vendors create their own Learn on Demand Systems (LODS) VM labs for their Learn content within Learn's LODS Dev tenant. This article describes:

- The process for registering as a Learn VM Lab Author.
- How to request access to Learn's LODS tenant.
- Key lab profile configurations.
- And more.

NOTE

You must [submit and track](#) all VM labs alongside their respective Learn modules in Azure DevOps. Plan your lab as part of your overall Learn content planning and proposal process.

VM lab costs

Review the [Learn Lab funding model](#) for details.

Register as an MS Learn VM lab author

All content authors and vendors wanting to create their own labs must register as an Learn VM lab author and request access to Learn's LODS Dev tenant.

1. Submit the [MS Learn: VM Lab Author Registration Form](#).



2. Allow for up to one week for your request to be fully processed.

- 24-48 hours: Access request reviewed by Learn, sent to LODS for processing.
- 72-96 hours: LODS processes access request, sets up organizational structure and permissions, and activates Dev tenant access.

Rules

IMPORTANT

Learn reserves the right to revoke all access to LODS resources for any person, at any time. Our priority is to deliver a high-quality user experience, ensure overall content quality and platform security, and prevent abuse of lab resources.

To keep access to Learn's LODS tenant, the content authors and their vendors should follow these guidelines. As

the content author, you're accountable for making sure your vendors review and follow these guidelines. You're also responsible for your vendors' overall use of Learn's tenant.

- Submit all new or significantly revised lab profiles to LODS.
 - They have to do a QA check for security and abuse.
 - Do so before publishing the related module(s) to Learn.
- Don't publicly share links to any lab profile in Learn's LODS Dev tenant.
- Don't directly embed raw lab URLs in Learn content.
- Follow guidance and best practices for creating secure and high-quality labs experiences. They're updated from time to time, so don't assume you know them.
- Incorporate configuration or content changes resulting from the QA of lab profiles.
- Maintain your modules and labs. Keep them up to date over time.
- Respond to feedback to resolve lab issues quickly. Feedback can come from either learners or from Learn.
- Don't misuse Learn's LODS tenant. Don't use Learn resources to develop labs that aren't meant for Learn.
Don't serve Learn Labs to multiple platforms.
- Follow Learn's content and instructional design guidelines, including the duration of the lab experience.

LODS lab authoring

Caution

DO NOT use LODS's IDLx tool for lab authoring.

Learn doesn't support IDLx. You should write all lab exercise steps and instructions in your Learn module. The system will pull this content directly from Learn and render it side-by-side with the lab environment.

Learn doesn't offer VM lab authoring support. For assistance with lab creation, refer to LODS's documentation and reference [Learn's preferred configurations](#). If you require assistance, contact [LODS support](#) directly.

- [LODS VM Lab Development Articles](#)
 - [Build a Windows Virtual Machine](#)
 - [LODS VM Lab Best Practices](#)
- [LODS "Cloud Slice" Azure Lab Development Articles](#)
 - [Azure Resource Template Best Practices](#)
- [LODS Help Docs](#)

Set up your lab profile for use on Learn

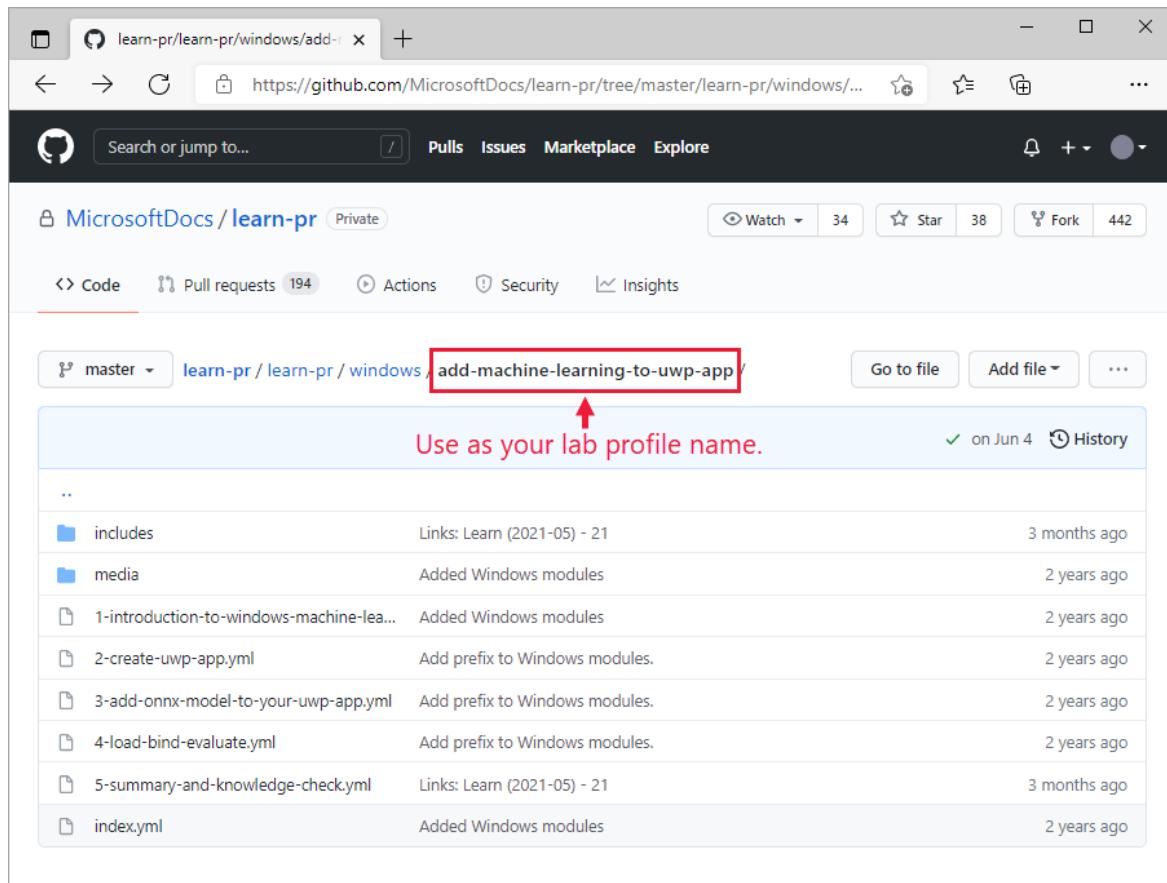
Below are the basic configurations for each tab of a LODS lab profile. Use them unless you're instructed to deviate by LODS or Learn.

Basic Information tab

Configure the basic information settings for your lab profile:

- **Number** - Product and incremental lab number. The field has a maximum of 50 characters. Example:
Dynamics CRM - 01.
- **Name** - Same as the name of the module's folder in your Learn GitHub repo.

Example:



- **Series** - This setting is flexible. Use it for tagging, and so on. Example: **Microsoft 365 Enterprise Admin Cert**.
- **Organization** - Same as your GitHub repo's name. Choose from the list.
- **Development Status** - Set to **In Development** until the lab is ready to be QAed by LODS. Then set it to **Awaiting Verification**.
- **Expected Duration** - 240 minutes.
- **Maximum Duration** - 240 minutes.
- **Description** - Add a meaningful description of your lab.
- **Instructions** - Select this option.
- **Enabled** - Select this option.
- **Owner Name** - Enter lab author's name.
- **Owner Email** - Enter the lab author's email address.

Virtual Machines tab

Configure VMs to suit the needs of the lab. Allocate an appropriate amount of resources (compute, memory, and storage) to support a good user experience. Follow the [LODS best practices](#). Be frugal with VM resources, but not so much that it results in a negative user experience or poor impression of our products.

Guidelines

Here are a few guidelines:

- Disable these options in Windows:
 - Windows Updates
 - Medic Service
 - Windows Defender

- Telemetry service
- Firewalls
- Set the background to a solid color.
- In Windows, under **System Properties > Performance Options**, set it to **adjust for best performance** and select **smooth edges of screen fonts**.
- Only connect to the internet if it's necessary for the scope of your lab. It's understood that you may require internet connectivity.
- Only use NAT network for internet. **DO NOT** use static IPs for labs on Learn.

Preferred settings

Here are some preferred settings:

- Basic Information tab
 - **Memory** - Not less than 8 GB, but enough to support a good user experience.
 - **Processors** - Not less than four cores, but use enough so the CPU isn't pegged in the lab.
 - Set resolution to 1024x768 to make it compatible across the widest array of devices.
 - To enable Dynamic Screen Resizing, install LOD Integration Services (requires DVD drive) and then, in the VM, run the application that is on the ISO.
 - Set VM heartbeat to 15 minutes.
 - Set inactivity timeout to 30 minutes.
- Networks Adapters tab
 - Set the MAC Address to **Static**.
 - Assign a static MAC Address.

Cloud tab

Configure the cloud settings for your lab profile:

- Orchestration
 - **Cloud Platform** - Azure
 - **Override Client Landing Page** - Clear this option. Don't select it.
 - **Append Lab data** - Clear this option. Don't select it.
 - **Subscription Pool** - Select **MS Learn - Dev Pool**.
 - **Deployment Failure** - Send User Notification
 - **User Accounts** - Add user accounts as needed.
 - **Name Prefix** - Use *LabUser-* for user accounts, and *LabAdmin-* for admin accounts.
- Cloud Resource Groups
 - Configure resources groups as needed for your lab.
 - Access Control Policies:
 - Use **Deny All** if the learner isn't expected to deploy any resources during the lab after the Azure Resource Manager deployment.
 - If the learner does need to deploy resources during the lab, they need more granular access control.

Resources tab

Configure the resources settings for your lab profile:

- **Name** - Give it a name. For example, use the name of the Learn module.
- **Type** - External Link
- **URL** - The URL requires a specific format:

- Paste in the URL to the first unit of your module that has lab instructions.

The screenshot shows a browser window with the URL <https://labondemand.com/LabProfile/Edit/48495>. The page title is "Edit Lab Profile" and the sub-page title is "Add Machine Learning to your UWP Apps". A modal dialog is open under the "Resources" tab, which is highlighted with a red box. The dialog contains fields for Name (TC Content), Type (External Link), URL (<https://docs.microsoft.com/en-us/learn/modules/add-m>), Description (Lab content), and Lab Manual (checkbox). The URL field is highlighted with a red box.

- Delete the locale from the URL.

For example, `en-us` is the locale in this unit URL.

The screenshot shows the same browser window and modal dialog as the previous one, but with a red box around the URL field. The URL now reads <https://docs.microsoft.com/learn/modules/add-m>. A red text overlay on the right side of the dialog says "Delete locale from unit URL." At the bottom of the dialog, there are "Save", "Save As", and "Cancel" buttons.

- Paste this *chromeless* extension to the end of your unit page URL.

```
context=context%2Fchromeless&FromOrigin=https://labclient.labondemand.com
```

It allows LODS to render the lab exercise content side by side with the lab environment.

Example of Unit URL with the chromeless extension:

```
https://docs.microsoft.com/learn/modules/add-machine-learning-to-uwp-app/2-create-uwp-app?
context=context%2Fchromeless&FromOrigin=https://labclient.labondemand.com
```

- Description** - Lab guide

- Lab Manual** - Select this option.

Advanced tab

Configure the advanced settings for your lab profile:

- Theme** - None.

- **End Redirect URL** - None.
- **Max Active Instances** - Set to 1.
- **Show Timer**: Select this option.
- **Enable Navigation Warning**: Select this option.
- **Show Navigation Bar**: Select this option.
- **Navigation Bar Width**: Leave the default.
- **Show Instructions tab**: Select this option.
- **Show Resources tab**: Select this option.
- **Show Help tab**: Select this option.
- **Custom Instructions tab Label**: Clear this option. Don't select it.
- **Custom Resources tab Label**: Clear this option. Don't select it.
- **Custom Help tab Label**: Clear this option. Don't select it.
- **Enable Type Text (if displayed)**: Select this option.
- **Show Virtual Machine Power Options**: Select this option.
- **Require Hyper-V Enhanced Controller**: Clear this option. Don't select it.
- **Enable Instance Link Sharing**: Clear this option. Don't select it.

Save/Cancel Options

- **Allow User to Cancel Labs**: Select this option.
- **Allow User to Save Labs**: Clear this option. Don't select it.
- **Allow User To Extend Time By**: Clear this option. Don't select it. It's not supported.
- **AutoSave Incomplete Labs**: Clear this option. Don't select it.
- **Save/Cancel Labs When Last Console Sync Exceeds**: Select this option. Set it to 15 minutes.
- **Save/Cancel Labs When Last Activity Exceeds**: Select this option. Set it to 30 minutes.

LTI

- **Scoring Policy**: Time Spent
- **Scoring Format**: % Complete
- **Time**: 15 minutes

Test your lab

NOTE

Because your module is not live on Learn yet, the lab exercises and instructions won't display correctly when the VM lab launches. This is normal expected behavior.

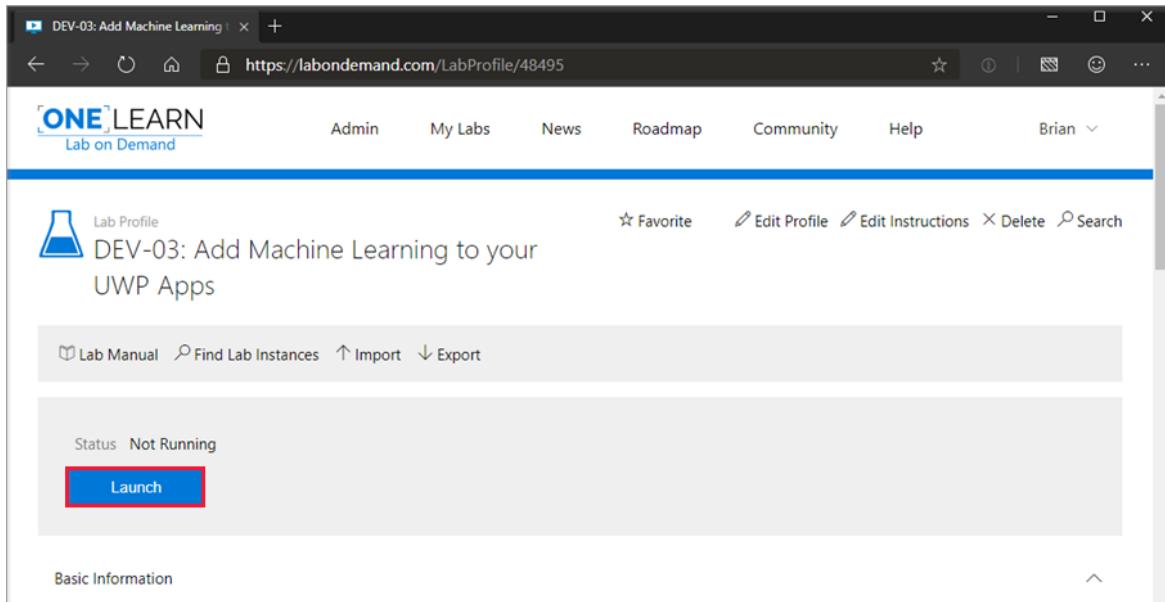
Once you've created your lab profile and configured for your lab experience, you can sign in and test it directly from the LODS portal.

1. Go to [Learn on Demand Systems](#) and sign in.
2. Find and open your lab profile.

3. In your lab profile, select **Launch**.

NOTE

The system pulls lab instructions from Learn and won't render correctly until Learn published your module and it goes live. This is normal expected behavior.



Security and abuse prevention QA

IMPORTANT

Modules with VM labs must have the lab environment QAed by LODS prior to going live on Learn. Modules at high risk for abuse may require additional QA before Learn allows them to go live.

Learn's shared security model

Content authors and Learn share responsibility for preventing lab abuse. Review [Lab security and preventing abuse](#) to learn more.

QA process

The final step before publishing your lab to Learn is to submit it to LODS. They'll then do a security and abuse QA. LODS will confirm that the lab is properly configured with proper policies and restrictions in place. Doing so prevents malicious actors from abusing the lab once it's live on Learn. The cost for the QA is \$250 per lab. Learn will fund this cost for new Learn Labs created in FY20.

Allow one to two weeks to finish the QA process.

1. Contact [Quinn Lamb](#), Learn's LODS Account Manager, to request the QA review.
2. Include links to each LODS Lab profile you want to submit to the QA process.
3. LODS will finish a review of the lab profile and configurations, and provide feedback about changes needed before publishing to Learn. LODS completes this QA on a first-come, first-served basis.

NOTE

LODS won't enable your lab in production until you've addressed all critical QA feedback.

4. LODS will now push your lab profile from Learn's Dev tenant to our Production tenant.
5. LODS will send you a new launch URL for your production lab. This URL includes a five-digit unique Lab ID. Make note of this Production Lab ID.
6. Your lab is now ready to publish with your Learn module, but you must first embed the lab in your module before publishing to Learn.

What does LODS check in the QA process?

LODS typically checks for the following items in the QA process:

- Does the lab use an Azure Resource Manager (ARM) template? If so does it properly deploy the necessary resources?
- Does the lab have restriction policies in place?
 - Do those policies allow the learner to create all necessary resources in the lab?
 - Do those policies successfully prevent the learner from creating any resources outside the scope of the lab? If not, you need to adjust the policies to only allow the learner to create needed resources.
- Does the lab follow the principle of "least possible privilege?" Meaning that you should configure the lab to allow the minimum permissions and resources required to finish the lab exercises as written.
- Does the lab use the fewest resources possible for items that the learner created? For example:
 - If the lab needs a VM, should an ARM template create it, or can the learner create it?
 - Does the VM use the fewest resources possible to achieve the learning scope of the lab?
- Do all resources created in the lab delete successfully when the learner cancels the lab? If not:
 - Is it a LOD platform issue?
 - Is it an issue with the Azure API?
- If the lab creates a public facing internet network, is it needed? Could you rework the lab so it doesn't require public internet access?
- If the VMs are LODS-hosted and require internet access:
 - Did you place them on a secure network?
 - Does that network mitigate abuse?
- Is the lab profile configured to use a public/static IP? LODS doesn't allow Public IPs in self-paced labs. Static IPs are only allowed on a case-by-case basis.
- Is the maximum duration of the lab configured correctly to follow Learn's guidelines to ensure the learner can't leave the session open or running for longer than intended?
- Is **Save** disabled on the lab profile?
- Is the option to extend the time remaining on the lab profile disabled?
- Are the heartbeat and inactivity timeouts configured correctly? The Defaults are 15 minutes for the heartbeat, and 30 minutes for the inactivity?

Embed your lab in your Learn module

Caution

Do not embed raw lab launch URLs directly in Learn content.

Get the right Lab ID: Development vs. Production

LODS generates a unique Lab ID for each lab profile created on their platform. Learn uses these Lab IDs to call the right lab profile when a learner launches a lab from a Learn module.

When a user creates a lab profile in Learn's Dev tenant, it's automatically assigned a Dev Lab ID. First LODS QAs the lab and to make sure it's ready to be published to Learn. Then they'll clone the lab profile into Learn's Production tenant. The cloning process will automatically create a **new** Lab ID for the production version of the lab.

As a Learn lab author, you don't have access to view the Production lab profile. LODS limits your access to Learn's Dev tenant. You can only view the Dev Lab ID in the lab profile.

When publishing your module to Learn, you must use the **Production** Lab ID or your lab won't launch. LODS will provide you with the launch URL to the version in production. Find the Lab ID in the production lab's launch URL (five-digit code at the end of the URL).



<https://labondemand.com/LTI/Launch/12345>

Contact [LODS support](#) if you need to confirm the Production Lab IDs for your labs.

Set up the unit YAML files

Next, embed your lab in your Learn module. All you have to do is add a metadata key value pair at the top of the YAML file for each unit in the module that has lab exercises. For example, if your module has three units of lab exercises, edit the YAML files for all three units.

1. Find the Lab ID in the production lab's launch URL. It's the five-digit code at the end of the URL.
2. Open the YAML file for the first unit page.
3. Find the metadata at the top of the page.
4. Add the `labId:` metadata tag, followed by the five-digit Lab ID.

The `labId:` tag is case-sensitive.

5. Save the file and repeat for each unit.

Here's an example of the YAML metadata for a unit with lab exercises. The Production Lab ID is `45597`.

```
### YamlMime:ModuleUnit
uid: learn-sandbox.sample-module-3.unit-2
title: Sample Unit 2 with lab url
metadata:
  description: Sample Unit 2 Description
labId: 45597
durationInMinutes: 12
content: |
  [!include[]](includes/unit-1.md)
```

Publishing to Learn

IMPORTANT

Before publishing your module to Learn, confirm that:

- You have completed a security and abuse prevention QA with LODS and fixed any issues identified in the review.
- LODS has pushed the lab to Learn's production environment.
- You added the [production lab id](#) to the YAML files for each unit that has lab exercises.
- You have added and configured the Learn URL in the **Resources** tab of the Lab Profile, and have selected the **Lab Manual** option.

Testing your lab after publishing to Learn

Finish one final test to confirm your lab launches as expected, and the lab experience correctly displays the lab

exercises as you expect.

1. Navigate to your new Learn module, then to the first unit with the VM lab embedded.
2. Select **Launch VM** button and wait for the LODS VM to load.
3. Confirm the lab loads.
4. Confirm the lab exercises display correctly in the iframe window.
5. If there's more than one unit of lab exercises, scroll to the bottom of the first page, select **Next** and confirm that the content from the next Learn unit loads as expected.

TIP

If a unit page fails to load, check that the `labId:` metadata tag is present in that unit's YAML file, and includes the Production Lab ID. The `labId:` is case-sensitive.

The screenshot shows a Microsoft Learn module titled "DEV-03: Add Machine Learning to your UWP Apps" with "3 Hr 59 Min Remaining". The "Instructions" tab is selected. The main content area displays a list of tasks:

- Store.
- Splash screen: SplashScreen, appears when your app starts.
- Lock screen: LockScreenLogo, appears on your app's lock screen.

Below this is a screenshot of the Visual Studio Solution Explorer for a project named "MNIST Demo (Universal Windows)". The Solution Explorer shows files like App.xaml, MainPage.xaml, and MainPage.xaml.cs. A callout box highlights the "Properties" node under the project node. At the bottom of the module content, there is a summary:

We reviewed our UWP app template. Now let's use the Windows ML API and built-in Visual Studio automation to add a pretrained machine learning model to our app.

At the very bottom are navigation buttons: "<Previous" and "Next >".

6. Select **Previous** and confirm that the content from the previous Learn unit loads as expected.

Disabling a lab that is live on Learn

Occasionally, it may be necessary to temporarily disable access to a lab without disabling or removing the entire Learn module.

1. Go to [Learn on Demand Systems](#) and sign in.
2. Find and open your lab profile.
3. Scroll to the bottom of the **Basic Information** tab.
4. Clear the **Enabled** option.

The screenshot shows a web-based form for editing a lab profile. The URL in the browser is <https://labondemand.com/LabProfile/Edit/48495>. The form includes fields for Development Status (In Development), Expected Duration (240 Minutes *), Maximum Duration (240 Minutes *), Language (English), Level (100), Evaluation (None), Advertising Campaign (None), and a large Description area containing the text "Microsoft Learn hosted environment for demonstrating Windows Development.". Below the description is an Instructions section with a checked checkbox for "Use Legacy Format (not recommended)". The Introduction Content URL field contains a placeholder URL. A "Show for at least" dropdown is set to 30 seconds. The "Enabled" checkbox is checked and highlighted with a red border. The Owner Name is listed as "TC Content" and the Owner E-mail is "someone@microsoft.com". At the bottom are three buttons: Save (highlighted in blue), Save As, and Cancel.

5. Select **Save**.

It may take a few minutes for the production version of the lab to reflect this change.

TIP

To re-enable the Lab, re-select the **Enabled** option and select **Save**.

Contact LODS

Need help with your LODS VM lab?

- Create a [LODS support ticket](#).

In the ticket, set **Enter Organization Name** to *Microsoft Learn - <your product name here>*

- Email [LODS support](#).

Include your LODS support ticket number

The Learn Dynamics 365 sandbox

1/14/2022 • 3 minutes to read

The Dynamics 365 sandbox allows a module to spin up a pre-configured D365 environment with sample data for sales and customer scenarios and give a user access to it for a short period of time to perform a set of exercise steps.

NOTE

Activating a D365 sandbox involves placing a real user into the tenant. We currently hide other users, and all organizational-level information is restricted as it would potentially show other activity happening in the tenant. For this reason, scenarios that involve *writing* to the organization are prohibited in this sandbox experience.

There are several steps to prepare your content to use the D365 sandbox.

1. Verify your lab scenario can run in the D365 sandbox.
2. Submit a registration request.
3. Add unit metadata.
4. Test and publish the module.

Step 1: Verify your lab can run in the sandbox

The only D365 product available is Sales, but more will be added over time. If the module is working with a different product, the current solution is to use an [LOD-based VM](#).

If you want to explore the pre-supplied data, you can [activate the pre-production D365 experience](#).

Step 2: Submit a registration request

Once you've determined the lab scenario is supportable, you must submit a request to register your module in the Learn sandbox database. This will make your module known to the system and provide the necessary UX to activate the experience when that unit is displayed.

1. Fill out a [Sandbox Module Onboarding Request Form](#) in SiteHelp with the following information in your request:
 - Select **MS Learn** as the choice for where the change is to be made (1st field on the form).
 - Select **Learn module Dynamics sandbox onboarding** for the Service Category (2nd field on the form).
 - A link to the **ModuleWorkItem** in Azure DevOps.
 - A link to your pull-request to the associated Learn repo.
 - The UID of the module (from `index.yml`).
 - Any downloads users are expected to execute in the lab, including download links.

The form will create a ticket in Azure DevOps and you will receive an email response with a link to that work item. Your lab will then be registered in the database by a member of the Learn team. It will take 1-2 days to complete and the team will reach out if there are any questions.

Step 3: Add unit metadata

Once the module has been registered with the Learn sandbox database, you can add support to each unit that

will be using the sandbox with the `sandbox: true` metadata key/value. This needs to be added to each exercise unit YAML file as shown below:

```
### YamlMime:ModuleUnit
uid: learn.dynamics.xyz
title: Some title here
sandbox: true
metadata:
  title: {SEO title}
...
```

Adding this key with a value of **true** to the unit will present a new HTML header on the top of the unit page that allows the user to either activate the sandbox, or see the current time remaining. Here's an example of the prompt before the Sandbox has been activated:

This module requires a sandbox to complete. You have used 1 of 20 sandboxes for today. More sandboxes will be available tomorrow.

[Activate sandbox](#)

The Sandbox only needs to be activated once in a module session. Once that's done, the header provided on each unit will display the remaining time for the Sandbox before it times out and releases all the created resources.

Sandbox activated! Time remaining: 3 hr 54 min

Launch the Dynamics 365 Sales sandbox and use the instructions on this page to complete the activity.

[Open sandbox in new window ↗](#)

You have used 1 of 20 sandboxes for today. More sandboxes will be available tomorrow.

Step 4: Test and Publish

Once the team notifies you that the module has been registered with the database, you will be able to test your module on the review site.

TIP

If you see the message **Module not associated with an account ID**, it means the module UID in `index.yml` doesn't match the one registered in the Learn sandbox database. Check to make sure you are using the correct module UID and if so, email LearnLabsAzSandbox@microsoft.com to fix the mismatch.

Once the module has been published to master, **make sure to test your exercise end-to-end**. This will ensure it's all registered properly and the environment activates.

The Try.NET experience

1/14/2022 • 2 minutes to read

MS Learn has an integrated C# Read-Execute-Print-Loop (REPL) experience based on Try.NET. When activated, students can type C# programs or statements into the hosted environment and execute them directly in the browser as shown below.

The screenshot shows a web-based development environment. At the top, there are navigation links ('< Previous' and 'Next >'), a title 'Unit 2 of 7', and a progress bar indicating '100 XP'. Below this is a section titled 'Exercise - "Hello World!"' with a duration of '5 minutes'. A descriptive text states: 'In this first hands-on exercise, you'll use C# to print a hallowed programmer's phrase to the Output window.' To the right is a '.NET Editor' pane containing the C# code 'Console.WriteLine("Hello World!");'. A green 'Run' button is located at the top right of this pane. Below the editor is an 'Output' pane showing the result 'Hello World!'. There is also a note at the bottom left: 'We'll explain how and why it works soon. But first, you should see it running, and execute your code.'

Adding support for Try.NET

Try.NET visibility in a unit is controlled by the `interactive` field, defined in a unit's YAML file:

```
### YamlMime:ModuleUnit
uid: learn.azure.xyz
title: Some title here
sandbox: true
interactive: csharp
metadata:
  title: {SEO title}
...
```

Setting the `interactive` key to `csharp` provides the Try.NET experience. Note that this is stateless - there is no storage provided.

Placing code blocks in the content

There are two supported approaches to showing code in the exercise.

Inline code

Use the `csharp-interactive` tag on inline code blocks to get a Run button. Here's an example:

```
```csharp-interactive
Console.WriteLine("Hello World!")
```
```

This will render in content with a **Run** button:

C#

 Copy

 Run

```
Console.WriteLine("Hello World!");
```

Clicking the Run button will paste the code into the REPL and execute it automatically.

NOTE

Inline code will always default to *method* execution. This approach only allows statements to be entered into the editor. If your exercise requires full program execution, you need to use the code snippets approach below.

Code snippets

The `:::code:::` extension can be used to pull in code fragments from files co-located with the content. This approach also allows full program execution (single-file).

```
:::code source="relativepathtosomefile.cs" interactive="try-dotnet-class":::
```

The `interactive` keyword should be set to:

- `try-dotnet-class` to enable students to enter a complete C# class into the editor.
- `try-dotnet-method` or `try-dotnet` to only allow statements (which are then put into a method) to execute.

For more information, see [Code snippets included by reference](#).

Class execution

Class execution places the typed content into the following template:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Globalization;
using System.Text.RegularExpressions;

namespace Program
{
    class Program
    {
        /* Content will be placed here */
    }
}
```

Method execution

Method execution places the typed content into the following template:

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Globalization;
using System.Text.RegularExpressions;

namespace Program
{
    class Program
    {
        static void Main(string[] args)
        {
            /* Content will be placed here */
        }
    }
}
```

Publish the content

Once the module has been published to master, **make sure to test your exercise end-to-end**. This will ensure it's all registered properly and the environment activates.

Guidance for creating technical exercise content

1/14/2022 • 4 minutes to read

Exercise content is a fantastic way to provide learners with explicit instruction to complete tasks. Coupling exercise content with the Learn Sandbox environment is one of the key features of Microsoft Learn. Not all services are supported in Sandbox, so alternatives such as Labs on Demand and bring-your-own-subscription may be alternatives. When creating technical exercise content, there are some recommendations to follow to improve the experience for learners. This doc details some best practices that we've found over the course of content creation.

General recommendations

Use the `<rgn>[Sandbox resource group name]</rgn>` text replacement to automatically populate the resource group when using the Learn Sandbox. Avoid setting this to a variable, as that could be lost if the cloud shell resets.

Avoid using too many variables, especially in a long module with exercises spanning multiple units. These will get reset if the cloud shell times out, and can make it hard to populate. If you do use variables, populate them at the start of each unit.

Use multi-line commands to eliminate side scrolling. For example, a long command such as below:

```
az vm create --resource-group test1 --name myVM1 --image "/subscriptions/fe5185df-4a60-4a88-a00a-cb649a3da69a/resourceGroups/mygallery-rg/providers/Microsoft.Compute/galleries/mygallery/images/mygallery-imgdef" --admin-username adminuser --admin-password <password>
```

Is much easier to read when formatted as a multi-line command, like below:

```
az vm create\  
  --resource-group test1 \  
  --name myVM1 \  
  --image "/subscriptions/fe5185df-4a60-4a88-a00a-cb649a3da69a/resourceGroups/mygallery-rg/providers/Microsoft.Compute/galleries/mygallery/images/mygallery-imgdef" \  
  --admin-username adminuser \  
  --admin-password <password>
```

When using multi-line commands put items that require the text to be modified on the last line. This allows the learner to backspace and replace in the shell since only the last line will be editable. This is especially useful if there's only a single item that needs to be replaced, often a password.

If you need to send the learner to a URL, generate a clickable URL to make it easy for the learner, as in the following example:

```
echo http://$(az network public-ip show \  
  --resource-group <rgn>[Sandbox resource group]</rgn> \  
  --name appGatewayPublicIp \  
  --query dnsSettings.fqdn \  
  --output tsv)
```

Use the watch command to continuously monitor the output of a command, like the following example. This also prevents the cloud shell from timing out.

```
watch --difference --interval 5 az network vnet-gateway list \  
--resource-group <rgn>[Sandbox resource group name]</rgn> \  
--output table
```

Don't add unnecessary parameters. If a command has optional parameters and you're not using them specifically for your command, don't include them.

Default to not specifying the region on commands. Most commands will inherit from the resource group, so this becomes an additional parameter that isn't necessary. Use the region only when you need to deploy resources into a specific Azure region.

When deploying resources via script, try to do it through an ARM template over a script if possible. Most resources can be deployed through template, and this allows for easier code reuse, and is not language/os/machine dependant.

Azure CLI recommendations

Use all caps for variable names like the following example, this is a Linux convention.

```
SERVERNAME='myfavoriteserver'  
USERNAME='adminuser'  
echo $SERVERNAME  
echo $USERNAME
```

Use jquery to filter out results from a command. This is a great way to refine the default columns that will come back from commands. For example, the following command lists virtual machines in a resource group, only displaying the name, provisioning state, and power state:

```
az vm list \  
--resource-group ce321c4c-ef4d-4763-a60e-ef242353e71e \  
--show-details \  
--query "[*].{Name:name, Provisioned:provisioningState, Power:powerState}" \  
--output table
```

| Name | Provisioned | Power |
|------------|-------------|------------|
| AppServer | Succeeded | VM running |
| DataServer | Succeeded | VM running |

Use the full `--parameter` instead of the abbreviated versions, like the following example. This helps add clarity to what the command is doing.

```
# Use this  
az network public-ip show --resource-group MyResourceGroup --name MyIp  
  
# Not this  
az network public-ip show -g MyResourceGroup -n MyIp
```

Azure PowerShell recommendations

Use lower or camel-case variable names, like the following example. This aligns more with PowerShell convention.

```
$vmsize = "standard_nv6"  
$vmname = "vm1"
```

Azure portal recommendations

Use the *Next step action* blue button link to make the portal link or cloud shell link obvious. The following is an example of next step action to open the cloud shell.

Azure Cloud Shell

```
> [!div class="nextstepaction"]  
> [Azure Cloud Shell](https://shell.azure.com/?azure-portal=true)
```

Use the [deploy to Azure button](#) to deploy resources from an ARM template.

Bring your own subscription (BYOS) modules (that don't support sandbox)

Because the learner will be footing the bill for these resources, provide up front transparency as to the costs of the resources. Create an estimate in the Azure Pricing Calculator with the resources deployed in the exercise so the learner can see their estimated costs in their currency. [See this unit](#) as an example of how this works.

Use shell.azure.com whenever possible to simplify environment setup and ensure a consistent experience for your exercise.

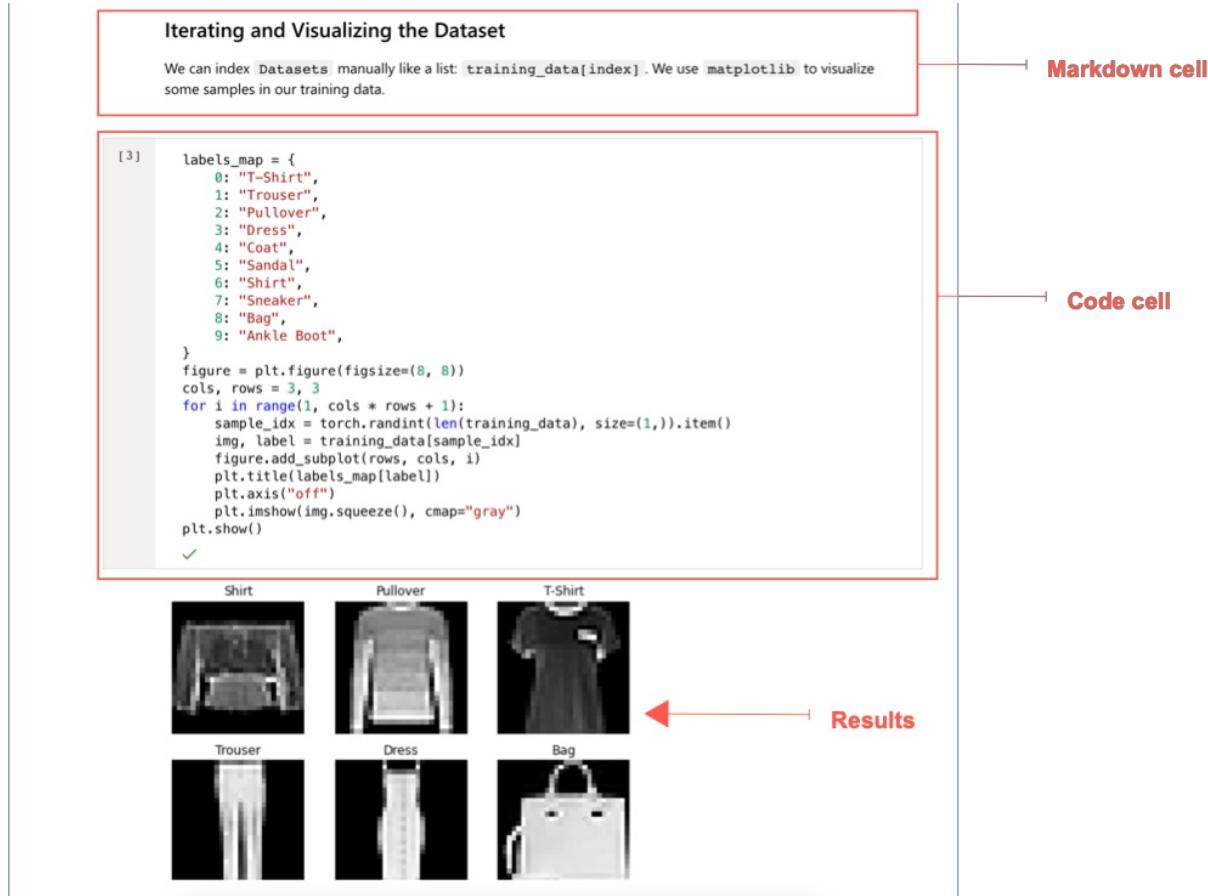
Because it's a normal subscription, you'll need to include the resource group creation in your steps. Use this single resource group for all subsequent commands to make it easy to clean up at the end of the module.

Make sure to clean up the resources when done. If you created everything in a single resource group, you should just need to delete the resource group. If that's not the case for your module, ensure you detail the steps to delete all resources that were created.

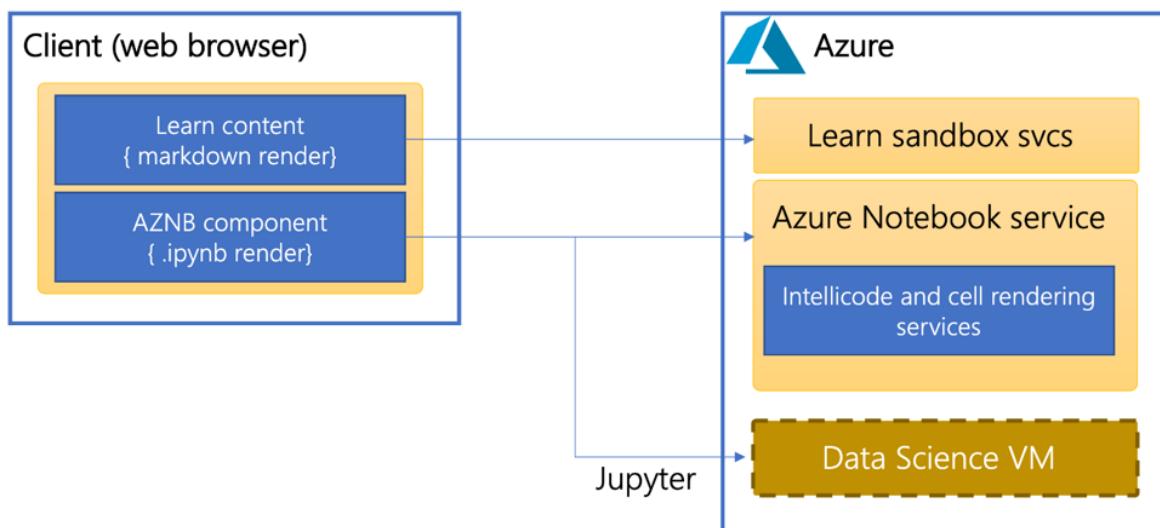
The Learn Notebooks sandbox

1/14/2022 • 7 minutes to read

The Jupyter Notebooks sandbox allows a module to present a Jupyter Notebook as the content for a single unit. The notebook provides the exercise instructions and coding areas as interactive Markdown and Code cells in the notebook, which can be modified and executed as part of the client experience.



There are several moving parts to this environment.



The web browser renders the unit page containing some placeholder content, including a title, estimated time,

and the sandbox activation prompt.

The screenshot shows a Microsoft Learn module titled "What are Tensors?". At the top, there's a navigation bar with "Learn", "Products", "Roles", "Learn TV", "Certifications", "FAQ & Help", and a progress bar indicating "LEVEL 10" with 19250/49599 XP. Below the navigation is a breadcrumb trail: "Docs / Learn / Browse / Get started with PyTorch / Introduction to PyTorch / What are Tensors?". There are "Previous" and "Next" links, and a "Bookmark" icon. The main content area has a title "What are Tensors?", a duration of "10 minutes", and a note about requiring a sandbox. A button labeled "Activate sandbox" is visible. A "200 XP" badge is in the top right corner.

This information all comes from the YAML - just like any other Learn module. The remainder of the content is the Azure Notebooks client component (AZNB), rendering the assigned notebook:

The screenshot shows the Azure Notebooks interface with a notebook titled "Tensors". The toolbar at the top includes icons for back, forward, refresh, and file operations. The notebook content starts with a cell containing "%matplotlib inline" and a heading "Tensors". The text explains that tensors are specialized data structures similar to arrays and matrices, used for model inputs and outputs. It also notes that tensors are similar to NumPy arrays but can run on GPUs. A code cell below shows the import statements "import torch" and "import numpy as np". The section "Initializing a Tensor" is shown with the text: "Tensors can be initialized in various ways. Take a look at the following examples:"

This content is rendered on the *client side* using an associated server-side Azure Notebook service. The notebook is fully editable, supports Intellisense, and is interactive without activating the sandbox. At this point, no compute resources have been created or dedicated to the notebook.

Like other interactive experiences in Learn, the notebooks sandbox requires a Microsoft account (MSA). The sandbox must be activated to *execute* a notebook cell. If the learner hasn't started the sandbox, the page will prompt them to do so if they attempt to run a cell in the notebook. Once activated, an Azure Virtual Machine (VM) will be created and assigned to the sandbox to execute the notebook cells, as shown in the above diagram. This VM is created in a Learn subscription and free to the user.

NOTE

Unlike the Azure sandbox, the Notebook virtual machine is not directly accessible through an Azure subscription to the user. It's spun up in a separate subscription, and the user is not given any access to it. It's specifically for use by the notebook component and not intended for direct user access.

Unit transitions

Unlike other Sandbox environments, each unit (notebook) in this environment is independent. Environmental changes such as installing packages flow across notebooks, but generally speaking, each notebook is executed independently and has variables locally scoped to that notebook. The notebook will always use the same VM and share data with the `%store` magic command to push the state in the global area. For more information, see the [store magic command](#).

Supporting a Notebooks sandbox

There are several steps to prepare your content to use the Notebooks sandbox.

1. Verify your lab scenario can run in the Notebooks sandbox.
2. Submit a registration request.
3. Add unit metadata and identify a location for the notebook.
4. Select a specific kernel for your notebook.
5. Test and publish the module on the review site.

Step 1: Verify your lab can run in the sandbox

Generally speaking, most notebook-based content should render correctly. The Notebook client will render the notebook experience before the sandbox is activated. There are two areas to keep in mind as you plan a notebook:

1. Ensure the proper Python packages are available.
2. Downloading assets needed for the notebook.

The Jupyter server is an [Azure Data Science Virtual Machine \(DSVM\)](#) running on Ubuntu. It has all the pre-loaded tools and environments available to that image. Currently, that is:

1. Python 3.6 or 3.7
2. PyTorch
3. TensorFlow

You can check the [documentation](#) to get specifics on what's available.

TIP

You can also activate a notebook in the pre-production environment [here](#) and use various commands to determine current packages installed and kernels available. For example `!pip list` or `!conda env list`.

Installing python packages

The VM image doesn't have any extra frameworks or python packages installed beyond the standard DSVM. Plan on having a step in your notebook to `!pip install` any updated requirements or additional packages you need before importing them. This step can be done with a shell command. For example, to upgrade to PyTorch 1.8.1 you can have a cell that executes:

```
!pip install pytorch==1.8.1
```

You can also use `conda install` if the package is available from Anaconda. PIP and Conda installs are made to the active environment on the Jupyter server - any change will persist for the *entire* module (all future units).

Downloading assets

If the notebook uses data or additional scripts, you will need to download it into the notebook before the step where they are required. You can use the `wget` command to retrieve data or scripts onto the VM.

```
!wget https://xyz
```

IMPORTANT

The DSVM has the same Network Security Group restrictions we place on all other Azure VMs used in Learn. This configuration means the places you can *retrieve* data from are limited. For example, most CDNs like S3 and Google Drive are not allowed today. Make sure to test the address to see it's an allowed source before assuming the notebook can download the data. GitHub is always allowed, so as a last resort, you can make the script or data available from GitHub and then download it as a raw URL.

Step 2: Submit a registration request

Once you've determined the lab scenario is supportable, you must submit a request to register your module in the Learn sandbox database. Registration will make your module known to the system and provide the necessary UX to activate the experience when that unit is displayed.

1. Fill out a [Sandbox Module Onboarding Request Form](#) in SiteHelp with the following information in your request:
 - Select **MS Learn** as the choice for where the change is to be made (1st field on the form).
 - Select **Learn module Jupyter Notebooks sandbox onboarding** for the Service Category (2nd field on the form).
 - A link to the **ModuleWorkItem** in Azure DevOps.
 - A link to your pull-request to the associated Learn repo.
 - The UID of the module (from `index.yml`).
 - Any downloads users are expected to execute in the lab, including download links.

The form will create a ticket in Azure DevOps, and you will receive an email response with a link to that work item. Your lab will then be registered in the database by a member of the Learn team. It will take 1-2 days to complete, and the team will reach out if there are any questions.

Step 3: Add unit metadata

Once the module has been registered with the Learn sandbox database, you can add support to each unit using the sandbox with the `sandbox: true` metadata key/value. This metadata must be added to each exercise **unit** YAML file as shown below:

```
### YamlMime:ModuleUnit
uid: learn.pytorch.intro-machine-learning.tensors
title: Some title here
sandbox: true
metadata:
    title: {SEO title}
...
```

Adding this key with a value of **true** to the unit will present a new HTML header on the top of the unit page that allows the user to activate the sandbox or see the current time remaining.

In addition, you will need to specify the public URL where the notebook is located. This value must be assigned to the `notebook` property in the unit YAML file:

```
### YamlMime:ModuleUnit
uid: learn.pytorch.intro-machine-learning.tensors
..
sandbox: true
notebook: https://raw.githubusercontent.com/MicrosoftDocs/pytorchfundamentals/main/intro-to-pytorch/2-tensors.ipynb
```

The URL specified here must be absolute, use `https://`, and be public. Today that means it can't be surfaced directly from docs since the hostname changes based on the environment. The current suggestion is to create a [MicrosoftDocs](#) repo and place all your notebooks for the module in that repo - using the raw public URL as the URL for the metadata.

Step 4: Select a specific kernel for your notebook

Jupyter notebooks are JSON files with metadata to provide hints as to how to execute them. The main document's metadata section provides a *kernel*/hint used to decide which Anaconda environment to use to run the notebook. You should ensure this value is set to a supported environment - otherwise, the user will be given an error when the VM is attached to the notebook and required to select a kernel.

Here's an example from Pytorch:

```
{
    "cells": [ ... ],
    "metadata": {
        "kernelspec": {
            "name": "conda-env-py37_pytorch-py",
            "language": "python",
            "display_name": "py37_pytorch"
        },
        "language_info": {
            "codemirror_mode": {
                "name": "ipython",
                "version": 3
            },
            "file_extension": ".py",
            "mimetype": "text/x-python",
            "name": "python",
            "nbconvert_exporter": "python",
            "pygments_lexer": "ipython3",
            "version": "3.7.9"
        }
    }
}
```

The specific fields you need to adjust are:

| PROPERTY | DESCRIPTION |
|--------------------------------------|---|
| <code>kernelspec.name</code> | This value should be set to the specific <code>conda-env</code> environment name for the DSVM. See below for options. |
| <code>kernelspec.display_name</code> | Set this value to the matching display name from the DSVM. See below for options. |
| <code>version</code> | Set this value to the Python version being used - based on the kernel selected. |

Here are the names to choose from. If you are doing TensorFlow or PyTorch, use those dedicated environments, otherwise use the base Python 3.7 image, and install any required packages.

| WHAT YOU GET | NAME | DISPLAY_NAME |
|-------------------------|---|------------------------------|
| Python 3.7 (base) | <code>conda-env-py37_default-py</code> | <code>py37_default</code> |
| Python 3.7 + TensorFlow | <code>conda-env-py37_tensorflow-py</code> | <code>py37_tensorflow</code> |
| Python 3.7 + PyTorch | <code>conda-env-py37_pytorch-py</code> | <code>py37_pytorch</code> |

In all cases, the current Python version is `3.7.9`.

Step 4: Test and publish

Once the team notifies you that the module is registered with the database, you can test your module on the review site.

TIP

If you see the message **module not associated with an account ID**, it means the module UID in `index.yml` doesn't match the one registered in the Learn sandbox database. Check to make sure you are using the correct module UID, and if so, email LearnLabsAzSandbox@microsoft.com to fix the mismatch.

Once the module has been published, **make sure to test your module end-to-end**. This step will ensure it's all appropriately registered and the environment activates.

Add zone pivots to a unit

1/14/2022 • 5 minutes to read

Zone pivots allow content authors the ability to provide two different versions of content, within the same unit - most notably to allow users to "choose their preference" of what content to read, such as Windows vs. Linux exercise instructions.

Learner experience

When learners view a unit with zone pivots, they will see a grey box near the top of the unit with the zone pivot selection and a prompt.



Here's an [example Markdown unit with a C# and PowerShell pivot](#)

Selecting a specific pivot will show all content from the Markdown source that is **not** in a pivot block (referred to as *shared* content), and all content contained in a pivot block with the selected id (in the above example 'Bash'). Content in other pivot blocks, such as 'PowerShell' will be hidden on the page. Note that the content is *always* rendered in the source HTML, it's simply hidden with CSS styles based on the pivot setting.

Add zone pivots to your content

Identify the zone pivot you should use

1. Open the [zone-pivot-groups.yml](#) in the root of the [learn-pr](#) GitHub repository. This file defines the zone pivots for all Learn content.
2. Each defined pivot has four elements defined in the YAML:
 - **id** - a unique identifier, this is a simple name such as 'platform' that defines the group. This is what you'll reference in the module units that use the pivot.
 - **title** - a title for the group.
 - **prompt** - a text prompt shown as the header for the pivot
 - **pivots** - an array of choices for the pivot with an *id* and *title*. The titles are shown in the pivot UI as selectable buttons, and the id is used in the content to designate a section of content that belongs in that pivot.

Here's an example:

```
- id: platform
  title: Platform
  prompt: Choose your platform
  pivots:
    - id: linux
      title: Linux
    - id: macos
      title: Mac
    - id: windows
      title: Windows
```

3. Find the zone pivot group that you would like to use in your content based on the prompt and pivot choices. If you don't see one that serves your needs, you will need to create or edit one (see below for steps on that approach). Remember these are global so try to find one you can use if possible!
4. Note the unique **id** of the group you want to use - 'platform' in the above example.

Add zone pivot support to your unit

Zone pivots are applied on a *unit* basis. You can use different pivots in each unit, or have some units have a pivot, while others do not. To add support for a pivot zone:

1. In your **unit** YAML file, add the following `zone_pivot_groups` key in the `metadata` section as shown here and assign it the **id** value of the pivot group you want to use in that unit. This must be done *in every unit* of the module where the pivot will be used. Here's an example:

```
### YamlMime:ModuleUnit
uid: learn.xyz
title: Title XYZ
metadata:
  zone_pivot_groups: platform
  ...
```

The value specified must a valid pivot zone group name from the global `zone-pivot-groups.yml` file.

2. In the Markdown content file for the unit, use the following syntax to reference the zone pivot groups and specify what content to pivot on.

```
## Section 1

Generalized content for both Windows and Linux. Always shown.

::: zone pivot="windows"

## Windows section
Content specific for Windows. Only shown when "Windows" is selected.

::: zone-end

::: zone pivot="linux"

## Linux section
Content specific for Linux. Only shown when "Linux" is selected.

::: zone-end
```

Each `::: zone` block must be terminated with a `::: zone-end` marker. You can have as many blocks as necessary in the content. When the content is rendered, zone blocks will be shown and hidden at runtime on the client (using `div`s) based on the selected zone. A single selection at the top of the unit will show and hide all associated blocks on the page.

Zone pivots containing indented content

If your zone pivot will *only* contain indented code, such as a code block for a numbered exercise instruction, the zone pivot syntax also needs to be indented for that content to render properly.

```
1. Step describing something done on multiple operating systems with an OS-specific pivot for the actual commands.
```

```
::: zone pivot="windows"

```bash
md hello-world
cd hello-world
code .
```

::: zone-end

::: zone pivot="linux"

```bash
mkdir hello-world
cd hello-world
code .
```

::: zone-end

...
```

Mixing pivots and pivot groups

When using pivot zones, try to avoid duplicating content as much as possible. The zone pivot feature allows you to combine pivots together - giving you flexibility in defining when content is visible.

Combining pivots (OR)

If needed, you can render content based on a selection of one or more 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 iOS pivot or the MacOS pivot were selected:

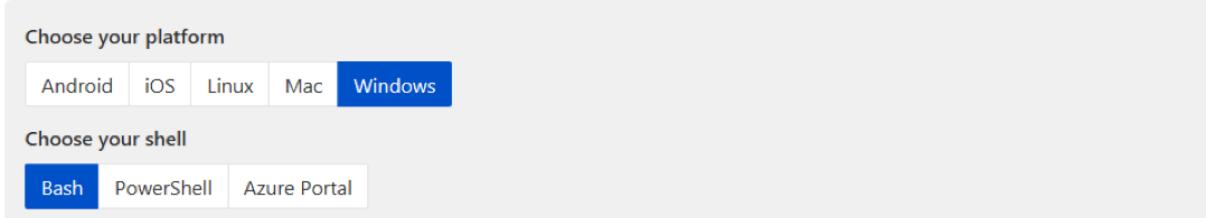
```
::: zone pivot="linux,macos"
## Unix like instructions
Content specific to Linux and Mac.
::: zone-end
```

Using multiple zone groups

You can use multiple zone pivots by including two or more values in the unit metadata. For example:

```
zone_pivot_groups: platform-mobile,azure-interface
```

Would produce *two* independent pivot controls:



Content can then mix the values to show and hide content as needed:

```
::: zone pivot="windows"
## Windows-only, any shell
::: zone-end

::: zone pivot="bash"
## Any platform, Bash only
::: zone-end
```

Combining pivots across groups (AND)

IMPORTANT

Zone pivots can't be embedded in other zone pivots.

Comma-separated zone pivot values that come from different pivot groups are treated as **AND** conditions. For example, to create a block of content that would be visible when the Windows *platform* pivot was select **AND** the Bash *shell* pivot was also selected:

```
::: zone pivot="windows,bash"
## Content visible for Windows + Bash
::: zone-end
```

Here's another example - Android *platform* **AND** Azure portal *shell*.

```
::: zone pivot="android,portal"
## Content visible for Android + Portal
::: zone-end
```

You can mix the AND and OR conditions.

```
::: zone pivot="android,ios,bash,powershell"
## Android OR iOS _AND_ Bash OR PowerShell
::: zone-end
```

NOTE

The syntax for relationships (OR vs. AND) is identical. The difference is in the pivots selected. If the specified pivots are in the same zone group, then either one will match (OR). If the pivots are in *different* groups, then the *both* pivots must be selected (AND).

Engineering Documentation

To view the zone pivot Engineering spec, view the [article on New Hope](#).

Create a new zone pivot group (if needed)

1/14/2022 • 2 minutes to read

IMPORTANT

Creating new zone pivots is not common. The system already defines several pivot groups - this section provides details on how to define a new zone pivot, but authors are encouraged to use an existing one if possible.

The specified pivot in the Markdown content must be defined in the `zone-pivot-groups.yml` defined in the [learn-pr](#) folder of the [learn-pr GitHub repository](#).

Zone pivots work in branches and pull-requests on the review site, so make sure to test your pivot before it's merged into master.

1. Navigate to the [zone-pivot-groups.yml](#) file.
2. Review the existing zone pivot groups to ensure that one does not already meet your needs. If one does, do **not** create a new one. Review the information above how to use the existing one in your content.
3. To create a new group, you will need to add a new block to the YAML file that includes the following information. The pivots will be shown in the order listed, and the first one will always be shown as default.

```
- id: <unique-id-for-the-zone-pivot-group>
  title: <Zone pivot group title that is not shown to users>
  prompt: <A short statement that prompts the user why they need to pick an option. This string will be shown above the switcher.>
  pivots:
    - id: <zone-pivot-id-1>
      title: <Name of pivot that is shown to users.>
    - id: <zone-pivot-id-2>
      title: <Name of pivot that is shown to users.>
    - id: <zone-pivot-id-3>
      title: <Name of pivot that is shown to users.>
```

Example:

```
- id: azure-interface
  title: Shell
  prompt: Choose your shell
  pivots:
    - id: bash
      title: Bash
    - id: powershell
      title: PowerShell
    - id: portal
      title: Azure Portal
```

4. Submit a PR and tag a Learn repo manager for approval and request to merge (@MicrosoftDocs/msft-learn-repo-managers).

Edit an existing zone pivot group

IMPORTANT

Zone pivot groups are shared assets in content, similar to includes files. Make sure that the change you are making is appropriate for all other content that may be using this group.

1. Submit a PR to the `zone-pivot-groups.yml` file in [learn-pr](#).
2. Tag a Learn repo manager for approval and request to merge (@MicrosoftDocs/msft-learn-repo-managers).

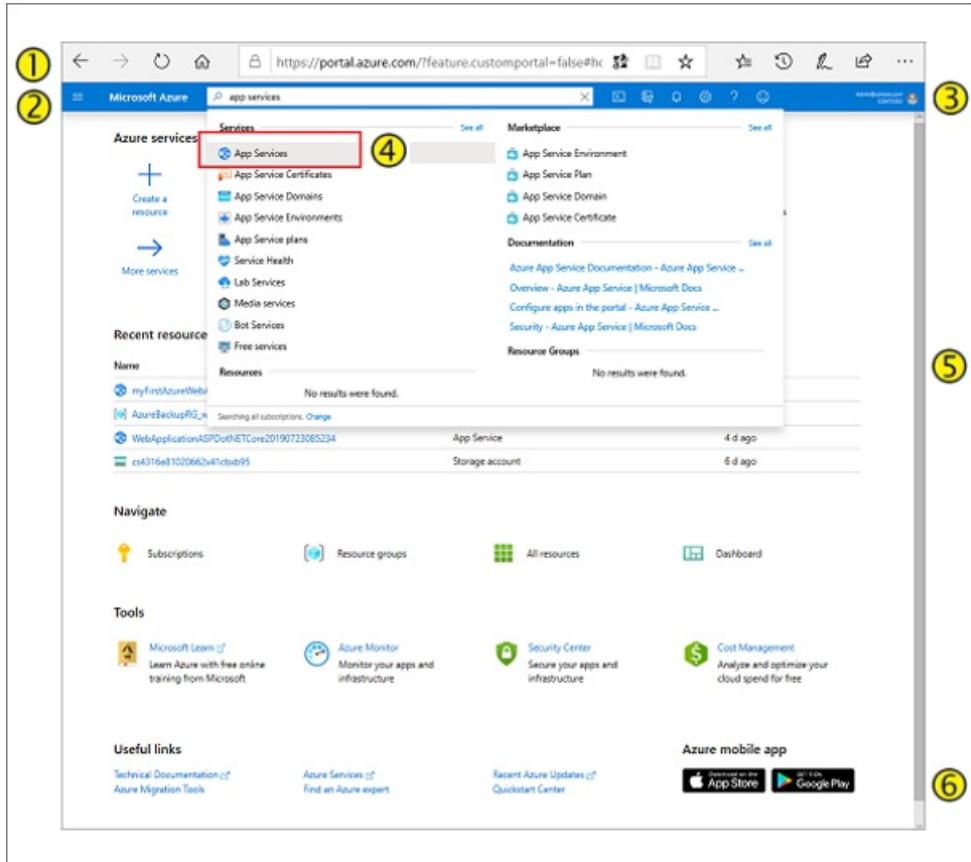
Create screenshots for documentation

1/14/2022 • 6 minutes to read

This article provides instructions for creating screenshots 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 images 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 image 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 [create a gray border](#).
- **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.
- **Windows 11:** As part of the design refresh, rounded corners have been brought back to UI elements and menus within Windows 11. Additionally, shadows under windows are turned on by default. For clean screen captures, it is recommended that you turn off this option:
 1. Press the **Windows Key** to enter search.
 2. Search for "*view advanced system settings*", then select.
 3. Within the **System Properties** window, click the **Advanced** tab.
 4. In the **Performance** section, click the **Settings** button.
 5. Within the **Performance Options** window, click the **Visual Effects** tab.
 6. Click the **Custom** settings, then uncheck the "*Show shadows under windows*" option.
 7. Click the **Apply** button.



Requirements for naming 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](#).

Tools for capturing images

Here are some common tools for taking screenshots:

- [Snipping Tool](#): Included with 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 doesn't 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 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.
- [Figma](#): Web-based, vector-based UI and UX design tool for creating websites, apps, or smaller user interface components in SVG format. Free to use. Highly recommended for the conversion of high-res screen captures from PNG (raster-based) to SVG (vector-based) for scalable, lossless integration into support documentation.

Add an image in Markdown

There are two approaches to adding 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":::
```

Conceptual art

1/14/2022 • 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:

1. Create a work item in the Azure DevOps board corresponding to your project to track the request:
 - [Docs-related projects template](#)
 - Learn-related projects: From your ModuleWorkItem, select the green "New Conceptual Image" button in the OneClick actions to create a new image work item that is associated with the corresponding Learn module. Assign that item to Tyler Stubbs (v-tstubbs@microsoft.com). If needed, here's a [template](#).
2. Attach any type of draft (for example, architecture, image) in the work item.
3. Email the item number to the artist, Tyler Stubbs, at v-tstubbs@microsoft.com.

Do it yourself

Follow these standards and practices to create conceptual art on your own.

File types

For conceptual art, SVG is preferred, though PNG is acceptable. [Screenshots](#) are best as PNGs.

- **SVG:** SVG is the preferred file type for conceptual art because it scales without loss of clarity, regardless of the browser window. SVGs don't, 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.)
 - If you use Visio to create your SVG, the default image background will most likely be transparent. However, transparent backgrounds should be avoided when publishing to Docs. Transparent backgrounds with black text make a very difficult read with a dark Docs theme. To easily add a white background to an SVG, open the file in a text editor like VS Code. Before the end of the `<svg>` heading section (right before the `>` character), insert the following code to add white background:

```
style="background-color: white"
```

- Conceptual images don't need to have a border. If you want to use lightbox functionality to zoom into your image, insert `border="false"` before your lightbox image link.

Here's an example:

```
:::image type="content" source="./media/overview-azure-hybrid-benefit-scope/duty-separation.svg" alt-text="Diagram showing the separation of duties." border="false" lightbox="./media/overview-azure-hybrid-benefit-scope/duty-separation.svg":::
```

Important: Don't use interactive or scripted SVGs on Docs (for design, quality, and security reasons).

To repo admins: Make sure the SVG file type is specified in your docfx.json file. Otherwise, the image won't appear in the published file. For implementation details, see [Images](#) in the Docs Markdown Reference.

- **PNG:** PNG is an acceptable format. As a raster format (pixels), if the art is viewed at its original saved size (100%), then it's 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 shouldn't be used. If you get a build-validation warning for missing alt text on an icon, use the [icon image syntax](#).

For more information about writing good alt text, see [Accessibility Guidelines for Multimedia](#).

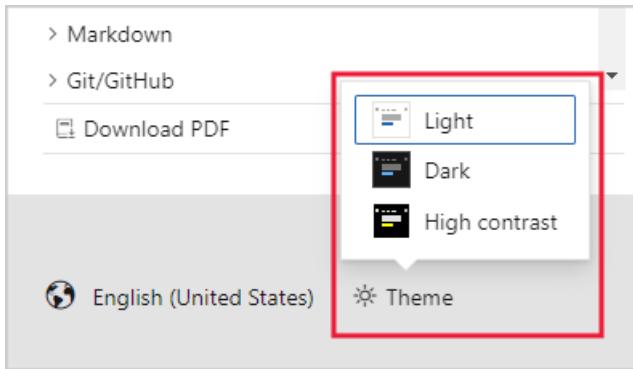
Localization

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 three of our docs.microsoft.com themes: **light**, **dark**, and **high contrast**.

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



- Correct use of transparency on icons:

APPLIES TO: SQL Server Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

- Incorrect background on icons:

APPLIES TO: SQL Server Azure SQL Database Azure SQL Data Warehouse Parallel Data Warehouse

What about the source files for conceptual art?

To help maintain the conceptual art as its associated document evolves, allow 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 isn't possible and the source file is 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](#)
- [Find the correct icon and logo galleries](#)

Create and publish animated GIFs in documentation

1/14/2022 • 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 reasonably be expected to work. 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:** Aim to keep your file under 5 MB. 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 GIFs, 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. See [Online services](#) for tools that can help you reduce the size of your GIF.
- **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:** The GIF 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 also helps animated GIFs show up in search results and helps 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

1. **ScreenToGif** is a free, open-source tool that makes screen recording on a Windows-based PC easy and saves to an animated GIF file.

For the download instructions and examples, see:

[ScreenToGif](#).

2. The **Game bar** is a built-in feature in Windows 10 that can take a screen recording and save it as an MP4 file. To convert an MP4 file into a GIF, you can use the open-source converter [FFmpeg](#). For more information, read the rest of this section.

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 convert the file, 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 Snagit's [online documentation](#). Contact your manager and admin to get a Snagit license key.
- **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. Use the *animate* command 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 that lets you set the loop count (see [these instructions](#)).

Adding code to enable start/stop capability for GIFs

Currently, [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 website 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 and 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 an expandable screenshot (lightbox)

1/14/2022 • 3 minutes to read

In this article you will:

- Learn what a lightbox is and how it can be used.
- Receive guidance for how to appropriately leverage lightboxes.
- Learn how to create a lightboxed image.

Screenshots are commonly used in documentation. However, a screenshot of an entire webpage tends to be large. Putting the entire image inline in documentation can disrupt the flow of the page and impair readability.

Lightboxes to the rescue! A lightbox is a special way to display an image on your page. When users click a lightboxed image, an expanded image overlaid over the rest of the page content will appear. This allows you to use scaled-down or cropped versions of images inline, while still allowing users to access the larger version of the image if they desire. Users who don't need to see the entire image don't have their experience disrupted and users who do can view it without leaving the page.

There are two ways to use this feature:

1. Use two different images for the inline and expanded images. When users click on the inline image, a lightbox with a different image will appear. This is appropriate for images that can be cropped to highlight the most important part of a webpage or application inline and then displayed in full in the lightbox.
2. Use the same image for the inline and expanded images. When users click on the inline image, a lightbox with a larger, zoomable version of the same image will appear. This is appropriate for images that will lose important context if cropped and need to be shown in full inline.

Examples

To see this feature in action, we'll be looking at a page from Xamarin's documentation: [Setup and Installation](#). Click any of the following images to see the lightbox in action.

NOTE

For the sake of clarity, the image filenames and paths as used on this page have been changed from the filenames and paths used in the original Xamarin article.

Using two different images

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

Choose a template for your new project

The screenshot shows a mobile application template selection screen. On the left, there's a sidebar with categories: Multiplatform (App, Library, Tests), iOS (App selected, Library, Tests), and Android (App, Library, Tests). The main area is titled "General" and contains five items: Single View App (selected, indicated by a blue background), Master-Detail App, Tabbed App, Page-Based App, and WebView App. Each item has a small icon to its left.

Using a lightbox here allows for significantly improved page flow. Users who want to better orient themselves by viewing the full interface can do so with a single click, and users who don't need that context don't have their experience disrupted with an overly large and verbose image.

Using the same image

Here's an example of using a larger version of the inline image as the expanded image:

This screenshot shows the same template selection interface as above, but with a larger, detailed image of a WatchKit App. The "WatchKit App" item in the General section is highlighted with a blue background. To the right of the list, a large, semi-transparent image of a smartwatch with a circular face and a rectangular display screen is overlaid. Below this image, the text "WatchKit App" is displayed, followed by a description: "Creates a WatchKit App and a WatchKit Extension for watchOS v2.0 or later." At the bottom of the screen, there are "Cancel", "Previous", and "Next" buttons.

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="media/article-folder-name/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="media/article-folder-name/image-file-inline.png" alt-text="Image alt text." lightbox="media/article-folder-name/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.

Add code samples to Microsoft Learn content

1/14/2022 • 6 minutes to read

This document outlines the guidance for creating and maintaining code samples for Microsoft Learn. The content below is applicable to all products and teams, regardless of the organization.

General guidelines

These principles set the baseline for quality standards of code that we ship for Learn. This list is less of a checklist, and more of a guiding set of recommendations that will help you build content that resonates with customers.

- Use the least amount of code possible to achieve your goal.
- Assume students will copy code into production apps.
 - Always follow industry good/best practices even if it means additional code.
 - Use proper [C# coding conventions](#) and refer to language style guides such as the [Python Style Guide](#).
 - Prefer to include appropriate error checking unless it significantly distracts from the core concepts.
- Always use appropriate method and variable names. As an example, for C#, the [Microsoft C# Naming Conventions](#) should be a good point of reference for what those should be.
- Avoid code comments (*if possible*) - the explanation should be in the unit itself.
- Use temporary variables if it helps convey meaning.
- Split a method to clarify usage intent - don't write one giant function to get things done.
- Always use real-world examples, especially when naming classes, methods, and variables.
- Use an ellipsis (...) in place of boiler-plate code or code not relevant to the current task/topic.

Organizing code samples for a module

Starter code, or code used in a module should always be placed into a GitHub repository in the [MicrosoftDocs](#) organization. Samples should be organized around the **solution** and use the naming convention `mslearn-{shortened-solution-name}`. For example, if the solution is "store-app-data-with-azure-blob-storage", a repository to hold the code might be `mslearn-store-data-in-azure`.

Creating new code repos

You can create a new MicrosoftDocs GitHub repository using the [Create a new repository](#) feature of the [internal open source tooling](#).

1. Select "GitHub for Open Source".
2. Give the repo a name using the naming standards (`mslearn-{solution-name}`).
3. Indicate it's a sample for a Microsoft Learn module in the description.
4. While it will need to be a **Public** repo before the module goes live, but it can start as **Private**.
5. For **Assign a Service or Opt-Out**, select the **Search services** field to open up the options.
6. At the top, select **Opt-out:No appropriate service**. This will add new fields for maintainers below this section.
7. For classifying the repository, select **Production**.
8. In the **Maintainers** section, make sure your name is added if you need to be able to manage this repo.
9. For the **Maintainers > Security group** field, set it to `learn-code-repos` and select the **MSFT Learn - Code Repo Maintainers** group option.

10. In the **Open source release approval** section, you'll make several selections.
 - a. New project
 - b. Yes, public open source
 - c. Sample code
 - d. MIT license
 - e. Yes, all created by my team
 - f. No to the project sending any telemetry
 - g. No to any cryptography (unless your sample actually does implement cryptography)
 - h. < 5,000 lines of code

11. For **Administrator Team permissions**, check the box next to **MSFT Learn Repo Managers**.

12. Skip the **Write permissions** section.

13. Confirm the **Repository template** is set to an MIT license with copyright set to Microsoft.

14. Pick an appropriate **.gitignore template** for the most common code that will be in your code repository (e.g., **VisualStudio** for a .NET project).

15. When you have confirmed everything, select **Create repository**.

Once the repo is created, each maintainer who needs **write** access to the repo should use just-in-time access to give their GitHub account **administrator** access. To do this, navigate to the **Direct owner access** tab and follow the instructions on the screen to elevate your access.

TIP

You should make it public initially unless the content is embargoed. Private repos can't be cloned without authentication which makes testing modules in the review site harder. You can always change the repo status after creation if you decide to change your mind later.

Guidelines for publishing code

| CHECKLIST ITEM | DESCRIPTION |
|------------------------|---|
| In public GitHub repo | <p>✓ DO
 Host your code in GitHub.
 Make the repo public.
 Use the MicrosoftDocs organization.
 Use the internal open source tooling to create a new repo.
 Name it with the prefix <code>mslearn-</code></p> <p>⊖ DO NOT
 Use a personal account/repository for the sample.
 Use VSTS/BitBucket/etc. for the sample.
 Make the repo private.</p> |
| Using simple branching | <p>✓ DO
 Use simple branching - <code>main</code> for content that is visible/public, any other branches for tests/experiments/validation.
 Place all sample content for a learning path in one branch.</p> <p>⊖ DO NOT
 Use elaborate branching for publication (e.g. one unit in one branch, another unit in a different branch).</p> |

| CHECKLIST ITEM | DESCRIPTION |
|---|---|
| One repo per solution | <p>✓ DO
 Create one repo per each solution. This could mean by module, multiple modules, or learning path.
 If the repo is for multiple modules, organize the code in your repo by module, where the folder name matches the module UID.
 Name your repo in the format of <code>mslearn-{module-id}</code>.
 Use your best judgment to determine a short name for the repo, with the <code>mslearn-</code> prefix.
 Use one repo for all languages in a module.</p> <p>⊖ DO NOT
 Create multiple repos for different languages within the same learning path.
 Use folder names that are inconsistent with the names of modules within the learning path.</p> |
| Include a <code>README.md</code> | <p>✓ DO
 Create a <code>README.md</code> file in the root of the repo.
 Outline the purpose of the samples.
 Include a link to the published module where the sample is used.
 Include a README for each full sample that is in the repo.</p> <p>⊖ DO NOT
 Leave the sample with no <code>README.md</code>
 Host documentation from within the sample repo.
 Have a very non-descriptive <code>README.md</code> file (e.g. "<i>This is a Learn Sample.</i>").</p> |
| Place all code in a <code>/src/</code> folder | <p>✓ DO
 Place all source in one parent folder.</p> <p>⊖ DO NOT
 Have the code scattered in the repository.</p> |
| Include an open-source license | <p>✓ DO
 Include a <code>LICENSE</code> file.
 Use MIT or Creative Commons licenses.</p> <p>⊖ DO NOT
 Skip a sample without a license.
 Use GPL or GPL-flavored licenses.</p> |
| Work with your community | <p>✓ DO
 Regularly check for PRs and issues.
 Respond to issues and PRs promptly.
 Deprecate the repository if sample is not used or no longer relevant.</p> <p>⊖ DO NOT
 Abandon the repository with no communication.
 Not respond to PRs and issues.</p> |

| CHECKLIST ITEM | DESCRIPTION |
|--|---|
| Prepare for indexing to the samples portal | <p><input checked="" type="checkbox"/> DO
 Create a manifest for the final, complete sample.
 Add a webhook to get your sample repo.</p> <p><input type="checkbox"/> DO NOT
 Onboard every unit sample to the samples portal.
 Ship a final, complete sample without a manifest file.</p> |
| Ensure Code is Validated | <p><input type="checkbox"/> DO NOT
 More details on this requirement will be available once the Constructors infrastructure is launched in the next months.</p> |

Picking the organization

All samples need to be placed in the [MicrosoftDocs](#) organization.

Including code in Docs

Refer to our official guidance on [how to include code in content](#). We prefer that you include code from code files directly, via the `[!INCLUDE]` syntax instead of using code fences/inline code. That ensures that we are always able to build the content from the latest and greatest code, and the author does not have to re-visit every single article that has the snippet to add changes. Additionally, this also allows us to test the code and ensure that it works before being shipped to customers.

When it comes to including code within your documentation set, you should rely on [Cross-Repo References](#).

Additional questions

Why can't I have samples in my own repo?

To make sure that learning paths and any associated content can be maintained in the long run, we need to ensure that we retain ownership of the code at all times, hence the requirement to place the code into one organization.

Other questions

When in doubt, please reach out to the [Learn General Channel](#), we will be able to point you in the right direction.

Add links to articles

1/14/2022 • 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 | 1 st | another-file.md | |
| Root relative path | 2 nd | ~/different-directory/another-file.md | |
| Site relative URL | 3 rd | /windows/uwp/get-started/get-set-up | |
| FQDN / Full URL | Avoid | https://doc.microsoft.com/help/contribute/links-how-to | |
| Cross repo reference | 1 st (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 |
|--------------------|---------|----------|---------|
| File relative path | Y | Partial | Y |
| Root relative path | Y | Partial | Y |
| Site relative URL | N | N | Y |
| FQDN / Full URL | N | N | N |
| Cross reference | Y | -- | Y |
| Reference | Partial | Partial | Y |
| HTML URL | N | N | Partial |

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 and in GitHub Preview. To verify that the link works, replace the branch name in the link's generated URL with **main** or **master** (depending on the default branch name). 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:

Bookmark links

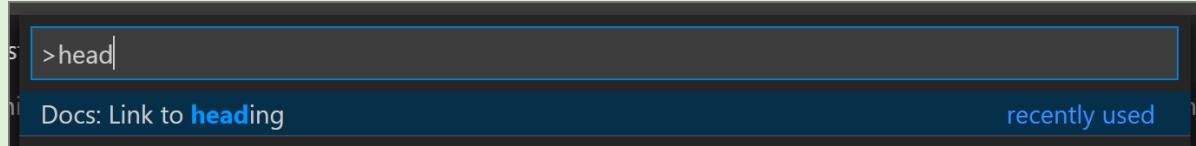
For a bookmark link to a heading in the *current* file, use a hash symbol followed by the words of the heading, with punctuation removed and spaces replaced with dashes:

```
markdown   
[Managed Disks](#managed-disks)
```

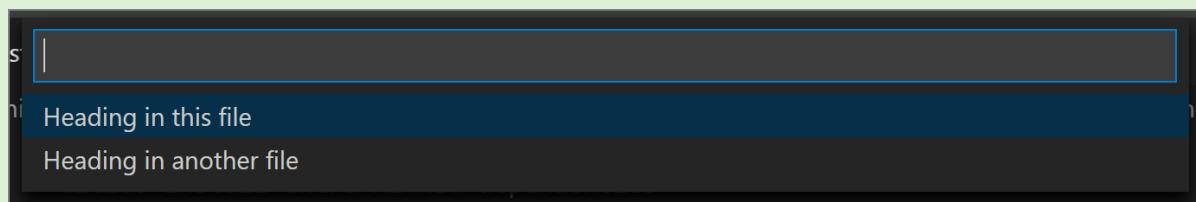
Select the link icon and then copy the bookmark anchor text from the URL (that is, the part after the hash).

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.

For more information about linking to versioned content, see [Write versioned content](#).

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.

For .NET or UWP APIs, 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.

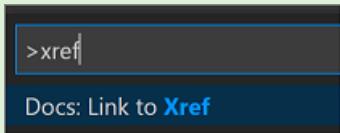
```
[link text](xref:UID)
```

Example: `[String class](xref:System.String)`

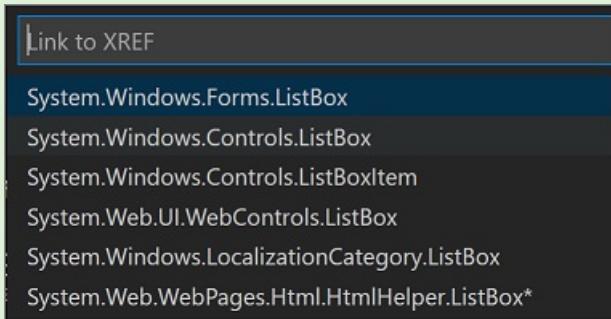
TIP

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.

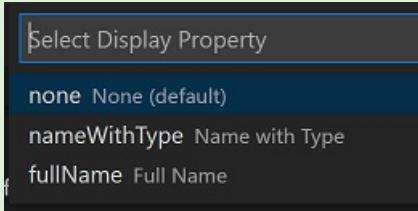


2. Type in part or all of the API's name, for example, `listbox`, and then press `Enter`. A list of matching APIs is shown.



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 |
|-----------------------|--|-----------------------------------|
| <code>fullName</code> | Displays the fully qualified API name. | <code>System.String.Trim()</code> |

| DISPLAY PROPERTY | DESCRIPTION | EXAMPLE |
|------------------|---|----------------------------|
| nameWithType | Displays <code>namespace.type</code> for types and <code>type.member</code> for type members. | <code>String.Trim()</code> |

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`:

```

87 <meta name="page_type" content="dotnet" />
88 <meta name="page_kind" content="class" />
89 <meta name="description" content="Represents text as a sequence of UTF-16 code units. " />
90 <meta name="toc_rel" content="_splitted/System/toc.json" />
91 <meta name="source_url" content="" />
92 <meta name="ms.assetid" content="System.String" />
93 <meta name="pdf_url_template" content="https://docs.microsoft.com/pdfstore/en-us/VS.dotnet-api-docs/{branchName}{pdfName}" />
94 <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": "4.5.1", "netframework": "4.5.2", "netframework45": "...", "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": "4.5.1", "netframework": "4.5.2", "netframework45": "...", "xamarinmac": "4.6", "netframework46": "...", "xamarinmac30": "...", "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 |
| * | %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-azurermp-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`. Using this parameter doesn't affect your ability to later update the FWLink.

Add alerts to Learn content

1/14/2022 • 2 minutes to read

We recommend minimizing the use of alert texts in your content. The alert types include Note, Tip, Caution, and Warnings. Avoid adding off-topic information as alert text. Always try to work out your alert text within the main content. However, if it's fully relevant and helpful to the task at hand, use the alerts.

NOTE

The information in this article is also available in the [Markdown reference Contributor Guide](#) article.

Implementation

Use the following syntax to add appropriate alert text.

```
> [<!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 render as the following examples on the Learn module.

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.

Add lists and bullets

1/14/2022 • 2 minutes to read

Bullets and lists can be used to break up walls of text and organize your content into logical, easy to read chunks. The bulleted list helps:

- Attract attention to important points.
- Scan textual content quickly.
- Present information in a clear and concise manner.
- Efficient communication with the learners.

Use bullets to break down conceptual content and numbered lists for exercise steps. When describing sequential steps, such as in an exercise, use numbered lists.

NOTE

The information in this article is also available in the [Markdown reference Contributor guide article](#).

Implementation

Bulleted list

Some important points to note when adding bulleted lists in markdown:

- Bullets and numbered lists should be surrounded by blank spaces, which means you should leave a blank row before and after a bulleted list. Not surrounding the bulleted list with a blank row generally doesn't break your content, but it's a good practice to keep your source content clean. Following this rule makes the wriggly green line before/after bulleted lists disappear.
- Use `-` or `*` characters for bullet style. There's only one bullet style (solid disc) in Markdown.
- For sub-bullets, manually enter two spaces to indent your sub-bullet under the main bullet. Use the `-` or `*` characters for bullet style, the rendered sub-bullet style in Markdown is a white disc.

```
- bullet one
- bullet two
  - sub-bullet one
  - sub-bullet two
- bullet three
```

The above bulleted list is rendered as follows:

- bullet one
- bullet two
 - sub-bullet one
 - sub-bullet two
- bullet three

Numbered lists

You can use incremental numbers for your bulleted list. However, you may need to remove an item from the middle of the numbered list. If you do, you'll have to manually change the number for all the following items. The best way to avoid the rework is to number all the items in your numbered list as `1`. Once your content is

built, the list is automatically rendered sequentially.

Use `1` for the nested list as well. These "numbers" are rendered as lowercase letters when published.

```
1. list item one
1. list item two
  1. sub-list item one
  1. sub-list item two
1. list item three
```

The above bulleted list is rendered as follows:

1. list item one
2. list item two
 - a. sub-list item one
 - b. sub-list item two
3. list item three

Checked lists

You can add the class `checklist` to a bulleted list to turn it into a checked list (with checkmark glyphs). The checked list is added by placing the bulleted list into a quote block and adding a `div` header as shown here:

```
> [&div class="checklist"]
> - Item 1
> - Item 2
```

This code will render something like:

✓ Item 1 ✓ Item 2

Dropdown lists (Selectors)

You can create a list of links in a dropdown (combobox) with the `op_single_selector` class:

```
> [&div class="op_single_selector" title1="Select the version of the service you want to use"]
> * [Version 1](data-factory-azure-sql-connector.md)
> * [Version 2 (current version)](../connector-azure-sql-database.md)
```

This code example renders a dropdown list of links to navigate to the respective pages and can be used when multiple options can be chosen.

For more information on selector lists, see the [Markdown reference documentation about selectors](#).

Include reusable content in articles

1/14/2022 • 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) ]
```

Create a learning path

1/14/2022 • 5 minutes to read

In this article you will:

- Learn about learning path guidelines and how to appropriately leverage the learning path feature
- Learn how to create the main learning path index.yml file
- Learn about learning path metadata

Overview of Microsoft Learn learning paths

Learning paths are small collections of modules that are presented to a learner in a recommended order. Learning Paths can be organized around roles (Developer, Architect, System Admin, etc.), technology (Web Apps, UWP apps, etc.), or even feature sets such as "Customizing Xamarin.Forms".

Notably, learning paths:

- Are small collections of modules (3-8) that are presented in a recommended order, but users are not required to complete them in that order.
- Are authored in YAML
- Grant trophy achievements upon completion

Example:

Refer to a [published learning path](#) for an example. The source files for this learning path are in the [learn-pr repo](#).

Implementation

File Structure

Learning paths are a single YAML file, contained in a folder, that displays the list of modules and metadata to the learner.

Create the learning path folder

Each learning path has its own folder in the "paths" folder of the repo. The folder name is a short version of the learning path name that is acceptable to be seen by learners (it will be a part of the learning path URL).

Example:

/learn-pr/paths/sample-path-short-name

Create the learning path index file

The learning path index file is a singular YAML file stored in the learning path folder that defines the learning path metadata and the display of metadata. The file should be named `index.yml`.

Sample learning path code

```

### YamlMime:LearningPath
uid: learn.architect-great-solutions-in-azure
title: Architect great solutions in Azure
summary: Learn how to design and build secure, scalable, performant solutions in Azure by examining the core principles found in every good architecture.
metadata:
  title: Architect great solutions in Azure learning path
  description: Learn how to design and build secure, scalable, performant solutions in Azure by examining the core principles found in every good architecture.
  ms.date: 09/24/2018
  author: markjulmar
  ms.author: smmark
  #ms.learn-contact: smmark
  ms.topic: interactive-tutorial
  ms.prod: learning-azure
prerequisites: None
iconUrl: /learn/achievements/architect-great-solutions-in-azure.svg
levels:
- beginner
- intermediate
roles:
- solution-architect
products:
- azure
- azure-storage
- azure-sql-database
- azure-cosmos-db
- azure-cost-management
- azure-advisor
- azure-monitor
- azure-application-insights
- azure-log-analytics
- azure-cdn
- azure-active-directory
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
trophy:
  uid: learn.architect-great-solutions-in-azure.trophy

```

Explanation of Metadata

All metadata is required unless explicitly marked as optional.

| METADATA | INFO | DESCRIPTION |
|---------------------------|------|---|
| ### YamlMime:LearningPath | | Specifies to the Docs platform the type of YAML file and metadata to expect. Must be the first line of the YAML file. |
| uid | | A human generated unique ID that follows the format: (repo-name). (learning-path-short-name) |
| title | | Learning path title. Displayed in the header and breadcrumbs. |

| METADATA | INFO | DESCRIPTION |
|-------------------------|---|--|
| summary | | A brief summary describing what the learning path is about. Appears on the main learning path page. Can be Markdown. |
| metadata | page-level metadata. | These properties are helpful in search engine optimization. Review the SEO Basics article in the Contributors Guide for more information. |
| metadata: title | Module title | This is the most important metadata for SEO. |
| metadata: description | A summary of the learning path. | Used in site search. Sometimes used on a search engine results page for improved SEO. |
| metadata: ms.date | A date in the format MM/DD/YYYY. | Displayed on the published page to indicate the last time the module was substantially edited or guaranteed fresh. If this field is missing, the date of the last commit is displayed instead, which may be incorrect for freshness. |
| metadata: author | The author's GitHub alias. | Identifies the author by GitHub ID in case there are questions about or problems with the content. In some cases, the author might be notified via GitHub automation of activity involving the file. |
| metadata: ms.author | The author's Microsoft alias, without "@microsoft.com". | Used for content reporting and BI. |
| metadata: ms.topic | Add "interactive-tutorial". | The type of content. |
| metadata: ms.prod | The type of product. Should be one of these values: <code>learning-azure</code> ,
<code>learning-powerapps</code> ,
<code>learning-powerbi</code> , <code>learning-flow</code> ,
<code>learning-d365</code> ,
<code>learning-<product></code> | Used for issue triage and reporting. |
| prerequisites | Bulleted list of module prerequisites or the word <code>None</code> | This appears on the module tile so that learners can make an informed decision about taking the specific module. |
| iconUrl | | The icon for the learning path. This should be the same as the achievement image (see below.) |
| levels, roles, products | | Arrays of strings. Must be one or more of these values . As these are arrays, make sure to append the values with a <code>-</code> . |

| METADATA | INFO | DESCRIPTION |
|--------------|--|--|
| modules | | Each item in this list represents a module in your learning path. |
| modules: uid | | Array of values matching the UID of each module. As this is an array, make sure to append the UID with a -.. |
| trophy: uid | A globally unique UID for the achievement trophy in the format (module uid).trophy | This trophy appears on successful completion of the module. |

Create modules

To create a module, view the [Create a module](#) article.

Add an achievement to the learning path

One of the unique features of Microsoft Learn learning path is that learners can earn achievement trophies upon completion of the learning paths. These achievements show up on their Docs user profile and can be shared via social media and other channels.

In order to get the learning path to successfully build, you will have to add an achievement to the content. For most projects, this means you will have to use a placeholder image until the finalized trophy artwork is ready. You cannot launch with placeholder images, so in order to not block your launch timeline - you will need to request to have achievement artwork created at the beginning of the content development project so that it can be ready by the time you are done developing and ready to launch. View the [Create an achievement](#) article for details about requesting achievement artwork.

1. In the learning path `index.yml` file, update iconURL to the following:

If you do NOT have the final achievement image:

```
iconUrl: http://via.placeholder.com/120x120
```

If you need to publish the learning path using a temporary achievement image:

```
iconUrl: /learn/achievements/generic-trophy.svg
```

Be sure to update the image with the final image once it's ready.

If you DO have the achievement image:

```
iconUrl: /learn/achievements/<learning-path-folder-name>.svg
```

2. Update the trophy uid to be globally unique (across all Learn repos). The easiest way to make it unique, and easy to manage is to use the learning path UID, and then append ".trophy" to the end.

```
trophy:
  uid: <learning-path-uid>.trophy
```

TIP

Learning path achievements are always **trophies**. If you try to create a badge for a learning path, it will throw a build error.

IMPORTANT

The achievements.yml file still exists, but is only for deprecated modules and learning paths. If a module or a learning path is not deprecated, there should not be any associated achievement information in the achievement.yml file.

IMPORTANT

Do NOT change the uid for an achievement after it is launched. This will deprecate the old achievement and cause a new achievement to be created in its place. This is a poor learner experience, as the users who earned the old one will not have credit for the new one.

Engineering Documentation

- <https://review.docs.microsoft.com/new-hope/specs/triple-crown/learning-paths>
- <https://review.docs.microsoft.com/new-hope/specs/learn/achievement-authoring>

Related Documentation

[Unified content model](#)

Topic Landing Pages

1/14/2022 • 7 minutes to read

Topic Landing Page Overview

Topic landing pages are a template for content on Learn that allows us to have a hero, links to various resources, some curated content, and some automated content. We wanted to create a consistent look and feel to the site, while giving people guideposts to be able to start from.

Topic landing pages allow us to have a place for specific audiences. The pages are built around the taxonomy on the site, to give people a constant source of content, but also give ourselves and our partner teams a place to be able to feature specific content and also link out to partner sites. Details are below about the 2 cases for which we use topic landing pages, taxonomy-driven and special programs.

Content Repo

The topic pages live in LearnShared repo, which is managed by Daniel Stafford, Sam Adadow, Pam Spier, and Derek Peterson. We have chosen to limit access to the template and repo in order to limit the number of teams creating these pages outside Learn guidelines.

Creating or Updating a Topic Landing Page

Details on how to request a new or update an existing topic landing page are [detailed below](#).

Guidelines

Topics

Currently we only allow landing pages to be built for concepts and terms which currently exist in the [Docs taxonomies](#) used on Learn (role, L1 product terms, level, and type) and special programs approved by Learn PM and Content teams. Topics must have 10 learning paths in the taxonomy values used to populate the popular cards to be considered.

For taxonomy-driven topics

- We are currently unable to manage and track topics which appear outside of how we currently tag Learn content. We understand this is limiting, but the scope of potential page topics will expand as our taxonomies and metadata are able to support them.
- Within the template, the final set of cards is an auto-generated set of recommended learning paths and modules based on selected tag which appear in our content. Creating topics which are not described by these tags could create situations where the auto-generated recommendations are unable to meet user expectations based on the rest of content on the page. This leads to a poor user experience, though we will be able to handle complex topics more easily as our metadata capabilities mature.

For special program topics

- Special programs encompass those topics which are specific to a defined initiative or program whose content has been integrated into Learn. These topics may take advantage of existing Docs taxonomies, but are more manually curated and defined by individual content teams.
- Topics which fall under “special programs” are required to discuss the need and purpose of their landing page with Learn PM to get approval for developing the landing page as it needs to fit within the larger Learn content strategy.
- Special programs are currently not exposed in the Learn navigation or on the Learn homepage, so teams

proposing a special program topic page will need to describe how they will drive awareness and traffic to this page.

- Likewise, special programs need to have content to support the purpose of the page. If the content does not currently exist on Learn, the requesting team will need to describe how they plan to create the units, modules, and/or learning paths needed.

Content Strategy

When proposing a new topic page, a cohesive content strategy should be in place to support the creation of the new page. This means that you should be able to answer three essential questions about the page you're looking to create:

1. **Who is this page for?** Because of the template design, a single audience or two closely related audiences is easier to curate for.
2. **What are their needs, or what are they hoping to do?** The template currently only allows content that exists on Learn, outside the three feature cards at the top of the page. The Learn content on the page should be able to fully meet the audience's needs. For instance, many of our certifications and exams benefit from and reference content from across the Docs platform, but we are likely unable to curate enough content solely on Learn for a topic page for the audience and user need.
3. **How will you drive awareness and traffic to this page?** The limitations and scope of the header navigation and homepage means we can't link to every topic landing page from the navigation itself or from the primary Learn homepage content. It is up to the content team proposing the topic page to define their approach for driving traffic to their landing page.

Your answers to the questions above will not only benefit the focus of the landing page but should also help you create curate the right pieces of content for the page.

Template Overview

The [student role page](#) is an example of the topic landing page template:

The screenshot shows the Microsoft Learn for Students topic landing page. At the top, there's a navigation bar with links for Microsoft, Docs, Documentation, Learn, Q&A, and Code Samples. On the right side of the nav bar are Search and Sign in buttons. Below the nav bar is a secondary navigation bar with Learn, Products, Roles, Learn TV, Certifications, and FAQ & Help. Underneath that is a breadcrumb trail showing Docs / Learn. To the right of the breadcrumb are Bookmark and Share buttons. The main content area has a dark blue header with the text "MICROSOFT LEARN FOR STUDENTS" and "Build tech skills for the future". Below the header, a paragraph reads: "Master new concepts at your speed and on your schedule. Start here to develop practical skills through fun, interactive modules and paths. Plus, educators can get access to Microsoft classroom materials and curriculum." A blue button labeled "Browse all paths for students" is visible. The main content area features several cards: one for "COLLECTIONS" titled "Learn with university courses" with a "Browse University collections" link; one for "PROGRAMS" titled "Become a Microsoft Learn Student Ambassador" with a "Browse program details" link; one for "PROGRAMS" titled "Are you an educator?" with a "Explore details" link; and three "MODULE" cards at the bottom: "Track wild polar bears with AI", "Classify endangered bird species", and "Track global air quality with Azure".

Learning Paths:

- Track wild polar bears with AI**: 1 hr 20 min, ★★★★★ 4.7 (200). Beginner, Developer, Azure.
- Classify endangered bird species with Custom Vision**: 32 min, ★★★★★ 3.4 (10). Beginner, Developer, Azure.
- Track global air quality with Azure Maps**: 36 min, ★★★★★ 4.7 (9). Beginner, Developer, Maps.

Modules:

- MODULE Create a chat bot to help students learn with Azure Bot Service**: 1 hr 16 min, ★★★★★ 4.9 (21). Beginner, Developer, Azure.
- MODULE Help remote farmers protect their crops with text message weather alerts using Azure Functions**: 35 min, ★★★★★ 4.5 (2). Beginner, Developer, Maps.

Popular learning paths and modules for students

MODULE Create an Azure account: 39 min, ★★★★★ 4.7 (113,025). Azure, Administrator, Beginner.

MODULE Cloud Concepts - Principles of cloud computing: 1 hr 2 min, ★★★★★ 4.8 (137,718). Azure, Administrator, Beginner.

MODULE Core Cloud Services - Azure architecture and service guarantees: 45 min, ★★★★★ 4.7 (58,307). Azure, Administrator, Beginner.

MODULE Core Cloud Services - Azure compute options: 38 min, ★★★★★ 4.7 (51,412). Azure, Developer, Beginner.

MODULE Core Cloud Services - Azure data storage options: 25 min, ★★★★★ 4.7 (45,186). Azure, Developer, Beginner.

MODULE Core Cloud Services - Manage services with the Azure portal: 1 hr 13 min, ★★★★★ 4.7 (43,211). Azure, Administrator, Beginner.

MODULE Core Cloud Services - Azure networking options: 28 min, ★★★★★ 4.8 (36,151). Azure, Developer, Beginner.

MODULE Security, responsibility, and trust in Azure: 1 hr 16 min, ★★★★★ 4.7 (37,499). Azure, Developer, Beginner.

MODULE Predict costs and optimize spending for Azure: 1 hr 14 min, ★★★★★ 4.7 (34,334). Azure, Developer, Beginner.

[See what others are learning >](#)

Template Elements

- Page title and description:** The title that appears on the browser tab and in search engine results. The description, also called the meta description, will only appear in search engine results.
- Hero (required):** This section appears at the top of the page to draw the user in and help them understand the content they are going to find.
 - Title (H1)
 - Subtitle
 - Summary

- Button/CTA leading to a related content set. This could include the Learn browse page, pre-filtered by metadata, or something like LearnTV. Whatever is linked to should be in-sync with the overall strategy and topic of the page.
- Header image, customizable
- **Featured cards (optional):** A maximum of three feature cards can be used in the template, with a recommendation of having either zero or three cards. These are the only places within the page that link to content outside of Learn, which should complement the focus of the topic page and provide more information for users interested in the topic.
 - Title, which is the type of resource being linked to
 - For example, we use "Collections" if the link is to a collection, "Certifications" if it is linking to a cert, etc
 - Subtitle
 - Summary
 - Link/CTA
- Proposed card image which will need to be approved by the design team and fit within the larger Learn brand.
- **Curated cards (optional):** A maximum of six cards can be manually curated to include specific learning paths and modules relevant to the topic page. These cards require review every quarter to ensure that the content remains up-to-date and relevant. The title of this section is also configurable.
- **"Popular" recommended cards (required):** A set of 9 cards which are auto-curated based on the page metadata (product, role, level, type). Metadata can be combined similarly to the algorithm on the Learn browse page, which uses AND logic (ex: a page based on metadata terms "AI engineer" AND "intermediate" AND "Machine Learning Service"). The title of this section is configurable.

Learn Topic Landing Pages Governance Workflows

Roles

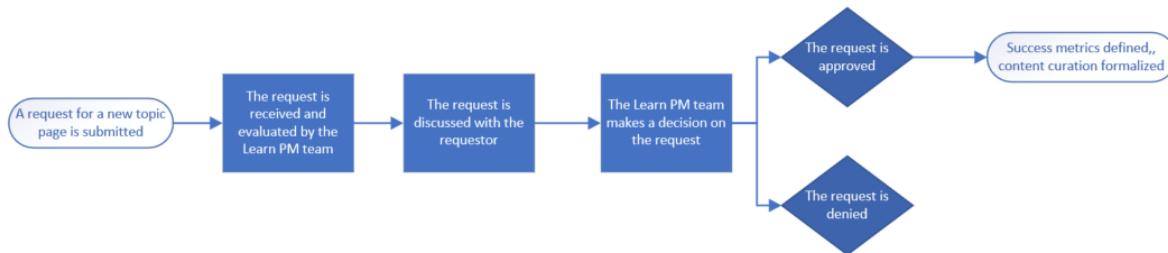
Requestor: A team or individual making the request for a new topic page to be created.

Daniel Stafford, Sr. Learn PM: Responsible for managing the LearnShared repo where topic pages are published. He is the primary approver for new page and update requests.

Learn content team: Focused on driving consistency and completeness for the content on the pages.

DevRel IA Team: Able to consult on and implement changes to the taxonomy/metadata issues as well as suitability of requests when needed.

Adding a new topic page



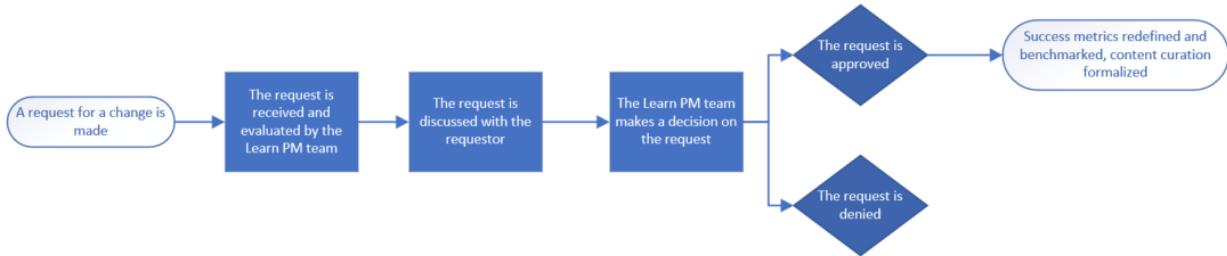
Requests

Please file a request using [this form](#). This will auto-generate a PM task in Azure DevOps, and send an email with the summary of the request and link to the PM task to a distribution that includes all PMs responsible for the pages and designated people from the C&L Partnerships, C&L Learn Content, and Azure Marketing teams.

The following is the information you'll need to fill out the form fully:

- Page requestor, including organization
- Intent of Page
 - Page purpose
 - Target audience
 - Timelines (ship date and duration)
 - Success metrics (projected number of users, completion rates, etc.)
- Marketing details
 - Promotion strategy
- All details needed to complete the template (detailed above)

Changing an existing topic page



Requests

Please file a request using [this form](#). This will send an email to a distribution that includes all PMs responsible for the pages and designated people from the C&L Partnerships, C&L Learn Content, and Azure Marketing teams.

The following is the information you'll need to fill out the form fully:

- Link to existing page
- Description of what is insufficient about the existing page
- Requested changes

Request Learn achievement images

1/14/2022 • 3 minutes to read

Overview

Microsoft Learn modules allow learners to earn achievement badges and trophies upon completion of the modules and learning paths. These achievements show up on their Docs user profile and can be shared via social media and other channels.

This achievement artwork is created by the Microsoft Learn Design team, and as a part of the content development project. The content developer will work with Learn designers to make sure that badge artwork is on-brand for both Learn and their product.

Badges vs. trophies vs. points

- **Badges** = Earned by completing **modules**. Must have badge artwork created by the Learn Design team and implemented in content by a content developer.
- **Trophies** = Earned by completing **learning paths**. Must have trophy artwork created by the Learn Design team and implemented in content by a content developer.
- **Points** = Earned by completing **units** and **bonus points for completing knowledge checks** on first try. These points are automatically distributed by the system and are not configured by content developers.

Learner experience of achievements

Learners don't have to sign in to the Docs platform to access Microsoft Learn content, but to earn an achievement, they do usually because the content has a knowledge check that is required to be marked complete. The user must be authenticated to validate the knowledge check. The achievement badge is also the same image that is used at the front of the module.

- User is consuming Microsoft Learn content (authenticated or non-authenticated)
- User gets to knowledge check (authenticated or non-authenticated)
- User answers multiple-choice questions (authenticated or non-authenticated)
- User submits answers for validation (authenticated or non-authenticated)
 - If not authenticated, will be asked to sign in.
- User completes all of the content in the modules, is marked complete, and earns an achievement.

Module view - from /learn page



Module - 6 Units - 48min

Build a serverless web app in Azure

Learn how to build a serverless website by using Azure Storage, Cosmos DB, API Management Service, and Functions.

Module view - unit listing



DOCS

Windows

Microsoft Azure

Visual Studio

Office

More ▾

ⓘ Explore Microsoft technology with live tutorials, including hands-on activities. This experience is in preview so

[Docs](#) / [Learn](#)



Build a serverless web app in Azure



Learn how to build a serverless website by using Azure Storage, Cosmos DB, API Management Service, and Functions.

[Host a website in Azure Blob storage](#)



[Create an instance of Azure Cosmos DB](#)



[Create Azure functions](#)



[Create an instance of the Azure Computer Vision API](#)



[Connect a storage account to the Event Grid API](#)



[Test your application and clean up your environment](#)



User profile with achievements

The screenshot shows the Microsoft Learn user profile for 'Ashley Johnson' (ashjohnson@microsoft.com). The top navigation bar includes links for Docs, Windows, Microsoft Azure, Visual Studio, Office, and More. A search bar and a user profile icon for 'Ashley' are also present. A message at the top encourages users to explore Microsoft technology with live tutorials, including hands-on activities, and invites them to send feedback. Below the message, there's a section titled 'Achievements' featuring two achievement icons: one for a robot and another for a cloud. A link 'Earn More Achievements' is provided. To the right, there's a section titled 'Your data' with a note about deleting saved progress and achievements, and buttons for 'Download' and 'Delete'.

Notification of earned achievement

A screenshot of a notification pop-up window. The title 'Great work!' is displayed prominently. The message states: 'By completing the quiz challenge you have unlocked the Deploy apps to Azure with Visual Studio and CI/CD achievement.' Below the message is a hexagonal achievement icon depicting a blue cube being deployed from a green base. A green button labeled 'More tutorials' is visible. At the bottom, there are social media sharing icons for Twitter, Facebook, and LinkedIn.

Request achievement artwork

The Learn Engineering Design team will be creating the artwork for badges and trophies. Once the module design document is confirmed, Microsoft Learn team will contact the Docs Platform Engineering Design team. The team needs time to create artwork and a name for the achievement that will be earned upon completion of your content. They'll also give you the image file to include in the module (iconURL). The entire process takes roughly 10-18 business days.

The final output is a pair of image files: an SVG to display on the Learn sites (ex: profile, landing pages, and so on) and a PNG for sharing the achievement on social media.

Process:

1. Content developer or PM requests a new achievement image ([badge, trophy](#)):
 - a. [Assign to Tyler Stubbs](#).
 - b. [Update area](#) to be in the correct content backlog, for example `Microsoft Learn\<Product> content`.
 - c. [Update work item title](#) to include title of content asset that the achievement is for (module or

- learning path).
- d. **Update work item description** to include a summary of the learning path(s) and related modules to give design context for creating the image.
 - e. **Include 'Target Release Date'** to ensure artwork is done in time for release.
 - f. **Optional: Include a description** of what typically is used to represent this topic/service/feature that could be incorporated into the produced image.
2. Achievement PM kicks off a new image request project with the Learn Design team. (One business day)
 3. The Learn Design team creates a draft of the achievement artwork. (10-12 business days)
 4. The Achievement PM will share the artwork draft with the requester to review. (1-2 business days)
 5. The requester will provide an approval or feedback. If feedback is given, the Design team will incorporate and provide a new version for review. Repeat until approved. (2-3 business days)
 6. Once approved, the Achievement PM will submit the artwork for a Global Readiness and CELA review. (1-3 business days)
 7. Once approved, the Achievement PM will attach completed SVG and PNG files to the AzDO work item. Then, the content developer can upload to their repo and implement in content. (1-2 business days)

Add achievement artwork to repo and content

Once the team creates the achievement artwork, it will need to be added to the repository that the content lives in and referenced in the content.

1. Rename the achievement artwork files to `[moduleorlp-folder-name].svg` and `[moduleorlp-folder-name]-social.png`.

NOTE

The file names for both svg and png files should be identical except for `-social` added at the end of the png file only.

1. Upload achievement artwork files to the `achievements` folder in the repo. This folder is located at the same level as the `docfx.json` file.

Sample folder structure

```
learn-pr\  
  achievements\  
    build-a-serverless-web-app.svg  
    build-a-serverless-web-app-social.png
```

2. Update the corresponding `iconURL` in the achievements and module/learning path YAML files. See the following articles for detailed instructions:
 - [Create a module: add an achievement to the module](#)
 - [Create a learning path: add an achievement to the learning path](#)

Hide modules and learning paths from search engines and Learn browse page

1/14/2022 • 2 minutes to read

Overview

There are a handful of cases where we want to publish content on Learn and not make it easily discoverable, such as:

- Event-driven (temporary)
- Beta
- Soon-to-be deprecated (in a window of overlap with a newer, standalone version)
- Only applicable for a select audience

For these scenarios, we would like to limit the ability to access the content. We'll do so by hiding it from the Learn browse page and making it NOINDEX to search engines. Then, it's only accessible via a direct link. The "hidden" feature solves the findability issue by allowing content authors to add a metadata tag to the `index.yml` file of modules and learning paths.

Learner experience

- Learners can visit hidden content by URL.
- Learners can't find the hidden content on Learn browse page.
- Learners can't find the hidden content on any search engines.
- We still track the progress and achievement for the hidden content.
- When learners access regular Learn URLs, for example `/search/learn?locale=en-us`, it will exclude all the hidden module/learning paths in the results.
- When learners access Learn URLs appended with `showhidden=true`, for example `/search/learn?locale=en-us&showhidden=true`, it will return all module/learning paths, including the hidden ones.

Implementation

1. Open the `index.yml` file of the module or learning path that you want to hide.
2. Add the property and value `hidden:true`.

Example in module `index.yml`

```
### YamlMime:Module
uid: learn-wwl.deploy-windows-update-in-microsoft-365
title: Deploy Windows Update in Microsoft 365
prerequisites: None.
iconUrl: /learn/achievements/deploy-windows-update-in-microsoft-365.svg
hidden:true
```

When this value is set, the build system will automatically add `ROBOTS:NOINDEX` in the system metadata and it won't appear in the content.

IMPORTANT

Since the content will only be accessible via direct link and not via browse, you will need to [manually build the production URL to find the content](#).

IMPORTANT

If the **content has already been published on the Learn browse page**, it will take 15-20 minutes for the content to be hidden after merging to live, and about 1-2 days for search engines to not find it anymore. For **new content**, it will take effect on the browse page and search engines at the same time when the build completes.

SEO basics for Learn

1/14/2022 • 2 minutes to read

To achieve maximum learner engagement, design your Learn content to be found by online search engines. There are many ways to make your content run optimally in web searches. You can also employ methods to drive traffic to Microsoft Learn.

Generally all basic Docs Search engine optimization (SEO) guidelines apply to Learn content. See the [SEO Guidelines](#) Contributor Guide article to understand the guidelines for search optimization.

For Learn content, the `title` and `description` defined in the page-level metadata in the YAML files primarily drive the SEO. So, it's important to ensure that your metadata is defined correctly to improve the visibility of your content in an online search. The page-level metadata section appears in the `index.yml` files for your learning paths and modules and in the YAML files for each unit. The Markdown content isn't used for SEO purpose.

Implementation

```
metadata:  
  title: Introduction  
  description: The Azure Portal provides a wealth of metrics that developers can use to monitor their Azure Blob Storage accounts. This module will show you how to write apps to access your Azure Blob Storage metrics programmatically.  
  ms.date: 04/12/2019  
  author: rmc Murray  
  ms.author: robmcm  
  ms.topic: interactive-tutorial  
  ms.prod: learning-azure
```

Explanation of metadata

Only the `title` and `description` metadata elements are used for SEO purposes. Each YAML file contains another `title` metadata element, which is used to display the actual title on the published learning path, module, or unit.

| METADATA | DESCRIPTION |
|--------------------------|---|
| <code>title</code> | The title of your learning path, module, or unit. For more information about writing titles, see the How to write titles for learning paths, modules, and units article. For tips on writing SEO-friendly titles, see the SEO: Tips for writing titles Contributor Guide article. |
| <code>description</code> | A brief description of your learning path, module, or unit. Ensure that your description is meaningful and correct. For tips on writing SEO-friendly descriptions see the SEO: How to write good meta descriptions Contributor Guide article. |

PR (pull request) review quality criteria

1/14/2022 • 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 September 22, 2021, all Docs repos and Learn repos are running under the new, smaller set of PR Review criteria covered in this section. This project has phased into 34 public repos, since beginning in January 2021. (Learn repos adopted this change on September 22.) 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 default 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 default 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.

The below criteria are outdated at this point, but still listed during this finalization stage. This page will be updated as the pilot becomes the standard operation, tentatively in November 2021.

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 <code>#sign-off</code> 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 default 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 default 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 default 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 default branch and accidentally #signed-off as ready by the author (because they didn't understand the implications of pointing to the default branch), 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 <code>#sign-off</code> comment must appear after the validation results and staging links. If the <code>#sign-off</code> 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 | <p>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 <code>docfx.json</code>, <code>.openpublishing.publish.config.json</code>, and any <code>.git*</code> 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.</p> | Helps ensure major configuration changes are reviewed by a repository owner/admin. |
| All | Repo integrity | <p>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.</p> <p>Items to watch for: <code>.DS_Store</code>, <code>desktop.ini</code>, <code>.gitignore</code>, the entire repo embedded in the root folder.</p> | Prevents repo from becoming random file storage, prevents cloning mistakes by users from being replicated to other users. |
| Docs | Repo integrity | <p>The PR contains fewer than 100 changed files, unless the PR is intentionally updating a release branch from the default branch.</p> <p>See the section in this article about large pull requests.</p> | Ensures iterative workflow and reviewability of changes. |
| Learn | Repo integrity | <p>For new modules, no more than one module per PR. Updates to existing content can go across modules in a single PR.</p> | 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 redirect file must contain a redirect for the deleted content. If the repo uses a main redirect file, redirects use only the main redirect method. File-based redirects are not allowed in repos that use main file redirection.
Repo build validation will report a Warning for any redirected file that's not deleted from the repo, so the file listed in the <code>source_path</code> of any new redirect must not exist in the repo. | Prevents usage of outdated redirect method and avoids build-blocking warnings. |
| 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 <code>title</code> attribute should be sentence cased in YML and MD files. | Basic compliance with editorial guidelines for text that may be displayed in search results. |

| APPLIES | CATEGORY | QUALITY REVIEW ITEM | RATIONALE |
|---------|--------------------|---|--|
| Learn | Metadata | YAML (.YML) files must not include <code>ROBOTS: NOINDEX</code> . | The platform automatically applies this attribute.
Manual settings revert the global update. |
| 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: Microsoft 365
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 <code>title</code> 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: MicrosoftDocs/msft-learn-repo-managers | 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 only approved HTML. Minor inline HTML tags (such as superscript, subscript, and special characters) are currently under review, but many HTML elements previously permitted by the Docs platform are now planned for deprecation, as noted on the HTML allowlist , and no longer allowed. Because HTML tables will be deprecated, new tables must be formatted only in Markdown. | Ensures alignment with basic content authoring guidelines for docs.microsoft.com. Markdown is our authoring standard. (Use of an HTML table is often an indication that the content needs to be simplified.) |
| 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 <code><svg></code> elements are present with enclosed <code><script></code> 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:
<ul style="list-style-type: none"> - 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 UI highlight 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 default 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 the default branch, 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 the default branch. 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 default 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 <code>title</code> ,
<code>description</code> , <code>author</code> ,
<code>ms.author</code> , <code>ms.date</code> ,
<code>ms.topic</code> , and
<code>ms.service</code> OR <code>ms.prod</code> , 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 <code>en-us</code> , <code>en-gb</code> , <code>en-in</code> , 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. |

Pre-publication checklist

1/14/2022 • 10 minutes to read

During the process of authoring, it's natural to stray from your original plan. There may be things you forgot to add. Or maybe you realized during the writing process that content needed to be reorganized. The design doc is there to guide the process, not to dictate it.

Use this checklist to make sure all parts of your module are technically correct and meet Learn's quality standards.

Instructional Design (ID) guidelines - High level

- Does the title meet [standard guidelines](#)?
- Do the Learning Objectives need to be updated?
- Are the Learning Objectives in the correct [format](#)?
- Do you have any prerequisites to add? Scan your module. Make sure that the prerequisites include anything the learner must know, but isn't taught. If there are no prerequisites, write "None".
- Does summary meet the [guidance](#)?
- Do [knowledge checks](#) align to learning objectives?
- All exercise units have *Exercise - * prefix in the title.
- Content must be factual rather than opinion or marketing. That is, technical content is "engineers teaching engineers".

Instructional Design (ID) guidelines - Modules

- Modules must stand alone. That is, they must manage to exist in multiple Learning Paths. For example, exercises shouldn't span across modules.
- Modules should be 60 minutes or less.
- Modules should be between 4 and 10 units.
- Modules must include conceptual learning content. That is, the module can't be solely a walk-through.
- Modules must include at least one knowledge check or exercise with task validation. We don't want a module that can be completed with "view only".
- Modules must be organized around 1-2 focused tasks with a total completion time of less than 60 minutes.
- Modules may include short supplementary videos but the content can't be primarily video-based.
- Modules must be organized around job tasks. That is, the module should be task oriented rather than subject oriented.
- Modules must follow the core Learn structure. They should begin with the standard introduction unit, contain learning content and interactivity units, and end with a summary unit.
- Did you use your [scenario](#) throughout the module to explain the content in context?

Instructional Design (ID) guidelines - Units

- Does each unit follow the [unit structure guidance](#)?
- No more than 10 minutes for learners to consume.
- Learning content must include everything needed for the learner to complete any further active-learning activity.
- Images must have appropriate license/approvals.

- No more than four screenshots.
- Include definitions for all introduced terminology.
- Include definitions for all used acronyms at first use.
- Company names used in examples or exercise must be taken from the Microsoft/CELA list of Approved fictitious names.

Instructional Design (ID) Guidelines - Units with exercises

- Provides cleanup instructions for any Azure resources created in the learner's account. Make sure to include the `azure-sandbox-cleanup` file in the summary if the module uses the Azure sandbox.
- Exercise instructions must be complete and clear. That is, the instructions must be easy enough to follow so that anyone can complete the exercise without any other required steps.
- Azure tasks that can be validated must have task validation included in the Unit `ym1` file.
- Must be based on real-world scenarios, for example, it shouldn't include a "hello, world" module.
- Code samples should include multiple programming languages.
- Coding exercises should ask the learner to write some original code and not be solely cut-and-paste.
- Provided source code should be in GitHub.
- Provided source code must be under the MIT license.
- Code on GitHub should be public and not require authentication/authorization to access.
- Code samples must follow modern coding practice.
- Don't use code comments to present new learning content. You can use them to connect previous learning content to the code sample.
- Sandbox state isn't maintained across modules and exercises can't span modules.

Instructional Design (ID) guidelines - Learning paths

- Between three and eight modules.

Video

- All MP4 video files must be hosted in Red Tiger. You may not embed a video from YouTube, Channel9, or any other hosting platform.
- All videos should be exported to 1920 pixels width by 1080 pixels height (1080p).
- All videos should be exported at a frame rate of 29.97 frames per second.
- All videos should be exported with the highest possible bitrate for both audio and video that your software will allow. The higher the better.
- Video captioning files must be upload in the TTML file format.
- All video must have high quality closed captions.
- The audio levels for voice should be set at 12 db and at 22 db for music. The audio Bitrate should be set to 320 kbps.
- Your content must be kept current with any concepts, functionality, or user interface changes to the latest version of the software.
- All Closed Captioning files must be translated into all the languages that your textual content is translated into.

Legal and Accessibility

- Content can't violate any [legal or accessibility standards](#).

Freshness

- Content must be kept up to date with any concepts, functionality, or user interface changes to the latest version of the software that is displayed in images or videos.
- Content can't be copied directly from other sources such as <https://docs.microsoft.com>.

Metadata - `index.yml`

- `title` is set.
- `description` is set, follows guidelines, and doesn't duplicate title.
- `summary` is set and follows guidelines.
- `abstract` is set and lists objectives (no periods).
- `prerequisites` value is set, or says `None`.
- `ms.date` is set to publication date.
- `author` and `ms.author` are set.
- `ms.prod` is set to `learning-nameofproduct`.
- `iconUrl` is set and points to a valid `svg`.
- All units are listed.

Metadata - Unit `.yml` files

- `title` is set.
- `description` is set, follows guidelines, and doesn't duplicate title.
- `ms.date` is set
- `author` and `ms.author` are set
- `durationInMinutes` is set and IS ACCURATE.
- Interactivity values are set correctly if enabled. Examples are `interactive`, `azureSandbox`, and `labid`.
- Knowledge checks reviewed for accuracy.
- Content has at least one knowledge check or task validation so users can't just navigate through content to get credit.

Content structure

- If you're publishing a module created inside a new content folder, verify you've added a new DocFX content rule to the `docfx.json` file for that repo that will serve the module from `/learn/modules/` instead of `/learn/{folder}`. For more information, see this [example, but swap language for your folder name](#).

PR review

- No build errors or warnings in your pull request.
- No merge conflicts in the pull request.
- All content has an Acrolinx score of 80 or higher.
- No article-related files, images, or folders are being added to the root directory of the repo.
- The pull request doesn't 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.
- For new modules, no more than one module per pull request. Updates to existing content can go across modules.
- If articles are deleted in the pull request, 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. It also ensures that

deletions are intentional.

- When an author deletes an article, the main redirect file must contain a redirect for the deleted content. If the repo uses a main redirect file, redirects use only the main redirect method. File-based redirects aren't allowed in repos that use main file redirection.
- Only Markdown (.MD), YAML (.YML), and image files are allowed in content repos.
- New files and folders introduced into the repo follow the File name and path guidelines.
- Markdown (.MD) and YAML (.YML) files must meet the [detailed Learn metadata requirements](#).
- The `ms.date` value can't be set more than five days in the future.
- The `title` attribute should be sentence cased in .yml and .md files.
- No valid spelling errors remain in the Acrolinx report. The author must fix the spelling errors.
- No remaining do-not-use terminology. The author must correct the terminology.
- Module content should be for the product area scoped for the repository. See the detailed mapping of content to repo. If content appears to be in the wrong repo, contact learn-repo-managers.
- 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. Don't publish private content, content subject to a non-disclosure agreement, or content that is otherwise confidential to the site.
- Elements that should be numbered lists are numbered. Elements that should be unordered lists are bulleted.
- 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 problem/solution canvas or plan in place for evaluating experiments.
- Articles can't 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.
- Source content doesn't contain 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 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.
- Custom Markdown elements are used where appropriate. For example, notes are coded through the [!NOTE] extension, not as plain text.
- The use of the Azure Resource Manager (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. Also, you can use "ARM template" only after spelling out at first mention. See the "Resource Manager" term in the [Microsoft Cloud Style Guide](#).
- Don't use the word `quiz` anywhere in the content. The only acceptable use is the YAML property `quiz:`.
- 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.
- SVG is the preferred format for conceptual art. It scales to whatever the size of the browser window. However, SVG files that are scripted (where SVG elements are present with enclosed script tags) aren't 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.
- All media links must point to media within the repo. You should have no hyperlinks to media hosted on [my-sharepoint.com](#), personal URLs, other websites, or other repositories.
- Links to pages on [microsoft.com](#) websites are coded as locale agnostic. Don't include `en-us`, `en-gb`, `en-in`, or any other locale in links to these sites. TechNet/MSDN forum links are an exception. Locales can't be removed from forum links on these sites.
- The content preview must be clean on staging in both light and dark view. It can't contain any obvious formatting issues, such as:
 - 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 because of faulty indentation.
- Leftover merge-conflict markers.
- Content must preview legibly in both light and dark view. Pay attention to conceptual artwork rendering correctly in dark view.
- All embedded videos must be on RedTiger hosting. Any external video links must go to Microsoft channels, for example, a Microsoft-sponsored channel on YouTube.
- Images use the correct callout style and color, and screenshots use the correct border and placeholder style. See the [guidance for screenshots](#).
- Images include alt-text. See the [guidance for alt text](#).
- All titles and headings are sentence case, per MSFT style.
- Verify all external hyperlinks resolve to content that exists and that appears to be high-quality and credible.
- Each new module must contain at least three units (YAML files) and three corresponding Markdown files. One unit must be an introduction, one unit must be a summary, and the introduction unit must have learning objectives and prerequisites. The module must include at least one knowledge check or task validation.

Formal editorial review for Microsoft Learn content

1/14/2022 • 2 minutes to read

The **formal editorial review** is intended to focus purely on English spelling, grammar, and style guide adherence.

- For Learn content created by the Developer Relation org, we leverage the same formal editorial service that the rest of the Developer Relations content org uses. For more information about this team, view the [Docs Contributor Guide](#)
- For BAG content, we leverage the editorial team that serves BAG's Docs content. For more information, contact Margo Crandall.

Scope

The scope for content formal editorial review is lightweight. It is expected that the content developer has already used Acrolinx and most issues would be caught in previous reviews.

The scope for the content formal editorial review is:

- Heading casing
- Basic Microsoft style
- Spelling
- Grammar
- Azure terms
- Typos
- Basic punctuation

NOTE

[Under review - Proposed additions to scope:

- **Title** that follows our [guidance](#)
- **Summary** that follows our [guidance](#)
- **Learning objectives** that follow our [guidance](#)
- **Prerequisites** that follow our [guidance](#)
- **Scenario** a real-world scenario that follows our [guidance](#)]
- Assessments: Do the assessments align with and promote competency on the unit objectives?
- Clarity: Does the Design Doc make clear where exercises and knowledge checks occur?
- Interaction: Are there opportunities for user interaction within the course (exercises and/or knowledge checks)?
- Practice: Does the course provide opportunities for deliberate practice?
- Casual voice: students learn better from a casual voice rather than formal language
- Plain English: Avoid verbose or flowery language. Be clear and direct.
- Story: Does the course tell a story, is there a logical flow from one unit to the next?
- "What is" content: Do we define all necessary terms?
- Completeness: Do we include all info learners need to be successful in the exercise/activity?
- Terminology: Do we avoid unnecessary technical jargon?
- Imagery: Are there opportunities for images or diagrams that would be useful for learning?

Using Acrolinx

Acrolinx is software that provides content authors with automated feedback on grammar, spelling, punctuation, writing style, terminology, and voice. For repos that it is enabled on, it is available both upstream and locally.

To view the status of what Learn repos have Acrolinx enabled, see the [Microsoft Learn GitHub repositories](#) article.

If you are working in a Learn repo that has Acrolinx enabled, view the [Acrolinx articles in the Docs Contributors Guide](#) for more information on how to set it up locally and use it.

Submit a request for formal editorial review

Use this template to create an "Edit request" task for the Aquent vendor team, in Azure DevOps:

<https://aka.ms/tc-edit-request>

Notes & decisions about editorial

This is a running list of notes and decisions that have been made that should eventually be incorporated into a Microsoft Learn content style guide.

- Guidance for ending punctuation for knowledge check answer options is to stay consistent within the questions in that unit. Note that multiple questions within the same unit will be shown on the page at the same time.
- Titles should be sentence cased.
- Voice is friendly.

Resources

All of the links below are from the [Microsoft Docs Contributor's Guide](#). Look there for more resources too!

- [Style guides for Learn and Docs content](#)
- [The five writing principles](#)
- [Writing principles checklist](#)
- [Writing principles job aid](#)
- [Everyday word list](#)
- [Accessibility guidelines](#)
- [Alt text](#)
- [Product names and terminology](#)
- [Legal guidelines](#)
- [Format text](#)
- [PoliCheck for GitHub](#)

How to build Learn URLs

1/14/2022 • 2 minutes to read

When you submit a PR, the build report should automatically provide the URLs to the edited content. You can manually "build" one yourself using the procedures below for different reasons. For example, manually run a build if the URLs aren't available in the build report.

Your URL will be affected by the system that you want to view the content in:

- **Production:** The Production URLs are generated when the content is published to the live site.
- **Review:** These URLs are for the content that isn't yet published. The review URLs are generated when you submit your PR and can be seen in the PR build report.

Learn URL structure

A typical Learn URL is made up of the following parts:

- **Microsoft docs home link:** `docs.microsoft.com` or `review.docs.microsoft.com` depending on whether it's a production URL or review URL.
- `/learn/`: All Learn URLs for modules, units, and learning paths roll up to `/learn/` no matter the location of the original content. The repo and the organizational folder structure used to store the content doesn't show up in the URL.
- `/modules/` or `/path/`: Depending on whether you're viewing module/unit or a learning path.
- **Module or path:** Folder name and location that stores the source YAML files.

For example, see the production URL for the following two modules stored in different repos.

- `align-requirements-in-azure` stored in `learn-pr` repo, **azure** product organizational folder gets this production URL: `https://docs.microsoft.com/learn/modules/align-requirements-in-azure/`.
- `analyze-dynamics-365-sales-data` stored in `learn-dynamics-pr` repo, **dyn365-sales** product organizational folder gets this production URL: `https://docs.microsoft.com/learn/modules/analyze-dynamics-365-sales-data/`.

Review URLs for a specific branch

You can manually create review URLs for a specific branch by adding `?branch=<branch name>` to the end of the review URL. To construct the URL for previewing your content that is still in your unmerged PR, add `pr-en-us-<PR#>` as the branch name.

Build module URLs

The module URL takes you to the module landing page (front door), which shows all of the information in a module `index.yml` file, and also provides links to all of the individual unit pages.

Module URL structure

- **Production:** `https://docs.microsoft.com/learn/modules/<module-folder-name>`
- **Review:** `https://review.docs.microsoft.com/learn/modules/<module-folder-name>/?branch=<branch-name>`

The following examples are for: [design-for-security-in-azure](#).

- **Production:** `https://docs.microsoft.com/learn/modules/design-for-security-in-azure/`
- **Review:** `https://review.docs.microsoft.com/learn/modules/design-for-security-in-azure/?branch=master`

Build unit URLs

The unit URL takes you to an individual unit page. This page shows some content from the unit YML, for example H1, derived from title, and knowledge check. The page also shows the content from the unit MD. These files can be difficult to construct. It's easier to just construct the module URL and use the unit links from that URL.

Unit URL structure

- Production:

```
https://docs.microsoft.com/learn/modules/<module-folder-name>/<unit-yaml-file-name-without-extension>
```

- Review:

```
https://review.docs.microsoft.com/learn/modules/<module-folder-name>/<unit-yaml-file-name-without-extension>?branch=<branch-name>
```

The following examples are for the unit 1 (introduction) in the module folder name (in the GitHub repo): [design-for-security-in-azure](#).

- Production: <https://docs.microsoft.com/learn/modules/design-for-security-in-azure/1-introduction>

- Review:

```
https://review.docs.microsoft.com/learn/modules/design-for-security-in-azure/1-introduction?branch=master
```

Build learning path URLs

The learning path URL takes you to the learning path landing page. This page presents the modules in suggested sequence. Learners are free to choose modules in any order they like from here.

Learning Path URL Structure

- Production: [https://docs.microsoft.com/learn/paths/<lp-folder-name>/](https://docs.microsoft.com/learn/paths/<lp-folder-name>)

- Review: <https://review.docs.microsoft.com/learn/paths/<lp-folder-name>/?branch=<branch-name>>

The following examples are for the learning path folder name (in GitHub repo): [secure-your-cloud-data](#).

- Production: <https://docs.microsoft.com/learn/paths/secure-your-cloud-data/>

- Review: <https://review.docs.microsoft.com/learn/paths/secure-your-cloud-data/?branch=master>

PR (pull request) review quality criteria

1/14/2022 • 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 September 22, 2021, all Docs repos and Learn repos are running under the new, smaller set of PR Review criteria covered in this section. This project has phased into 34 public repos, since beginning in January 2021. (Learn repos adopted this change on September 22.) 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 default 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 default 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.

The below criteria are outdated at this point, but still listed during this finalization stage. This page will be updated as the pilot becomes the standard operation, tentatively in November 2021.

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 <code>#sign-off</code> 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 default 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 default 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 default 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 default branch and accidentally #signed-off as ready by the author (because they didn't understand the implications of pointing to the default branch), 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 <code>#sign-off</code> comment must appear after the validation results and staging links. If the <code>#sign-off</code> 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 | <p>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 <code>docfx.json</code>, <code>.openpublishing.publish.config.json</code>, and any <code>.git*</code> 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.</p> | Helps ensure major configuration changes are reviewed by a repository owner/admin. |
| All | Repo integrity | <p>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.</p> <p>Items to watch for: <code>.DS_Store</code>, <code>desktop.ini</code>, <code>.gitignore</code>, the entire repo embedded in the root folder.</p> | Prevents repo from becoming random file storage, prevents cloning mistakes by users from being replicated to other users. |
| Docs | Repo integrity | <p>The PR contains fewer than 100 changed files, unless the PR is intentionally updating a release branch from the default branch.</p> <p>See the section in this article about large pull requests.</p> | Ensures iterative workflow and reviewability of changes. |
| Learn | Repo integrity | <p>For new modules, no more than one module per PR. Updates to existing content can go across modules in a single PR.</p> | 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 redirect file must contain a redirect for the deleted content. If the repo uses a main redirect file, redirects use only the main redirect method. File-based redirects are not allowed in repos that use main file redirection.
Repo build validation will report a Warning for any redirected file that's not deleted from the repo, so the file listed in the <code>source_path</code> of any new redirect must not exist in the repo. | Prevents usage of outdated redirect method and avoids build-blocking warnings. |
| 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 <code>title</code> attribute should be sentence cased in YML and MD files. | Basic compliance with editorial guidelines for text that may be displayed in search results. |

| APPLIES | CATEGORY | QUALITY REVIEW ITEM | RATIONALE |
|---------|--------------------|---|--|
| Learn | Metadata | YAML (.YML) files must not include <code>ROBOTS: NOINDEX</code> . | The platform automatically applies this attribute.
Manual settings revert the global update. |
| 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: Microsoft 365
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 <code>title</code> 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: MicrosoftDocs/msft-learn-repo-managers | 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 only approved HTML. Minor inline HTML tags (such as superscript, subscript, and special characters) are currently under review, but many HTML elements previously permitted by the Docs platform are now planned for deprecation, as noted on the HTML allowlist , and no longer allowed. Because HTML tables will be deprecated, new tables must be formatted only in Markdown. | Ensures alignment with basic content authoring guidelines for docs.microsoft.com. Markdown is our authoring standard. (Use of an HTML table is often an indication that the content needs to be simplified.) |
| 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 <code><svg></code> elements are present with enclosed <code><script></code> 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:
<ul style="list-style-type: none"> - 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 UI highlight 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 default 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 the default branch, 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 the default branch. 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 default 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 <code>title</code> ,
<code>description</code> , <code>author</code> ,
<code>ms.author</code> , <code>ms.date</code> ,
<code>ms.topic</code> , and
<code>ms.service</code> OR <code>ms.prod</code> , 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 <code>en-us</code> , <code>en-gb</code> , <code>en-in</code> , 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. |

Launching a new group

1/14/2022 • 2 minutes to read

The onboarding team PM is responsible for providing support to an external group to publish their content on the Microsoft Learn platform. One of the important steps is launching the content on the live platform. The purpose of this article is to provide a list of items that should be checked/verified when launching a new group.

The checklist is divided into the following groups:

- Support activity - Includes items to be checked from support perspective
- Repo management - Includes the items for managing and preparing the repo for final launch
- General verification - Includes general pre-launch UI verification
- Communication - Includes things to be communicated to the groups

Support activity

1. Set up support triage security group for the new group; update support pages with info about team.
2. Update Support FAQ page to reflect new products. In particular the language/product support section.

Repo management

1. Merge content branches into master, test on review.
2. Merge content branches from master to live, test on live.
3. Create new release branches for special cases, though most [release branches should be requested via Content Production Services](#).
4. Verify and adjust repo permissions to desired "in production" state. Verify that the branches are protected and correct access is provided to users.

The following one-time activities should be done when launching a new repo for a group. For details on these tasks, refer to [go-live tasks section of the Creating new Learn repos](#)

1. Do final "go-live" repo tasks.
2. Set up cpub publishing for the new repo (master-live 2x daily, 1x week loc snapshot).

General verification

1. Verify home page updates are made (if applicable).
2. Launch new landing pages (if applicable).
3. Sync LearnShared from master-live, if changes were made to it.

Communication

Communicate the following to the new group:

1. Support process and expectations.
2. Metrics decks (how to access).
3. How loc works (snapshots and updates).

Add notifications to modules

1/14/2022 • 2 minutes to read

Overview

There are multiple scenarios in Learn that justify the need to show a notification to users. For example, these scenarios include lab disabled, content will temporarily be unavailable, and so on. The system only has the capability to show a special notification when we have **disabled labs**. In the future, there will be more notification options.

IMPORTANT

If you have a need to add a notification to a module other than disabling labs, reach out to your content manager for recommendations of how to proceed.

Add a 'lab disabled' message

If a lab of any type is having issues, you can disable the lab and trigger the messaging via a YAML file in LearnShared.

Disabling the lab helps the learner by:

- Preventing confusion and frustration by clearly notifying them that they may be unable to activate the lab, and as such, unable to complete the module.
- Allowing them to choose to delay working on the module or shift to another module that is working correctly.
- Reducing our support burden. Learners are less likely to submit a bug for a malfunctioning lab if they know we're already aware.

Once the YAML file is updated, the temporarily unavailable message will show to users immediately for all content (in English and other languages). Also, the **Activate sandbox** and **Launch VM mode** buttons will be blocked.

Notification example

✓ 100 XP

Sample Unit 2 with Cloud Shell

11 minutes

We're sorry. The sandbox needed to complete the exercises in this module is temporarily unavailable.

During the build process, Packer creates temporary Azure resources for the base VM. To capture that base VM for use as an image, you must define a resource group. An Azure resource group is a logical container into which Azure resources are deployed and managed.

Instructions

NOTE

Only *Learn repo administrators* can add this messaging. Reach out to learn-repo-managers@microsoft.com if you need a message enabled.

1. Go to the module availability YAML file in LearnShared:

<https://github.com/MicrosoftDocs/LearnShared/blob/master/LearnShared/module-availability/index.yml>

2. Depending on what you need to disable, add the following properties:

| PROPERTY | VALUE | REQUIRED | DESCRIPTION |
|----------------------------------|--------------------------------|----------|---|
| <code>disableAllLabs</code> | <code>true / false</code> | Required | <p><code>True</code> shows the disabled message on all unit pages with labs.
<i>Message if set to true : We're sorry. The lab environment needed to complete the exercises in this module is temporarily unavailable.</i></p> |
| <code>disableAllSandboxes</code> | <code>true / false</code> | Required | <p><code>True</code> shows the disabled message on all unit pages with sandboxes.
<i>Message if set to true : We're sorry. The sandbox needed to complete the exercises in this module is temporarily unavailable.</i></p> |
| <code>disabledSandboxes</code> | List of module UIDs to disable | Optional | <p>The disabled message will show on these modules' unit pages only.
<i>Message if set to true : We're sorry. The sandbox needed to complete the exercises in this module is temporarily unavailable.</i></p> |
| <code>disabledLabs</code> | List of module UIDs to disable | Optional | <p>The disabled message will show on these modules' unit pages only.
<i>Message to users: We're sorry. The lab environment needed to complete the exercises in this module is temporarily unavailable.</i></p> |

Sample code for ALL interactivity disabled

```
### YamlMime:ModuleAvailability

disableAllLabs: true

disableAllSandboxes: true
```

Sample code for NO interactivity disabled

```
### YamlMime:ModuleAvailability

disableAllLabs: false

disableAllSandboxes: false
```

Sample code for specific labs and sandboxes disabled

```
### YamlMime:ModuleAvailability

disableAllLabs: false

disableAllSandboxes: false

disabledSandboxes:
  - "example.example-module-uid-1"

disabledLabs:
  - "example.example-module-uid-2"
  - "example.example-module-uid-3"
```

Other documentation

For more information about temporarily unavailable modules, see [Show temporarily unavailable status of modules](#).

Add a new content area path in Azure DevOps

1/14/2022 • 2 minutes to read

Overview

Work items within Azure DevOps are categorized by an **Area Path** value. These can help compartmentalize things for getting customer feedback into the hands of the appropriate content team. When we add content that may not fit best in an existing area path, someone with administrator permissions in Azure DevOps can create a new one.

Additionally, when new modules are created that will fall in a new area path, the mapping from module to an auto-assigned area path on the customer feedback system can be created to make sure feedback from the new modules is seen by the right content team triage queries.

Add the area path in Azure DevOps

First, create the new area path under an existing area path in the Microsoft Learn Azure DevOps project. This requires special permissions in Azure DevOps.

1. Navigate to the [Areas settings](#).
2. Under the **Customer Feedback > Content** area group, select the appropriate parent node for your new path.
3. Select **New child** from the top of the area path list. Alternatively, you can right-click an existing node and select **New child** from the contextual menu that appears.
4. Enter the new child value in **Area name**.
5. To create your new area, select **Save and close**.

The new area will now be available for use in work items.

Map modules to an area path to route customer feedback

Once an area path exists, customer feedback can be mapped to the module via a configuration system controlled by the Engineering team. (Jeremy Danyow [[jdanyow](#) on GitHub] has previously offered to validate pull requests with configuration changes and may be a good place to start for future PRs.)

1. Fork the [MicrosoftDocs/DocsRoot](#) GitHub repo to your GitHub account.
2. In the [feedback configuration JSON file](#), add the desired mappings from a content repo folder to an area path.

For example, to map feedback submitted to modules within the **dyn365-commerce** folder of the [MicrosoftDocs/learn-dynamics-pr](#) repo to the **Dynamics > Commerce** area path in Azure DevOps, the following line was added.

```
{"repo": "learn-dynamics-pr/dyn365-commerce", "area": "Dynamics/Commerce"},
```

3. Submit a pull request with those changes with an appropriate name and description. (Here is an [example PR](#).)

When that configuration PR is merged, incoming feedback from the given modules will be set to the correct area path as they are imported into Azure DevOps.

Microsoft Learn support overview

1/14/2022 • 2 minutes to read

When learners consume Microsoft Learn content, they have multiple ways to provide feedback and request support.

Report an issue via Report feedback form

1. Users select **reporting an issue** from the link featured in each unit.
2. User is directed to the [Report feedback](#) form.

Depending on the page they're viewing, information about the page will be included in the report to help with triage and issue resolution.

- Accessible to use by anyone browsing Microsoft Learn content.
- No sign-in required to provide feedback.
- Feedback is private to DevRel.
- Presented to learners as one-way communication. The expectation is set that we're not going to respond.

3. User selects a classification of issue from a set of radio button lists.
4. Feedback is saved to a system that imports it into Azure DevOps. Once there, it's routed to the relevant stakeholder to fix.

Star ratings and verbatim comments for content

- See the [New Rating System - Epic spec](#).
- Accessible by users who have attempted Microsoft Learn modules and are authenticated to the Docs platform.
- Star rating is public to other users and shown on the module and path tiles.
- Written feedback (verbatim comments) is private to DevRel.

Azure DevOps Customer Feedback workflow

1/14/2022 • 10 minutes to read

Overview of Reporting feedback customer workflow

Users of Microsoft Learn can submit bug tickets by selecting the [reporting an issue](#) link found on the bottom of each unit page.

Using the **Report feedback** form, users self-select from a sequence of drop-down menus and then provide reproduction steps and screenshots of their issue. The form is a one-way means of communication. Learn doesn't collect contact information or personal data from the user during this process.

Triage workflow (get feedback to the right team)

- Based on the user's selection, a tool called the Docs Web Service converts every [reporting an issue](#) ticket into an Azure DevOps Customer Feedback work item in the Microsoft Learn project. It also fills in the following fields in the work item. See [Customer Feedback work item fields](#) for descriptions and team uses of the Customer Feedback work item fields.
 - **Area:** Path based on category selection and location of user before selecting [reporting an issue](#).
 - **Description:** Text entered by user describing the problem.
 - **URL:** Web location where user reported the issue, including the requestor's locale.
 - **UID:** Unit ID from the YAML metadata representing the page at the given URL.
 - **Repo MSFT Learn:** Repo name for where the content lives on GitHub.
 - **Requestor locale:** Locale of the user reporting the issue.
 - **Content locale:** Locale of the original content at the given URL.
 - **Session ID:** JSLL/AppInsights.
 - **Attachments:** Screenshots uploaded by users to help describe their problem.
- For issues reported for non-english Learn modules:
 - The items are machine translated.
 - The original feedback, in the original language, is added as a comment in the **Discussion** field.
- A vendor triages all non-categorized work items (Microsoft Learn/Customer Feedback) daily.

Manually submitting feedback

Microsoft Learn internal users and partners can also submit feedback by manually creating a Customer Feedback work item. To do so, follow these steps:

1. Create a [New Customer Feedback work item](#) in the Microsoft Learn Azure DevOps organization.
2. Enter short description of the issue in the **Title** field.
3. Add reproduction steps in the **Description Field**.
4. If necessary, you can add screenshots with the **Add Attachment** feature.
5. Set the **Severity** value based on [Microsoft Learn Severity definitions](#).
6. Fill out the following work item fields:
 - **URL:** enter the current browser page URL

- **UID**: enter the meta tag value for **uid**
- **Repo MSFT Learn**: enter the meta tag value for **search.ms_docsetname**
- **Content Team**: do your best to set the right value

TIP

Many of these fields can be filled out from information found in your browser's **View source HTML** in the different metadata tags: `<meta name="..." content="..." ... />`

7. Assign the work item based on the **Content Team** category below.

8. Update **Area Path** according to the organizational charts below.

IMPORTANT

To get the **MS.Author**, **URL**, **Repo MSFT Learn** and **UID** values, view the page source: **Crtl + U**.

Resolve your Customer Feedback work items

When you're assigned a Customer Feedback content work item, you can fix it in several different ways. Once you resolve an issue, pick a value for the **Cause of issue** field to help us diagnose ways to improve Learn content and prevent future user-reported issues. For details on the **Cause of issue** field choices, see [Customer Feedback work item fields](#).

Fix the issue

You fix the issue and submit a pull request (PR) to the appropriate content repo.

1. Link the GitHub PR from the **Development > Add Link** button, if you haven't used the `AB#{WorkItemNumber}` syntax in the GitHub description.
2. Set the **Resolved Reason** field to **Fixed**.
3. If there were any useful details about the fix, add them in a comment.
4. Select a value for **Cause of issue** that indicates the root cause of this reported content problem, like **Content problem**, **Content needed clarity**, or **Covered product changed**.
5. When the PR reviewer merges the PR, change the work item's **State** to **Closed**.

Unrelated issue

It turns out the issue is unrelated to your content: platform issues, localization issues, and so on.

1. Reassign the work item according to the [feedback taxonomy](#), setting the **Area** and **Assigned** fields to match.
2. Select a value for **Cause of issue** that indicates the root cause of this reported content problem.
3. Add a comment describing why you think the issue is related to the assigned group.

User error issue

The issue seems to be a result of customer error.

You think that more detail or content in the module could prevent the issue:

1. Submit any changes in a PR.
2. Follow the steps for that process above, likely selecting **Content needed clarity** as the value for **Cause of issue**.

You don't think more content clarity or details will help:

1. Set **Learn Feedback Category** to **User Error**.

2. Set **Resolved Reason** to **As Designed** or **Cannot Reproduce**, as appropriate for the work item.
3. Add a comment describing the situation. It can help identify common user errors that may mean the content could use more clarity or context.
4. Select a value for **Cause of issue** that indicates the root cause of this reported content problem, likely **Cannot reproduce** or **User error**.
5. Set the work item's **State** to **Closed**.

Unsure an issue exists

You're unsure if there's an issue:

1. Set **Triage Status** to **More Info**.
2. Set **Resolved Reason** to **Cannot Reproduce** or **Deferred**, as appropriate for the work item.
3. Add a comment describing the situation.

It can help identify common user errors that may mean the content could use more clarity or context.

4. Select a value for **Cause of issue** that indicates the root cause of this reported content problem, likely **Cannot reproduce** or **Other**.
5. Set the work item's **State** to **Closed**.

Customer feedback taxonomy

Content issue

- **Example Error Messages:** Incorrect Information/Spelling/Content Mistake; I have questions about the learning content
- **Area Path:** Microsoft Learn\Customer Feedback\Content\{Product or Product Family}\{Content sub-team}

| CONTENT TEAM | REPO | AREA PATH | TEAM SUPPORT LEAD | ASSIGN TO: |
|--------------------------|----------|--|-------------------|------------|
| Azure Content | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure | Adrian Stevens | ms.author |
| Cloud Advocates (CA) | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure\Advocates | John Papa | ms.author |
| Core Azure Content | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure\Core | Adrian Stevens | ms.author |
| WWL Data&AI Content | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure\Data & AI | Adrian Stevens | ms.author |
| Azure Databricks Content | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure\Databricks | Kiran Chandratrey | ms.author |

| CONTENT TEAM | REPO | AREA PATH | TEAM SUPPORT LEAD | ASSIGN TO: |
|-------------------------------------|-------------------|---|-------------------|---------------------------|
| Azure DevOps Content | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure\DevOps | Thomas Petchel | ms.author |
| Azure Fundamentals | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure\Fundamentals | Rob Barefoot | robbarefoot@microsoft.com |
| Azure Student Evangelism | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure\Student Evangelism | Shana Matthews | ms.author |
| Azure WWL MBA Content | learn-pr | Microsoft Learn\Customer Feedback\Content\Azure\WWL MBA | Benjamin Jiang | pdets@microsoft.com |
| .NET Content | learn-pr | Microsoft Learn\Customer Feedback\Content.NET | Scott Addie | ms.author |
| Power Platform Content (BAG) | learn-bizapps-pr | Microsoft Learn\Customer Feedback\Content\Power Platform | Dave Beasley | ms.author |
| Power Automate (Flow) Content (BAG) | learn-bizapps-pr | Microsoft Learn\Customer Feedback\Content\Power Platform\Flow | Dave Beasley | ms.author |
| PowerBI Content (BAG) | learn-bizapps-pr | Microsoft Learn\Customer Feedback\Content\Power Platform\Power BI | Robert Deupree | ms.author |
| Power Apps Content (BAG) | learn-bizapps-pr | Microsoft Learn\Customer Feedback\Content\Power Platform\Power Apps | Dave Beasley | ms.author |
| Dynamics 365 Sales | learn-dynamics-pr | Microsoft Learn\Customer Feedback\Content\Dynamics | Dave Beasley | ms.author |

| CONTENT TEAM | REPO | AREA PATH | TEAM SUPPORT LEAD | ASSIGN TO: |
|--------------------------------------|--------------------|--|------------------------------|------------------------|
| Dynamics 365 Marketing | learn-dynamics-pr | Microsoft Learn\Customer Feedback\Content\Dynamics | Dave Beasley | ms.author |
| Dynamics 365 Customer Service | learn-dynamics-pr | Microsoft Learn\Customer Feedback\Content\Dynamics | Dave Beasley | ms.author |
| Dynamics 365 Field Service | learn-dynamics-pr | Microsoft Learn\Customer Feedback\Content\Dynamics | Dave Beasley | ms.author |
| Dynamics 365 Finance | learn-dynamics-pr | Microsoft Learn\Customer Feedback\Content\Dynamics | Claire Nielsen | ms.author |
| Dynamics 365 Supply Chain Management | learn-dynamics-pr | Microsoft Learn\Customer Feedback\Content\Dynamics | Claire Nielsen | ms.author |
| Dynamics 365 Business Central | learn-dynamics-pr | Microsoft Learn\Customer Feedback\Content\Dynamics | Kevin Schimke | ms.author |
| Dynamics WWL MBA Content | learn-dynamics-pr | Microsoft Learn\Customer Feedback\Content\Dynamics\WWL MBA | WWL CW Sustained Engineering | cwsuseng@microsoft.com |
| Windows Content | learn-windows-pr | Microsoft Learn\Customer Feedback\Content\Windows | John Kennedy | ms.author |
| Microsoft 365 Content | learn-m365-pr | Microsoft Learn\Customer Feedback\Content\M365 | Liza Poggemeyer | ms.author |
| Microsoft 365 WWL MBA Content | learn-pr | Microsoft Learn\Customer Feedback\Content\M365\WWL MBA | WWL CW Sustained Engineering | cwsuseng@microsoft.com |
| SQL Server Content | learn-sqlserver-pr | Microsoft Learn\Customer Feedback\Content\SQL Server | Terry Christiani | terrychr@microsoft.com |

| CONTENT TEAM | REPO | AREA PATH | TEAM SUPPORT LEAD | ASSIGN TO: |
|-----------------|------------------|---|-------------------|------------|
| Xamarin Content | learn-xamarin-pr | Microsoft Learn\Customer Feedback\Content\Xamarin | Adam Patridge | ms.author |

Engineering issue

| ISSUE TYPE | AREA PATH | PM/TEAM LEAD | ASSIGN TO: | EXAMPLES/KEYWORDS |
|---|----------------------------|-----------------|------------------------|--|
| Gamification | \Engineering\Gamification | Pam Spier | paspier@microsoft.com | I'm not granted XP, badges, trophies for completion, achievements not showing on profile, Cloud Skills Challenge |
| Interactivity | \Engineering\Interactivity | Yun Lu | lyn@microsoft.com | Sandbox, Cloud Shell, knowledge check, check your work |
| Profile | \Engineering\Profile | Hui Xie | huxie@microsoft.com | Problems with registration/signing-in, profile merging |
| User Experience, General, Feature request | \Engineering\Experience | Daniel Stafford | dastaffo@microsoft.com | Video, UX |

Localization issue

| ISSUE TYPE | AREA PATH | TEAM LEAD | ASSIGN TO |
|---|---|----------------|-----------|
| Azure issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\Azure | Bhargavi Desai | (Default) |
| SQL Server issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\SQL Server | Bhargavi Desai | (Default) |
| Xamarin issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\Xamarin | Bhargavi Desai | (Default) |
| ASP.NET issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization.NET | Bhargavi Desai | (Default) |

| ISSUE TYPE | AREA PATH | TEAM LEAD | ASSIGN TO |
|---|--|-------------------|-----------|
| Windows issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\Windows | Bhargavi Desai | (Default) |
| Power Platform overall issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\Power Platform | Robin Sacek | (Default) |
| Power BI issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\Power Platform\Power BI | Bhargavi Desai | (Default) |
| Power Flow Issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\Power Platform\Flow | Robin Sacek | (Default) |
| Power Apps Issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\Power Platform\Power Apps | Robin Sacek | (Default) |
| Microsoft 365 Issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\M365 | Charles Hemstreet | (Default) |
| Dynamics Issues with translation quality; Non-translation localization issues | Microsoft Learn\Customer Feedback\Localization\Dynamics | Robin Sacek | (Default) |

Other

| ISSUE TYPE | AREA PATH | TEAM LEAD | ASSIGN TO |
|--|-----------------------------------|------------|----------------------|
| General feedback; Blank; Not-enough info | Microsoft Learn\Customer Feedback | Sam Adadow | Saadad@microsoft.com |

NOTE

For any unknown issues, please put in the default [Microsoft Learn/Customer Feedback] Area Path to be re-triaged by our vendor team.

Severity and SLAs

| SEVERITY | TITLE | DESCRIPTION | EXAMPLES | RESOLUTION EXPECTATIONS | SLA |
|----------|-------|-------------|----------|-------------------------|-----|
| | | | | | |

| SEVERITY | TITLE | DESCRIPTION | EXAMPLES | RESOLUTION EXPECTATIONS | SLA |
|----------|----------|--|--|--|----------|
| 1 | Critical | User can't finish content because of issue.

No workaround. | <ul style="list-style-type: none"> • Site down. • 404 error. • Labs don't work. • Sandbox issues. • Product name/trademark issues. • Compliance issues. | <ul style="list-style-type: none"> • Must fix before shipping. • Inform Engineering, if related. | 48 hours |
| 2 | High | User has difficulty completing content because of issue.

Workaround exists, but user has to figure it out themselves. | <ul style="list-style-type: none"> • Broken images. • Videos not working. • Editorial issues that aren't P0 that affect multiple pieces of content.
Example: includes file issues. • Merge brackets still in content. • Incorrect "messaging."
Example: Not how product teams wanted to talk about the issue. | <ul style="list-style-type: none"> • Must fix before shipping • Bulk update and push fixes that are ready two to three times a week. | One week |

| Severity | Title | Description | Examples | Resolution Expectations | SLA |
|----------|--------|---|--|--|-----------|
| 3 | Medium | Content is functional but visibly flawed. | <ul style="list-style-type: none"> • Typo in non-code content. • Blurry, hard-to-read images. • Incorrect numbering. • Metadata update. | <ul style="list-style-type: none"> • Must fix before shipping. • Bulk update and push fixes once a week. | Two weeks |
| 4 | Low | Issue doesn't significantly degrade functionality or quality. | <ul style="list-style-type: none"> • Non-urgent editorial issue. Examples: grammar, bolding, tone, and so on. • Feature requests. • Content requests. • Non-issue work • Ongoing updates. | <ul style="list-style-type: none"> • Don't have to fix to ship. • Up to content manager. • Bulk update and push when a large number of issues accumulate. | One month |

Recommended content feedback resolution workflow

Assign work items to the relevant content author or group in Azure DevOps. Assign each to the appropriate **Area Path**. Example: Microsoft Learn\Customer Feedback\Content\Azure\Core.

Find your relevant feedback

You can use a query to find assigned tickets:

- Work Item Type = Customer Feedback
- Area Path = Microsoft Learn\Customer Feedback\Content\{Product or Product Family}\{Content subteam}

How are other teams doing it?

We recommend that you triage issues daily. At a minimum, triage issues few times a week. The Azure Content team rotates triage members daily to get all members acquainted with the support issues and processed.

In the triage meeting, the Azure Content team reviews the tickets to:

- Set the **Severity level**.
- Populate **Issue Found By**.
- Set the **Triage Status** to:
 - **Approved** if it's an issue.

- Rejected and set **Status** to **Closed** if it isn't an issue that the team wants to fix.
- Assign to responsible team member and add a note in the **Discussion** section.

TIP

We recommend not assigning duplicate issues. Instead try to:

- Go to **Links** and add a link to main issue with **Link Type** set to **Duplicate Of**.
- Set **Triage Status** to **Approve**.
- Set to **Closed** with a **Reason of Duplicate**.

To bug burn, resolve the issue within the recommended SLA for the Severity level.

Caution

Use caution with the status of issues that you've fixed, but haven't yet republished the article.

We recommend using **Related Work** to link the Customer Feedback ticket to the main bug ticket and set **State** to **In Progress**. If you've burned the bug, update the **State** of all related work items to **Closed**.

Resources

- [Support Dashboard 1](#)
- [Support Dashboard 2](#)
- [create a bug here](#)
- [open a LSI](#)
- Support Contact: Sam Adadow
- Azure DevOps contact: Mark Smith, Ashley Johnson, or Adam Patridge

Internal support escalation and tracking

1/14/2022 • 2 minutes to read

Troubleshooting known issues

For a list of current known issues and the recommended fix/workaround, refer to our publicly facing [Troubleshooting and FAQ](#) pages.

Report a content issue

All Learn content issues should be submitted via the reporting link found at the bottom of each Learn page. This action ensures all issues are reported consistently. It also ensures issues end up in the right place for teams to triage and resolve.

1. Go to the Unit page of the content where the issue was found.
2. Select the **Report an Issue** link found at the bottom of the page.
3. You'll be redirected to the **Report Feedback** form where you can describe the issue that you're seeing.
4. This action automatically creates a Customer Feedback work item in DevOps and routes the issue to the correct content owner.

Raising Live Site Incidents (LSI)

If there's a Live Site Incident that affects multiple users on MS Learn, create a ticket for this issue in the [C + AI Dev Rel Site Help, Incident Management System \(IcM\)](#).

There it will be escalated and you can follow up with the end users by tracking the ticket through resolution.

Daily triage process

1/14/2022 • 4 minutes to read

General triage

1. Go to the **Need to Triage** query.
2. Using Mometa-Bot or by viewing the source via **CTRL+U** on the page URL, update the work item so that it contains:
 - Correct area path as dictated by the [Triage and Support Workflow](#)
 - For issues related to Profile or Gamification, assign work items to v-anbo@microsoft.com instead of Scott Ge.
 - For content-related issues, assign tickets to respective teams based on the courseURL. Obtain the area path from the Momenta-Bot.
 - Any missing meta data from Mometa-Bot, such as:
 - Module ID
 - Unit ID (UID)
 - URL
 - Learn Repo
 - Assign **severity** based on the [severity definitions](#).
 - If this issue is a non-English language issue, the item may need to be machine translated. It should be updated with the translated feedback pasted into the description of the work item as a comment in the **Discussion** field.
 - If it's a content issue, assign it to the **ms.author**.
 - If the item is a feature request, add the **Feature request** tag.
 - If the item is related to Certification, then:
 - Add the **Certification** tag.
 - Set **State** to **Closed**.
 - Assign the item to Sam at saadad@microsoft.com.

WWL content issues

1. Go to [WWL Open Issues Query](#).
2. Tag PDETS@Microsoft.com and AskLearning@Microsoft.com(AskLearning@Microsoft.com) in the discussion section of all new tickets.

Engineering work item process

Interactivity

This process is for Interactivity items, specifically where **Area Path** is MicrosoftLearn, Customer Feedback, Engineering, or Interactivity.

To submit a sandbox user appeal

Submit a Customer Feedback work item with **Report an Issue** as the feedback source and add a description

about users with blocked access to sandbox. Criteria for action is:

- Keywords in the description might include: sandbox, can't activate sandbox, appeal.
 - The user's email address is provided.
1. Change the status of the customer feedback work item into **In Review**.
 2. Go to <https://sitehelp.microsoft.com/new> to create a ticket.
 3. Select **MS Learn** as the Service and select **Sandbox user appeal review** in the Service Category.
 4. In **What is the request or issue**, write, "{user email address} is blocked from accessing sandbox."
 5. In **What tags need to be added for the work item**, leave it as **MS Learn**.
 6. In **Attach a file, optional**, add any attached files the user added in the reported issue work item.
 7. In **Tell us more**, add any related customer reported issue URL and copy the description in the customer reported issue.
 8. Select **Submit** and a ticket will be created and triaged by MS Learn SRE team. You can follow that ticket to see the status of the ticket.
 9. When the ticket is closed, close the customer feedback ticket as well.

To review known bugs

1. Check this query to see all known issues about sandbox: https://ceapex.visualstudio.com/Engineering/_queries/query/888dd5e9-6fcc-4723-bf79-3bc553a56d2f
2. If a user-reported issue is a known bug but in closed status, change the bug status to **Active** and @ mention `lyn@microsoft.com` in the comment.
3. If a user reported issue is a known bug but in active status:
 - Leave this bug as active.
 - Add this customer feedback as a comment in the bug.
 - Add the bug as a comment in the customer feedback work item.
 - Close the customer feedback work item.

All Engineering Work Items

Use this process for all engineering work items, including Experience, Profile, Gamification, and Interactivity:

1. Check the **Engineering Need to Triage (MAQ)** query.
2. Sort work items by **Created Date**.
3. Ensure that the work item is sorted into the correct area path.
4. If the work item is a feature request, then add the **Feature Request** tag and ensure the `\Experience Area` is included. Set **State** to **Blocked**.
5. If no action is required, set **State** to **Closed**.

Use this process to close duplicate work items.

1. Check the URL or UID in the Customer Feedback work item under **Troubleshooting**.
2. Check the description in the Customer Feedback work item.
3. If the URL (or UID) is the same as other work items, and customers are talking about the same issue in the description, close all duplicated work items, link closed work items as related work item in the open

work item. Use # mention in the description.

4. If it's an Interactivity issue and more than 10 users reported the same issue in three days, create a bug in the engineering Azure DevOps organization backlog. Assign the PM and link the open Customer Feedback work item in the Bug work item.
5. If the issue is reproducible, list all relevant reproduction steps and screenshots. Tag the PM responsible for the specific area path in the **Discussion** field. Ensure that the PM is included in the **Assigned To** field. Set **State** to **New**.
6. If the issue isn't reproducible but might still possibly be a bug (support agent is unsure), leave **State** set to **New**. Ensure the relevant PM is listed in **Assigned To** and tag the PM in the discussion field with the message, "Not able to reproduce".
7. If the issue isn't reproducible and likely not a bug, set **State** to **Closed**.
8. After you've viewed/handled ticket, set **Triage Status** to **Approved**.

Promoting or using Microsoft Learn at an event

1/14/2022 • 2 minutes to read

Are you promoting or using Microsoft Learn at a conference, workshop, or other event? If so, familiarize yourself with the following key resources.

Internal resources

- [Microsoft Learn frequently asked questions](#)
- [Triage and support issues](#), like registration, sign in, or sandbox use.

Public-facing resources

- [Microsoft Learn help](#)
- [Troubleshooting known issues](#)
- [Reporting an issue](#)

Editing Microsoft Learn content

1/14/2022 • 7 minutes to read

We consider technical accuracy to be a primary goal of every module we publish in Microsoft Learn. However, sometimes things change, new approaches are developed, or we miss something that causes an issue in the content. The most critical thing to be mindful of is **factual errors**. This can include incorrect statements, missing steps in an exercise, or a misuse of the technology when another product would be a better approach. Some of the questions we ask ourselves as we build content include:

- Is the story or scenario being used accurate for the targeted role and technology? We don't want contrived scenarios or to be promoting the technology for something it wasn't intended for.
- Is the exercise complete? Watch out particularly for missing steps - it's best if you actually *try* to do the exercise step by step without adding any details on your own. Does it produce the expected results?
- Is the technology or service being promoted the current approach to solving the problems being presented in the module?
- Are there known changes to the technology or service that this module doesn't account for?
- Check the exercises and any example code. Does it follow best practices for the given platform? Is it concise and easily understandable?
- Do any questions asked in a Knowledge Check make sense? Can the student answer them based on the information we've told them in the units leading up to the check?

We also want to make sure that any code or project we present is **legally usable** and that students can take it and include in their own projects. If you see the following cases, make sure to note them if you feel we might be violating a license.

- Does any presented code use packages that prohibit use in a commercial product?
- Does the code rely on external data, if so do we have permission to use it?

What to do if you find an error

If you have encountered a problem with content published in Microsoft Learn, there are several ways to report and/or fix the issue, listed in order of priority:

1. Report the issue to the author/owner of the module.
2. Submit feedback on the module.
3. Contribute a change with a GitHub pull request (PR).

Each of these reporting methods are described in greater detail below:

Report the issue to the author/owner of the module

If there are significant changes you think need to be made or you are uncertain whether what you are seeing is correct or not, the best approach is to email the owner of the module and report the problem.

TIP

Because multiple teams contribute to MS Learn, it's always best to start with this approach.

To identify the owner of a module:

1. Navigate to the page (unit) where you noted the issue in your browser.

2. Use the **View Source** context menu item in the browser to look at the HTML source code for the page.
3. Scroll to the top of the source and look for the `<head>` tag which has metadata about the page.
4. Inside that section, look for a `<meta>` tag with the name `author` and `ms.author`. For example:

```
<meta name="author" content="markjulmar" />
<meta name="ms.author" content="smmark" />
```

TAG	DESCRIPTION
<code>author</code>	This is the GitHub user id of the author/owner. In the above example, this is markjulmar .
<code>ms.author</code>	This is the Microsoft alias of the author/owner. In the above example, this is smmark .

Email the Microsoft alias (`ms.author`) and send the following information:

1. The URL of the page where you noticed the issue.
2. A brief description of what you think the problem is, including steps (if needed) to reproduce the issue.
3. If you have an idea of how to solve the issue, please include a brief description of that as well.
4. If you are willing and able to provide the corrections, please let them know.

It may take a few days for the author to respond.

Submit feedback on the module

If there is no author, or the problem is minor, you can submit feedback on the module using the public reporting mechanism. This is done through the **Report an issue** link at the bottom of the module page. Alternatively, you can go to the [Feedback page](#) and report an issue there.

This approach is best if you notice something *missing* that isn't essential, but would be useful to mention, or you expected to see something covered that isn't present.

Another helpful piece of feedback would be "What would make this module better?" As you read through the module, what would help you learn it better? Is there a spot where we are missing an opportunity to engage you? Is there a way the writing could be improved, would a video or image help explain the topic? Note these things as possible improvements in an issue against the module.

Contribute a change with a GitHub pull request (PR)

If it's a minor fix, or you know exactly what needs to be done, you can make the change yourself and submit a pull request to merge it into the content.

The source for all docs and MS Learn content is stored in [GitHub](#). If you are unfamiliar with that environment, there are some basics described [here](#).

There are several steps to submitting changes:

1. Identify the GitHub repository where the content is.
2. Fork and clone the repo to your own local instance.
3. Edit the content and check it in your local instance.
4. Submit a pull request (PR) back to the master GitHub repository.

Identify the GitHub repo

The first step is to identify the specific GitHub repo the module is contained in.

1. Navigate to the page containing the error in your browser.
2. Use the **View Source** context menu item in the browser to look at the HTML source code for the page.
3. Scroll to the top of the source and look for the `<head>` tag which has metadata about the page.
4. Inside that section, look for a `<meta>` tag with the name `original_ref_skeleton_git_url`. It will look something like:

```
<meta name="original_ref_skeleton_git_url"
      content="https://github.com/MicrosoftDocs/learn-pr/blob/live/learn-pr/
                data-ai-cert/secure-azure-storage-account/1-introduction.yml" />
```

The `content` value provides a URL that has the necessary pieces to find the source for the page. You should be able to point your browser at that URL to see the YAML file that describes the specific page of content you are looking at on Microsoft Learn.

IMPORTANT

If you get a 404 error, you either need to sign into GitHub, or you don't have access to this repository. Learn uses *private* repositories, so you'll need to request access to the <https://github.com/MicrosoftDocs> organization by [linking your GitHub user id with your Microsoft account](#). This is one-time step that needs to be performed.

The URL will have the following elements:

```
https://github.com/
  {organization}/
  {repository}/
  blob/
  {branch}/
  {root-folder}/
  {content-folder}/
  {module}/
  {unit}.yml
```

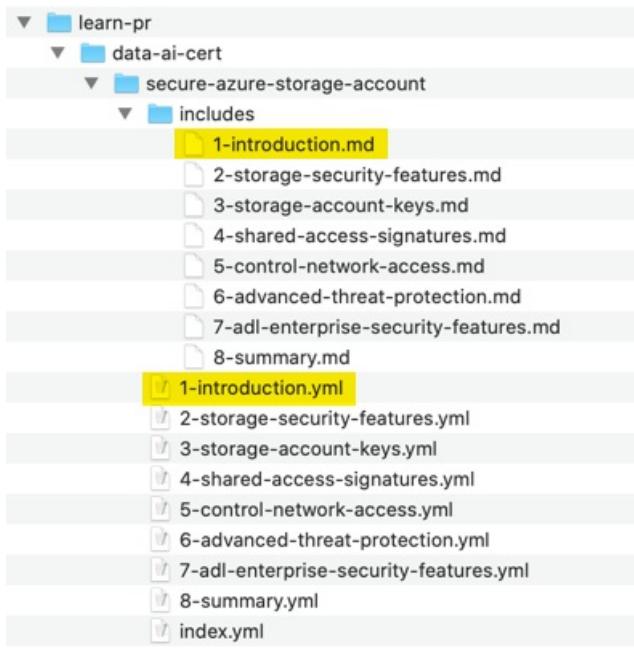
In the example

<https://github.com/MicrosoftDocs/learn-pr/blob/live/learn-pr/data-ai-cert/secure-azure-storage-account/1-introduction.yml>

- `MicrosoftDocs` is the organization, this will always be the same value.
- `learn-pr` is the repository. This can change.
- `live` is the branch, this will always be the same value for live content.
- `learn-pr` is the root folder. In the specified repository, this will be the root folder.
- `data-ai-cert` is the content folder under the root.
- `secure-azure-storage-account` is the module folder.
- `1-introduction-yml` is the YAML file representing the unit page.

Content organization

Each page in a module is built from two source files: a YAML file that describes the page (that's what the above GitHub URL identifies), and a Markdown file that provides the actual content. Most often you'll want to change the Markdown file. It should have the same filename as the YAML file but be in a child folder named `includes`. Here's an example folder structure for the above module we identified:



In this case, the file we'd want to change would be `includes/1-introduction.md`. This contains Markdown that is converted to HTML when it's published. If you aren't familiar with Markdown, we recommend reading the docs guide on [writing and editing Markdown in docs](#) before you make any changes.

Here are few other resources that describe the files that comprise a Learn module.

- [Description of the index.yml file that defines a module](#)
- [Description of the unit YAML file](#)
- [Writing Learn content](#)

Once you've identified the and understand what you plan to change, [follow our step-by-step guide on forking the repo and submitting a PR](#). When you submit the pull request, you'll get an emailed report from the publishing system - this will go to the email connected to your GitHub account. Check to make sure there are no warnings or errors and use the provided review links to make sure the content looks the way you expect.

Finally, tag the author (use the instructions above to identify the author) - you can type a comment and tag them using the `@user` syntax, or use the **Reviewers** section at the top/right of the pull request page to identify a reviewer. It's also helpful to email the author and let them know you submitted the PR (give them the URL) and why you made the change.

If you have questions, feel free to email [Mark Smith](#) or [Adrian Stevens](#).

Retiring published content

1/14/2022 • 4 minutes to read

There are a few reasons why a module may need to be retired. Some of these include:

- Content is out of date with product features and need to be extensively reworked or updated.
- Content doesn't meet editorial or functional standards.
- Security or engineering issues create a business need to pull down the module.

Once it is determined that a module or learning path needs to be retired, there are a few steps that should be taken. While most of these come from the Learn repo admin team, some of these may require people from other teams to resolve.

IMPORTANT

Confirm and talk to your content manager before removing any published module/learning path.

Implications of removing content

When you remove a published module or learning path from Microsoft Learn, make sure that you inform all the concerned teams involved.

The following steps should be taken by the Learn point person when removing the content to confirm that we are not broken elsewhere on the live site.

- Contact engineering to verify that the content you wish to retire isn't showing up on any of the hub/landing pages for the site. Note that these instances are hard-coded by engineering and may not properly redirect, causing 404s for users. Send an email or Teams message with the UID(s) you are planning to retire and the UID(s) that will replace them to [Daniel Stafford](#) to ensure any curated content links are updated.
- Contact localization to make sure that they're aware of the changes, as removing any module from a learning path can cause their tooling to break.

Removing a module

Removing a module isn't just one step of removing the module files from the repo. You need to take a few other actions to ensure that there are no broken links due to removal and the build works fine.

When removing a module, follow these steps in the content repo:

1. Before proceeding with any edits, verify you have completed the above [Implications of removing content steps](#).
2. Remove the entire module folder from the master or release branch.
3. If the badge for the retiring module badge should continue to display on user profile, move the `badge` achievement from the module's `index.yml` file to an entry in `achievements.yml` with the appropriate `title`, `summary`, and `iconUrl` fields that were implied from the module achievement previously.
4. Follow the instructions in the [Redirections](#) article to implement the redirections for the removed content.
5. Comment the module out of any learning paths. The build will fail with a missing UID for any paths that aren't updated with the removal.

Removing a module has some implications within Azure DevOps as well:

1. Close any remaining open bugs or customer feedback for the retired module as obsolete. These can typically be found by querying for the UID containing the module UID root.
2. On the module work item in the Learn Azure DevOps project, set the **State** to **Removed**. (To help finding the live module work item, you can [find the retiring module's UID in this list of all live Learn module work items](#).)
3. If the module is part of any learning paths, remove the child relationship from the parent learning path to the module work item. These can be found, and removed, in the module work item's **Related Work** section by selecting the **X** icon displayed when hovering over the relationship with the parent learning path.

Removing a learning path

When removing a learning path, follow the steps listed below:

1. Before proceeding with edits, verify you have completed the above [Implications of removing content steps](#).
2. Remove the path index file from the master or release branch.
3. If the trophy for the retiring learning path should continue to display on user profile, move the `trophy` achievement from the learning path's `index.yml` file to an entry in `achievements.yml` with the appropriate `title`, `summary`, and `iconUrl` fields that were implied from the learning path achievement previously.
4. Follow the instructions in the [Redirections](#) article to implement the redirections for the learning path.
5. If you also need to remove all the modules in the learning path, follow the instruction to remove a module.

Going live

After the changes above have been made in a pull request against the appropriate branch, the changes may need to be manually merged into master for it to appear. For emergency changes, this will need to be done rather than waiting for the regularly scheduled morning and afternoon weekday merges.

Post publish considerations

Once the changes have been published, two additional processes must complete before a module or learning path is completely delisted.

- The search results on Learn are handled by an indexer service that runs periodically. It will need to do the next run before the retired content stops appearing in Learn search results.
- The suggestions on not found pages (HTTP status code 404) are pulled from a Docs-wide indexer service that runs periodically. It will need to do the next run before the retired content stops appearing on any 404 page recommendations.

Implement redirection for retired Microsoft Learn content

1/14/2022 • 4 minutes to read

When you decide to remove or retire a published Microsoft Learn module or learning path, create a redirection to send users to useful areas that cover similar subject matter. The preferred redirection locations are:

- Redirect the user to another module that covers similar tasks.
- If one doesn't exist, link to an appropriate quickstart/tutorial on the product Docs.
- If there are no tutorials, link to a specific landing page on product Docs.
- If no appropriate landing page exists, then link to an individual article within product Docs.

For more info on the redirection system and its use elsewhere, see [How to redirect obsolete articles](#) in the general [Contributor Guide](#).

When to redirect

Redirections aren't only required when you retire or delete a published file. Redirections are also required when you rename a file or folder that has published content. The file or folder name is part of the URL for a landing page or unit. Any changes to the file or folder names, without proper redirections, will lead to broken URL links. For details on how the build process constructs Learn URLs, see the [Build Learn URLs](#) article.

Apply redirections when you rename or remove:

- **Module folders** - If you rename or remove a module folder, you need to apply redirections for each YAML file within that folder. The module folder name is the part of the URL for each YAML file within the module folder. If you don't apply a redirection, it will result in broken links for all YAML files in the folder (including the `index.yml`, which forms the landing page for the module). For the full details of retiring a module, see the [Removing a module](#) section of the [Retiring Microsoft Learn content](#) docs.
- **Learning path folders** - If you change the learning path folder name only, without any changes to the actual learning path `index.yml`, the redirections are still required for the `index.yml`. The URL for this page will change because of the change of the name of the folder.
- **YAML files** - Renaming or deleting any YAML file will require redirection for that specific file.

Don't redirect includes files

If there are changes to the file name in the `includes` folder, that specifically has the content markdown, there's no need to apply redirections. However, you'll still need to update the file name reference in the respective unit YAML file.

Redirection file

All Learn repos should contain the `.openpublishing.redirection.json` file in the first-level folder. See [Microsoft Learn repo structure](#) for details on how we organize a Microsoft Learn repo. However, if the file doesn't exist, follow these steps:

1. Copy the [.openpublishing.redirection.json file](#) and paste it in the [first-level folder](#) in your repo.
2. Remove all exiting redirections from the copied file.
3. Add your redirections as required.

If you need help with the process, contact [Ashley Johnson](#).

Module redirections

When applying redirections for a module, you must redirect at the `index.yml` and `unit.yml` levels for both the whole module and individual unit pages. Do so by substituting `modulename.md` in the `openpublishing.redirection.json` file. Check out the [redirects in the learn repo](#) for good examples.

NOTE

Make sure to remove any locale, like `en-us`, from the destination URL.

Redirect modules and units

This section describes how to redirect modules and units.

Implement module redirection

Add this code block to redirect each module:

```
{  
  "source_path": "learn-pr/azure/tour-azure-services-and-features/index.md",  
  "redirect_url": "https://docs.microsoft.com/learn/modules/tour-azure-portal/",  
  "redirect_document_id": false  
}
```

While the `index.md` file never physically existed in the Learn repo, it's the target that we use for modules.

Implement unit redirection

Add this code block to redirect each unit:

```
{  
  "source_path": "learn-pr/databricks/understand-the-sql-dw-connector-with-azure-databricks/3-create-user-assign-user-class.md",  
  "redirect_url": "https://docs.microsoft.com/learn/modules/understand-the-sql-dw-connector-with-azure-databricks/3-create-user-assign-resource-class",  
  "redirect_document_id": false  
}
```

Explanation of redirection metadata

FIELD	DESCRIPTION
<code>source_path</code>	The existing repo path to the module's index yaml file or unit yaml files. It's <i>not</i> the old URL path.
<code>redirect_url</code>	The new url. Remove any reference to locale. For example, <code>/en-us/</code> .
<code>redirect_document_id</code>	Should you preserve the original document ID? A value of <code>true</code> allows the original page views and rankings to carry forward.

NOTE

Though the source redirection URL points to index and unit yaml files, use the .md extension here.

If you're redirecting to *mostly* identical content, you may want to preserve the original analytics data through to the new location. By switching `redirect_document_id` to `true`, the original page data will carry through. If you're retiring a unit or module, set `redirect_document_id` to `false`.

Learning path redirection

You may have to remove an entire learning path or rename the folder that contains the learning path index yaml file. If so, apply appropriate redirections in the global redirection file.

Implementation

Add this code block to redirect each learning path:

```
{  
    "source_path": "learn-pr(paths/modern-data-warehouse/index.md",  
    "redirect_url": "https://docs.microsoft.com/learn/paths/data-engineering-with-databricks/",  
    "redirect_document_id": false  
},
```

While the `index.md` file never physically existed in the Learn repo, it's the target that we use for modules.

Explanation of json metadata

FIELD	DESCRIPTION
<code>source_path</code>	The existing repo path for the learning path index yaml (referred to as index.md in the redirection file).
<code>redirect_url</code>	The new url. Remove any reference to locale. For example, /en-us/.
<code>redirect_document_id</code>	Should you preserve the original document ID? A value of <code>true</code> allows the original page views and rankings to carry forward.

If you're redirecting to *mostly* identical content, you may want to preserve the original analytics data through to the new location. By switching `redirect_document_id` to `true`, the original page data will carry through. If you're retiring a unit or module, set `redirect_document_id` to `false`.

Testing new redirects in a pull request

When you submit a pull request containing new redirects, it's best to test them out before the `#sign-off` comment. If the review goes as expected, the process will reject failing redirects from merging.

To test your redirect, create a URL to the page in your redirect's `source_path` and append the query string for your pull request's build.

For example:

- The redirect is for the `source_path` of `learn-pr/azure/intro-to-mobile-app-service/1-introduction.md`.
- It's submitted in pull request #6643.

The PR number is found next to the PR's title.

The URL to test would be

<https://review.docs.microsoft.com/learn/modules/intro-to-mobile-app-service/1-introduction?branch=pr-en-us-6443>

View Learn content portfolio and roadmap

1/14/2022 • 2 minutes to read

We track Learn content development in our Microsoft Learn Azure DevOps project.

Content developers are responsible for creating and updating their work items in the project. For information, see [Track Learn content development in Azure DevOps](#).

View content portfolio and roadmap queries

We have multiple shared queries to help people see the “slice” of the Learn content roadmap that’s most applicable for them. You can run queries against your own content. For example, we’re using AzDO as our source of truth and keeping it updated weekly.

Notable query links and descriptions are below.

- [Learn Content Portfolio - Published \(Live\)](#)
- [Learn Content Portfolio - Published \(Live\) and Deprecated](#)
- [Learn Content Roadmap - Current Quarter](#)
- [Learn Content Roadmap - Current Quarter Content with Labs](#)
- [*Learn Content Roadmap - By Content Team \(Note: Running this query isn't the same as viewing content by product, as there may be multiple content teams producing content for the same product.\)*](#)

To view or bookmark the queries, they’re all stored in our Azure DevOps project, in the [Roadmap Pivot shared query folder](#).

IMPORTANT

If you don’t have access when you click the links above, please join the applicable security group for your role via IDWeb. It may take up to an hour to populate and give you access.

- [MSFT Learn - FTE Contributors](#)
- [MSFT Learn - Vendor Contributors](#)

Notes about roadmap

- The information provided is only as up-to-date as to what the contributing content teams have input. The Learn team works with the content teams. They ensure that the data is kept as fresh as possible, and the content should continue to improve as our processes mature.
- We don’t currently have a way of tracking in AzDO when we know we have a content onboarding project. We’re likely to do this quarter (for example, AI School), but don’t know the exact modules or counts yet. We’re working on how to represent this here.

Power BI reports and dashboards

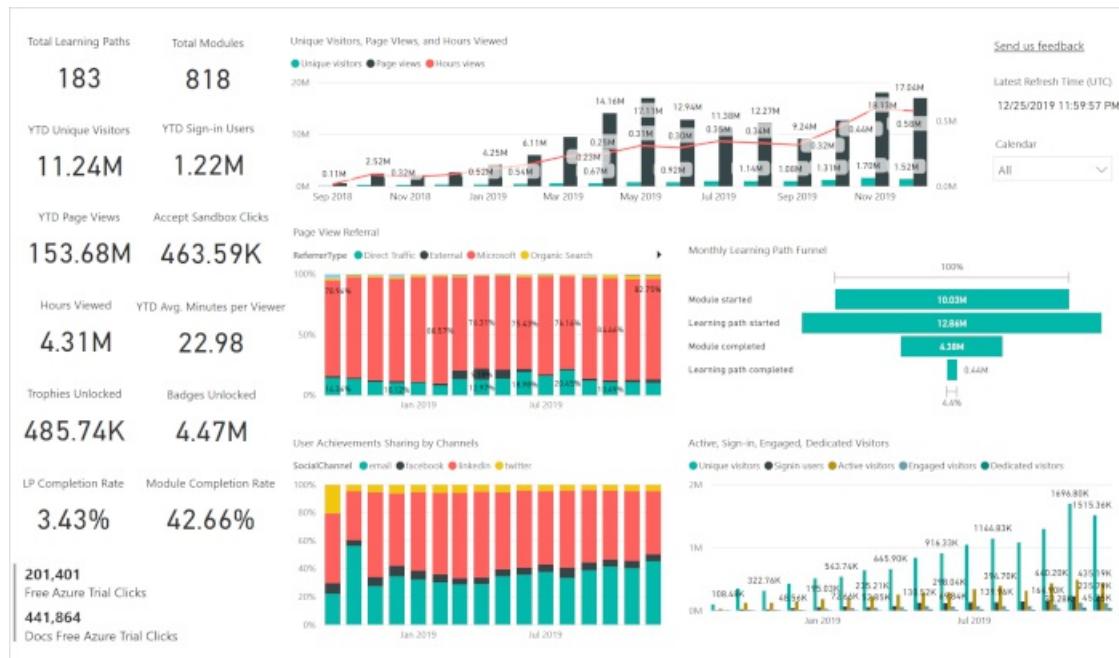
1/14/2022 • 2 minutes to read

Several [Power BI reports and dashboards](#) allow you to measure your content's performance. The data on each report can be filtered based on different criteria such as products and roles. You can view the data for a single calendar month. You can also compare the trend by populating data for multiple calendar months.

Available reports

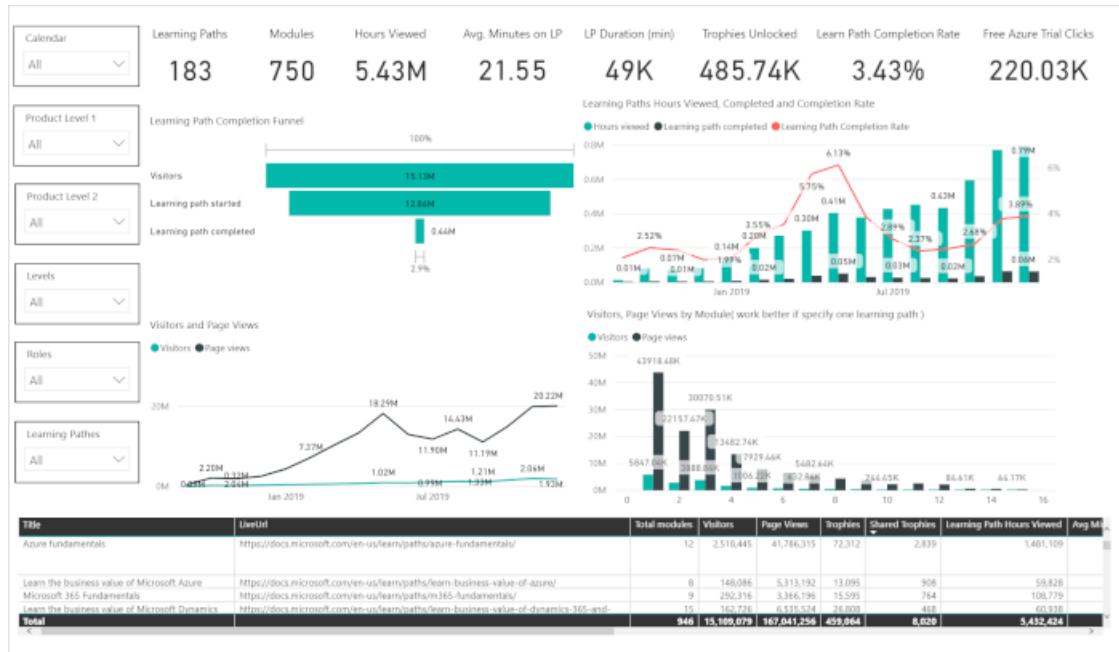
The following reports are available:

- **Top level summary** - Gives a high-level overview of all Learn content, including the total number of published learning paths, modules, and unlocked trophies and badges.

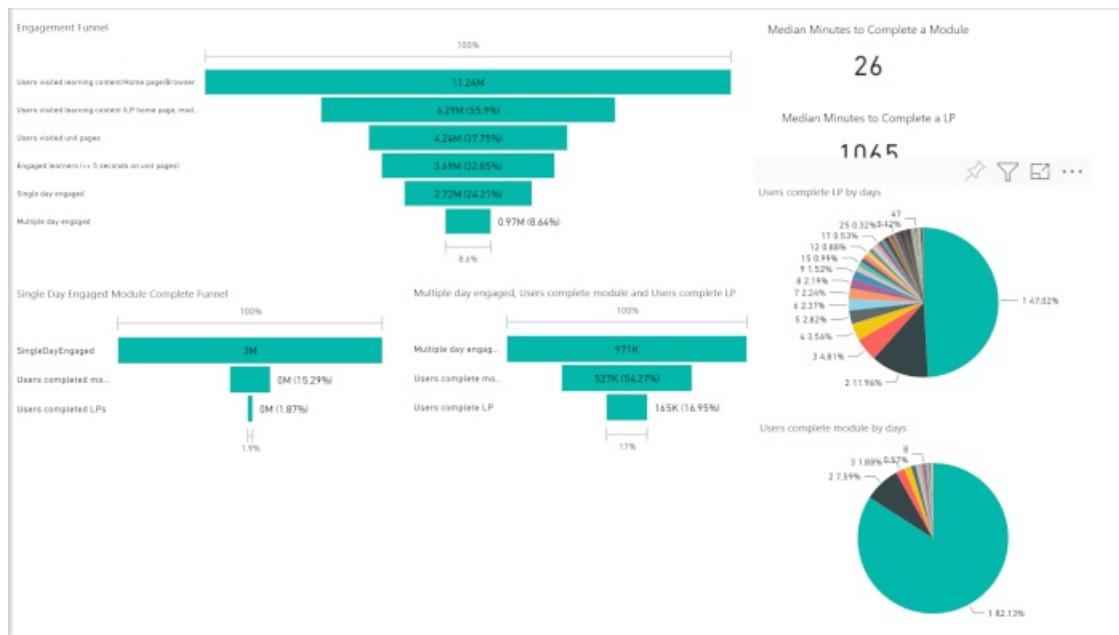


- **Learning path, module, and unit** - These reports show data on all published learning paths, modules, and units. You can choose to view specific data based on:

- Products
- Levels
- Roles
- Learning path
- Module
- Unit names



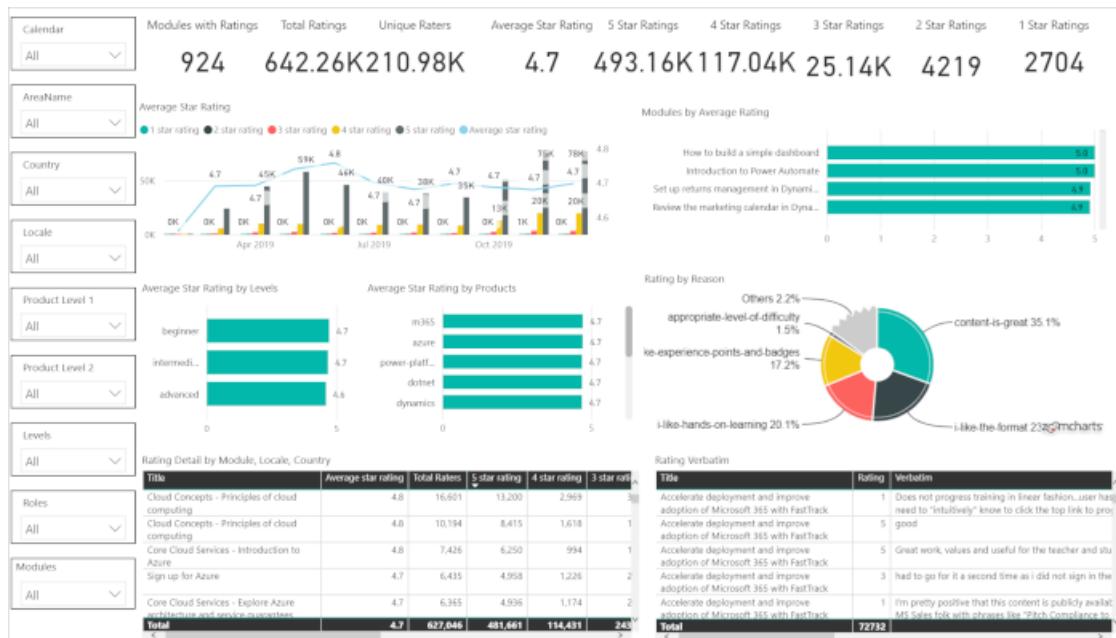
- **Engagement** - This report shows data on learner engagement with MS Learn. It gives a view of the MS Learn engagement funnel. The funnel tracks learners' movements from the Learn home page to learning path to productive engagement with a single unit. You can also view data on completed learning paths and modules by number of days.



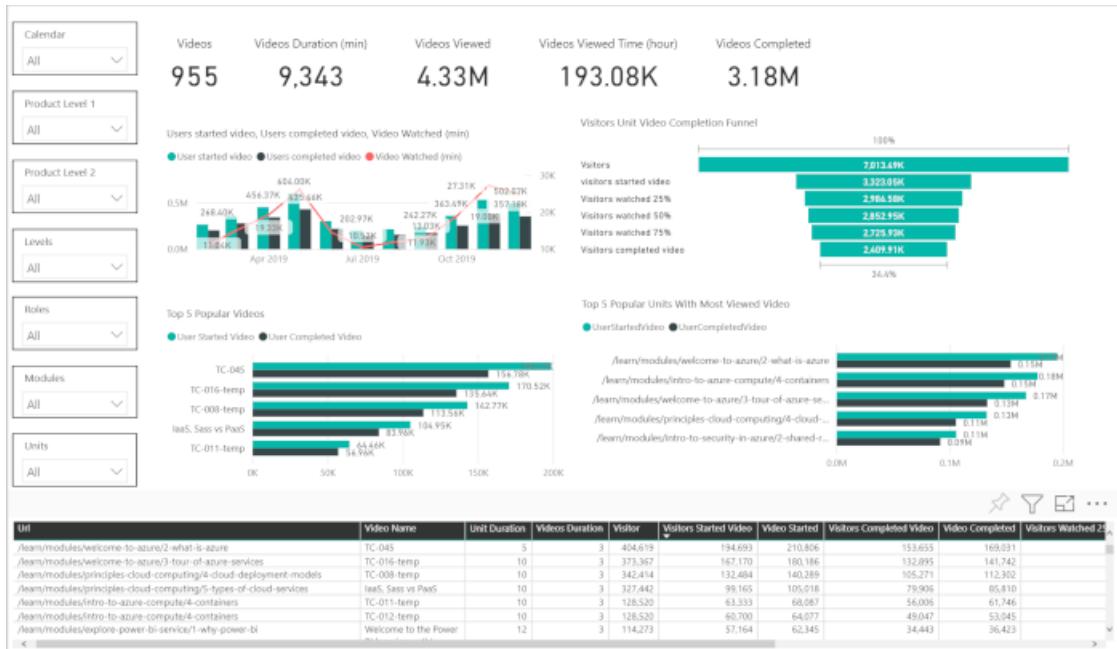
- **Achievements** - View the data on the total number of trophies and badges unlocked by the learners for each product and role. View data for a single calendar month or compare trends for multiple months.



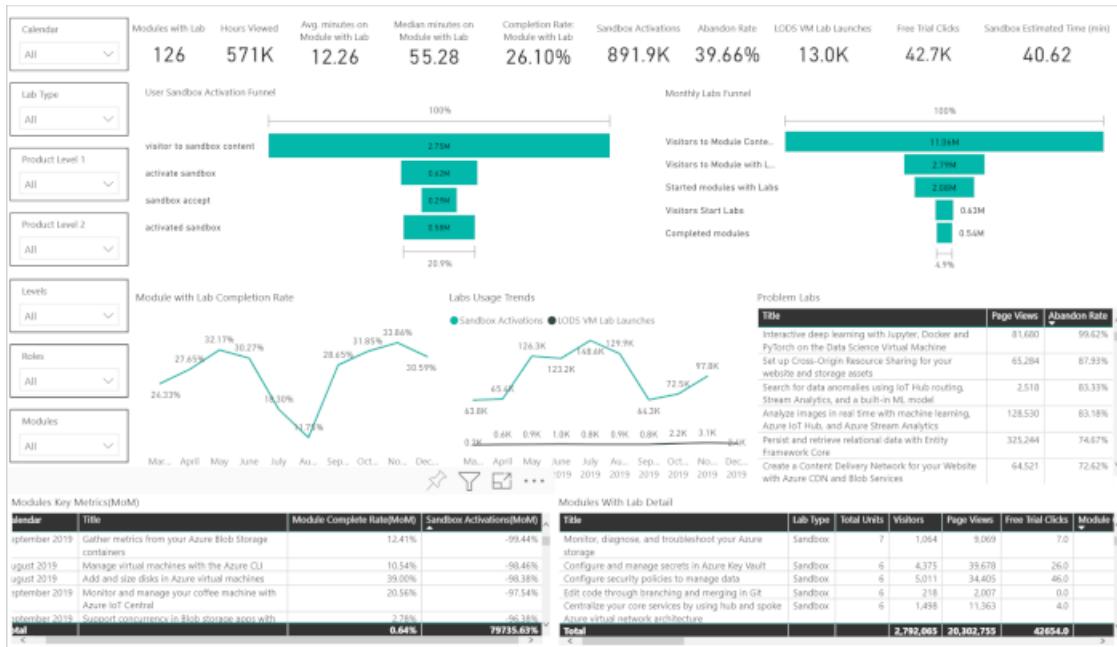
- Ratings** - This report shows data on star ratings and verbatim on Learn modules. Apart from product, level, roles and module names, you can also filter this report for a specific geographical area, country, or locale.



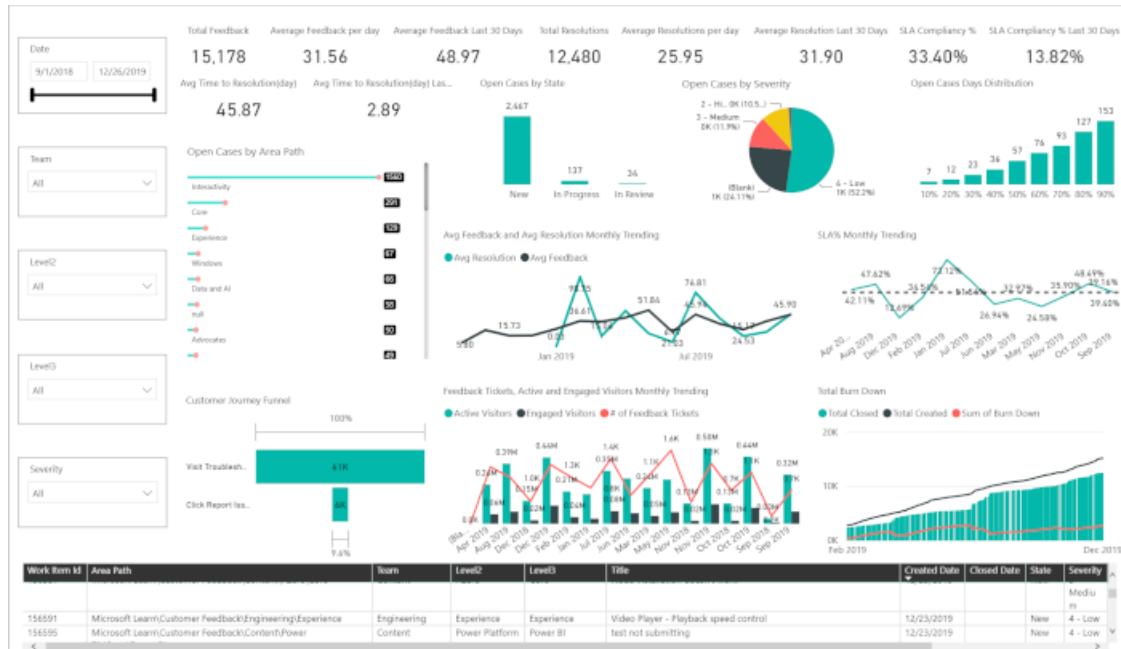
- Video** - Measure the effectiveness of videos in your Learn content by looking at the numbers of visits to the videos and completion rates. View the stand-alone data for your content or compare it with other Learn data to know more about trends and user engagement.



- **Learn labs** - View the performance of your Learn sandbox and LOD labs. See the user completion and abandon rates to find labs with high and low performance. Compare the results and apply solutions as required.



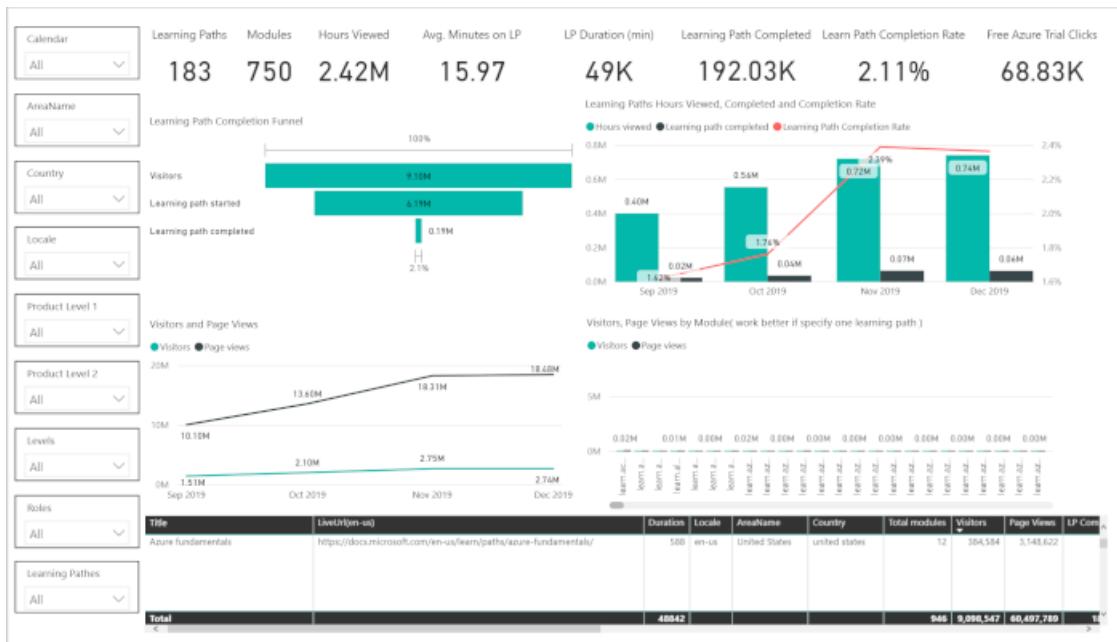
- **Customer feedback** - This report shows data on customer reported issues via the [Reporting an Issue](#) link at the bottom of each unit. You can view the data on issues reported for specific team or area.



- **Area** - This report gives a high-level overview of all published Learn content by specific geographical area and country. If you filter out the data for All/areas and countries, this data is similar to what displays in the **Top level summary** report.



- **Learning path and module by area and locale** - Two reports are available to view data on Learning paths and modules by individual geographical area and locale.



- **By product and role** - View published Learn content data filtered by each product and role.



- **KPI by Country and user role** - The two reports are available that show different KPI metrics for Learn content by country and users (Microsoft employees vs. non-employees).

Metrics

Avg Time/Visitor	Module Completed	Sandbox Activation
Engaged Visitors	Multiple Engaged	Signin Visitors
Learning Path Completed	Pageview	Unique Visitor

All KPI Comparison

As of Date	Metric Name	Current Value	KPI Status	Comparison Value	Sparklines
11/1/2019 12:00:00 AM	Avg Time/Visitor	5,030.40	-12.68 %	5,818.36	
11/1/2019 12:00:00 AM	Engaged Visitors	510,427.00	+15.77 %	480,880.00	
11/1/2019 12:00:00 AM	Learning Path Completed	65,618.00	+77.93 %	36,879.00	
11/1/2019 12:00:00 AM	Module Completed	590,759.00	+50.63 %	392,194.00	
11/1/2019 12:00:00 AM	Multiple Engaged	130,970.00	+26.35 %	103,655.00	
11/1/2019 12:00:00 AM	Pageview	18,111,356.00	+42.77 %	12,685,332.00	
11/1/2019 12:00:00 AM	Sandbox Activation	118,045.00	+60.69 %	73,462.00	
11/1/2019 12:00:00 AM	Signin Visitors	256,457.00	+49.50 %	171,542.00	
11/1/2019 12:00:00 AM	Unique Visitor	1,730,538.00	+31.10 %	1,320,058.00	

AreaName

AreaName	Current Month Value	% Current Month Value	Previous Month Value	% MoM
CorpHQ				
EMEA HQ				
Central and Eastern Europe	18,921	7.38%	9,032	109.49%
Germany	10,620	4.14%	5,630	83.36%
Western Europe	3,362	15.74%	21,091	84.38%
	558	0.22%	333	67.57%
India	22,760	8.87%	13,613	67.19%
France	5,873	2.29%	3,605	62.31%
UK	23,138	9.02%	14,617	58.10%
APAC	13,720	5.15%	9,340	46.90%
Canada	8,285	3.23%	5,760	43.64%
Australia	7,266	2.83%	5,239	38.69%
Japan	5,861	2.29%	4,229	38.59%
Total	256,457	100.00%	171,542	49.50%

Country

Country	Current Month Value	% Current Month Value	Previous Month Value	% MoM
united states	23,649	24.10%	14,532	33.16%
united kingdom	23,449	8.59%	14,532	58.39%
india	22,760	8.86%	13,613	67.19%
germany	10,620	4.14%	5,630	85.36%
netherlands	8,936	3.47%	5,247	69.74%
canada	8,284	3.23%	5,767	43.64%
australia	7,236	2.82%	5,216	38.73%
brazil	6,165	2.40%	5,927	4.02%
france	5,871	2.29%	3,602	62.99%
japan	5,861	2.29%	4,229	38.59%
spain	4,743	1.69%	2,991	58.58%
sweden	4,645	1.81%	2,059	125.59%
austria	3,395	1.32%	717	373.50%
china	3,352	1.31%	2,662	25.92%
singapore	3,319	1.29%	2,232	48.70%
new zealand	3,111	1.14%	2,074	51.14%
Total	256,447	100.00%	171,542	49.50%

Get page-level metrics in your browser: Microsoft Docs Metrics

1/14/2022 • 13 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, channel9.msdn.com, cloudblogs.microsoft.com, code.msdn.microsoft.com, code.visualstudio.com, devblogs.microsoft.com, developer.microsoft.com, docs.azure.cn, dotnet.microsoft.com, gallery.technet.microsoft.com, msdn.microsoft.com, review.docs.microsoft.com/help/ (*Internal Contributor Guide*), social.msdn.microsoft.com, social.technet.microsoft.com, techcommunity.microsoft.com, technet.microsoft.com, techprofile.microsoft.com, www.azure.cn, www.microsoft.com

MDM on Edge is GA (recommended to receive automatic updates), but remains unsigned in beta on Chrome.

Edge: install & login

Microsoft Docs Metrics on Edge is GA (offering auto-updates).

Install the tool:

1. Get the latest extension from [version: 3.1.6](#)

- 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 browser 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 browser 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.

Resources

@microsoft.com | [log out](#) | [View my topics](#) | [Privacy](#)

See also: [How to use this tool](#) | [C&L reports](#) | [Send feedback](#)

Version: 2.4.0 [Upgrade to latest version: 3.0.0](#)

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. Note that permissions may take up to 48 hours to propagate. Thank you for your patience.
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

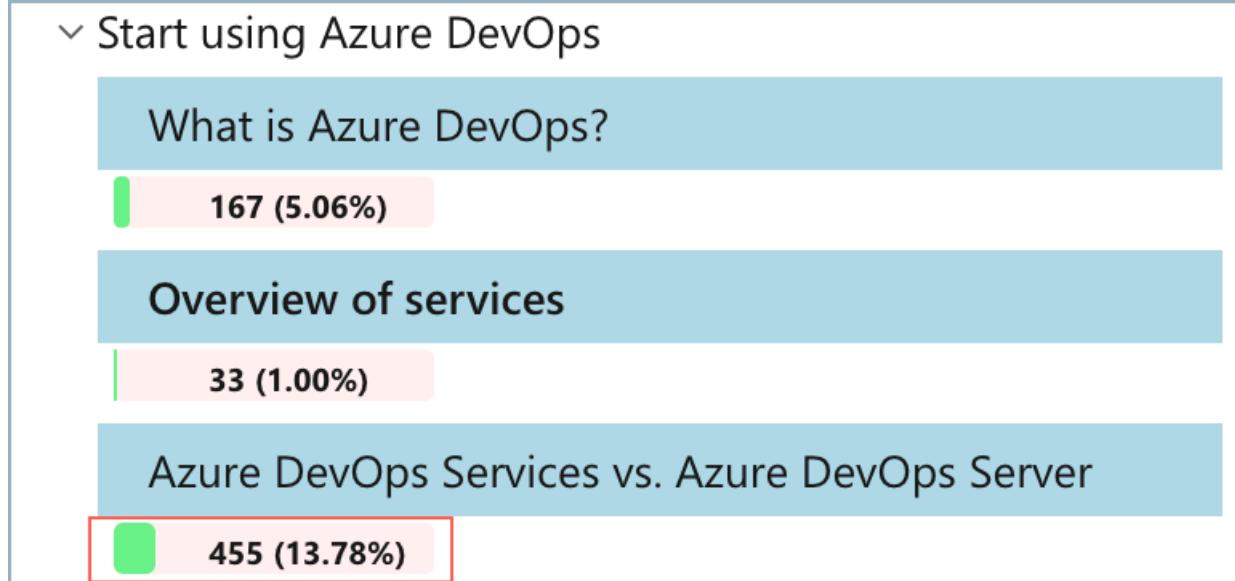
To see what outgoing links were clicked (click-through), select the button **Show link clicks (Clicked PV, CTR)**. When you select this button, all the links clicked during the selected time range appear on the page.

The data appears on the page with this syntax `Clicked PV (CTR %)`:

VALUE	DESCRIPTION

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 repeated links on a page pointing to the same URL, their counts will be differentiated or aggregated where this is not possible. Hovering over a repeated link highlights all instances of the link on the page.
CTR	The percentage of page views that visited the linked URL (click-through rate).

You'll see how many page views a URL got by the visual bar.



You'll also see repeated links highlighted showing their respective click counts to the same URL targets.

Version

Azure DevOps Services

Filter by title

- Get started 47 (1.35%)
- Start using Azure DevOps
 - What is Azure DevOps? 10 (0.29%)
 - Overview of services 235 (6.75%)
 - Azure DevOps hosted vs. on-premises 35 (1.01%)
- > Get started for end users
- > Get started for administrators
- > Key concepts

What is Azure DevOps?

01/22/2021 • 3 minutes to read • 5 comments

Azure DevOps Services | Azure DevOps Server 2020 | Azure DevOps

Azure DevOps provides developer services for support teams to help them build and deploy applications. Azure DevOps supports a community of developers and project managers and contributors together to help organizations to create and improve products at a faster pace using modern development approaches.

You can work in the cloud using Azure DevOps Services or on-premises, or both. This article provides information on the differences between the cloud versus on-premises environments.

Azure DevOps Services and Azure DevOps Server 87 (2.50%)

To see a heat map of clicked PV counts, select **Show heat map**. When you select this button, all the links clicked during the selected time range are colored to give a visual cue to the popularity of a link based on their CTR value. The in-page heat map legend shows the spectrum of values and their relation to colors.

The screenshot shows the Microsoft Docs Metrics interface for the page 'What is Azure DevOps?'. The top navigation bar includes links for Microsoft, Docs, Documentation, Learn, More, Search, and Sign in. The main content area displays the page's title, a summary of its content, and a sidebar with 'Is this page helpful?' options (Yes, No). The right side features a detailed metrics dashboard with sections for Traffic, Engagement, Satisfaction, GitHub Issues, and Referrers.

Traffic

- Unique visitors: 2,513 (-1.99%)
- Page views: 3,482 (-2.16%)

Engagement

Rank percentile	
Bounce rate: 9.48% (0.00)	62.4
Click-through rate: 39.86% (+0.01)	51.8
Visitors Scroll, Copy, Try rate: 78.87% (0.00)	37.5
Dwell % of read time: 15.00% (-0.01)	73.4
Score: 1.09 (+0.01)	47.4

Satisfaction

Helpful rate: 100.00% (0)
Responses: 2 (+1)
Verbatims: 0 (0)
Response rate: 0.08% (0.00)

Github Issues

New issues opened: 0 (0)
Total open issues: 0 (0)

Referrers

Direct: 7.81% (-0.00)
Microsoft: 72.40% (-0.01)
Organic search: 18.50% (+0.02)
External: 1.29% (0.00)
Social: 0.00% (-0.00)

Less performing referrals (Advanced)

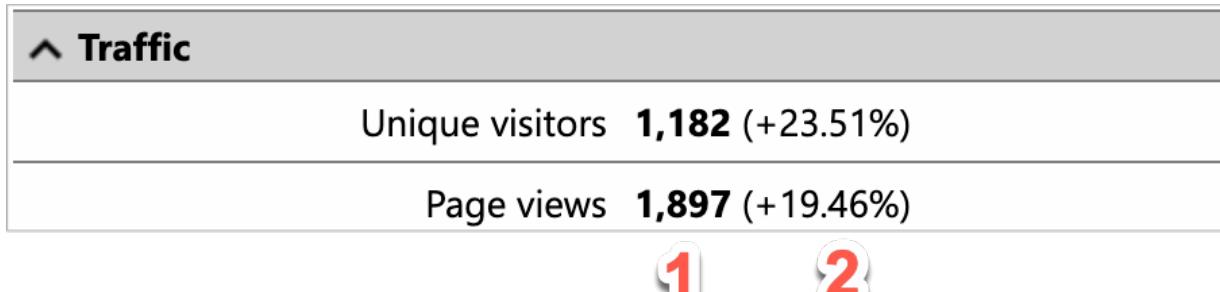
Page views
Organic Search: www.google.com/ 582

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
Dwell % of read time	All	Dwelt on the page for a while	The higher, the better

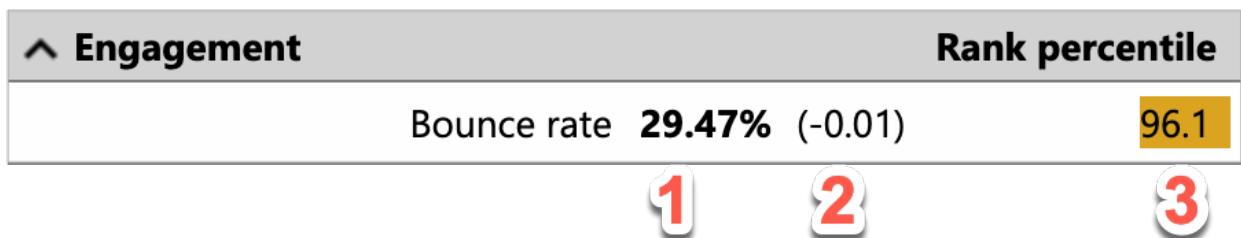
Value	Group	Description	Rank Percentile
Monthly completion rate	Learn	Completed the learning path	The higher, the better
Score	Docs	Roll-up of the engagement metrics into a single number	The higher, the better

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 than 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

3.1.4

Version [3.1.4](#), released on **2021-08-01**, and contains the following major fixes

- Add support for the following Microsoft sites: review.docs.microsoft.com/help/ (*Internal Contributor Guide*) and dotnet.microsoft.com
- Add differentiated counts for repeated links
- Add highlighting for repeated links
- Add heat map for clicked links

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 overlay 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.

Community Kusto queries

1/14/2022 • 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 `cгадатамал` 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(FirstReviewed) 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(if(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](#).

MS Learn Data Warehouse for AAD Users

1/14/2022 • 10 minutes to read

The MS Learn Data Warehouse is a SQL Server based data warehouse that provides authorized users access to the raw data of the MS Learn gamification system for Microsoft Employees only that have given their consent. For more information on how the gamification system works as a whole, see the [Gamification backend service spec](#).

Permissions to see Microsoft Employee Data

We grant permissions to access the Microsoft Employee data via:

1. The **MS Learn MSFT Data Access** Active Directory group, managed via IDWEB. This group was created for the purpose of the need for Worldwide Learning (WWL) team.
2. Granting permissions to access the Microsoft Employee data by allowlisting a set of IP ranges.

Permissions are granted by the [Dev Relations Service Engineering team](#).

Permissions to see Microsoft Partner Data

We grant permissions to access the Microsoft Employee data via:

1. The **PX Analytics Data** Active Directory group managed by [Madhu Ramaswamy](#).

Database information

Once added to the corresponding group, a user can access the database server itself via SQL Server Management Studio. When logging in,

- use **userlearnreport.database.windows.net** as the *Server Name*;
- select **Active Directory - Universal with MFA support** for *Authentication*;
- use your full email address: **{alias}@microsoft.com** for *User name*;
- click on *Options* and connect to the **userlearnreport** database.

The database contains the data feed for all users who have signed with an ADD account AND have consented to share their data during the registration process (for new users) or when they sign in to Microsoft Learn (for existing users) via their Tech Profile.

You can either access Microsoft Employees or MS Partner data depending on which Security Group you joined. For better access the data, you might want to choose one of the views:

1. **Microsoft Employees data:** All prefixed with **msft**.
2. **Microsoft Partner data:** All prefixed with **partner**.

NOTE

The partner.Users view is limited to users with a tenantID that match the list of tenantIDs provided by the Microsoft Partner team.

NOTE

Any user with a tenantID that is neither a Microsoft tenantID nor in the list provided by the Partner team is currently not visible to any reporting users.

Data refresh

- The data is pulled from the production database every four hours.
- Once a week on Saturdays at midnight Pacific Time, the entire data set is cleared and rebuilt. This is in order to be complaint with GDPR regulations by getting rid of users' data that may have been deleted from the source.

About the dates

Dates are `datetimeoffset` which means they have offset from UTC included in the timestamp. See [datetimeoffset](#) for more information

Data Definitions for Learn Data Export

There are multiple collections of data, all of which can be found in SQL Server views for Users, Achievements, Level, XPTotal and Progress. Data can be joined between these collections through the `UserId` field, which represents each user in the system.

In addition, there are lookup tables that provide metadata about the entities define the points needed per level (LevelRule) and the relationships between achievements, modules, learning paths and units, which is known as the hierarchy data.

Users View

The list of users in the system that have registered at techprofile.microsoft.com.

FIELD NAME	DATA TYPE	DEFINITION
UserId	guid	The user's AAD ObjectId.
UserName	string	The user name associated with this UserId, chosen by the user when creating a tech profile. It is unique.
CreatedOn	string	The date the user created a tech profile. It contains no time information and is represented in English short time format.

Achievements View

Achievements are earned based on the user completing certain paths or modules.

FIELD NAME	DATA TYPE	DEFINITION
UserId	guid	The user's AAD ObjectId.
UserName	string	The user name associated with this UserId, chosen by the user when creating a tech profile. It is unique.

FIELD NAME	DATA TYPE	DEFINITION
AwardType	string	An enumeration of possible achievement awards. Valid values are Badge and Trophy .
AwardUid	string	An identifier that defines the award received. It maps to a Uid in the hierarchy data, where additional metadata can be found about the award.
Reason	string	An enumeration of possible reasons the achievement was awarded. Valid values conform to the following convention: {AwardType}.{path or module}.basic.completed .
SourceType	string	An enumeration of possible achievement categories. Valid values are Unit , Module and LearningPath .
sourceUid	string	The source Uid is an identifier that defines the reason for the achievement. It maps to a Uid in the hierarchy data, where additional metadata can be found about the task that was completed.
AwardedOn	dateTimeOffset	The date and time when the achievement was awarded.

Level View

Level is based on the user's total XP. Levels are determined by a rules table. Each time a user attains a certain amount of XP, they level up. To understand the progress of a user over time, examining the level table will reveal the user's pace.

FIELD NAME	DATA TYPE	DEFINITION
UserId	guid	The user's AAD Object Id.
totalPoints	int32	The user's current total points in the MS Learn system.
currentLevel	int32	The user's current level in the MS Learn system. To understand how level correlates to XP, see the <i>LevelRule</i> table.
currentLevelLow	int32	A lookup to a static value that represents the minimum number of points for the user's current level.
currentLevelHigh	int32	A lookup to a static value that represents the maximum number of points for the user's current level.

FIELD NAME	DATA TYPE	DEFINITION
nextLevel	int32	The level the user is at plus one.
currentLevelPointsEarned	int32	The number of points the user has achieved since attaining the current level.
pointsToNextLevel	int32	The number of points required to attain the next level.
DateLevelAchieved	dateTimeOffset	The date and time when the achievement was awarded.

LevelRule Lookup Table

LevelRule is a look up table used to determine how many points are required to level up.

FIELD NAME	DATA TYPE	DEFINITION
Level	int32	The level itself, a numerical value from 1 to 20.
Low	int32	The minimum amount of XP required to attain this level.
High	int32	The maximum amount of XP to still be in this level.

Progress View

Progress is the running list of tasks the user completed to earn points and how many points they earned for completing the task.

FIELD NAME	DATA TYPE	DEFINITION
UserId	guid	The user's AAD ObjectId.
UserName	string	The user name associated with this UserId, chosen by the user when creating a tech profile. It is unique.
Reason	string	An enumeration of possible reasons the points were awarded. Valid values conform to the following convention: points.{path or module or unit}. {basic or bonus}.{completed or past-quiz-at-first-trial} .
SourceType	string	An enumeration of possible tasks types to earn points. Valid values are Unit, Module and LearningPath .

FIELD NAME	DATA TYPE	DEFINITION
SourceUid	string	The sourceUid is an identifier that defines the task that was completed to earn the points. It maps to a Uid in the hierarchy data, where additional metadata can be found about the task that was completed.
XP	int32	A numerical value of points earned for completing the task.
AwardedOn	dateTimeOffset	Date and time when the points were awarded.

XPTotal

XPTotal is a view that sums the users XP.

FIELD NAME	DATA TYPE	DEFINITION
Userid	guid	The user's object id (AAD) or PUID (MSA).
TotalXP	string	A calculation of the user's total XP.

Hierarchy Data Tables

Hierarchy data is the domain encyclopedia highlighting the relationships between Learning paths to Modules to Units to Achievements. Learning Path is the parent of the Microsoft Learn program and each Learning Path is associated with one or more Modules. Each Module contains a collection of Units which needs to be completed in order to complete the Module. Therefore, the logical relationship are as follows:

- One Unit has only one parent Module
- One Module may have several parent Learning Paths (which means Modules are shared across Learning Paths)
- Some Modules do not have any Learning Path parents and are designed to be independent

Achievements are awarded as and when a user finishes a Module or completes a Learning Path. They earn a badge or a trophy depending on their progress of completion.

LearningPath

FIELD NAME	DATA TYPE	DEFINITION
LearningPathUid	string	The unique identifier of the learning path.
title	string	Title of the learning path.
url	string	Relative URL of the learning path.
durationInMinutes	int	The time taken to finish the LP.
points	int	The XP points the user would earn if they finish the learning path.

FIELD NAME	DATA TYPE	DEFINITION
iconUrl	string	The relative URL to the icon for the learning path when seen on MS Learn.

Module

FIELD NAME	DATA TYPE	DEFINITION
moduleUid	string	The unique identifier of the module.
title	string	Title of the module.
url	string	Relative URL of the module.
durationInMinutes	int	The time (in minutes) taken to finish the module.
points	int	The XP points the user would earn if they finish the module.
iconUrl	string	The relative URL to the icon for the module when seen on MS Learn.

Unit

FIELD NAME	DATA TYPE	DEFINITION
unitUid	string	The unique identifier of the unit.
title	string	Title of the unit.
url	string	Relative URL of the unit.
durationInMinutes	int	The time (in minutes) taken to finish the unit.
points	int	The XP points the user would earn if they finish the unit.
moduleUid	string	The parent module to which this unit is related.

LearningPathModule

It describes the relationship between Learning Path and Module.

FIELD NAME	DATA TYPE	DEFINITION
LearningPathUid	string	The unique identifier of a learning path.
ModuleUid	string	The unique identifier of a module.

Achievement

FIELD NAME	DATA TYPE	DEFINITION
depotName	string	The category to which this achievement belongs to. Ex: Docs, CRM, etc.
locale	string	The locale of the achievement.
AchievementUid	string	The unique identifier of the achievement.
title	string	The title of the achievement. If the language is other than English, this title is localized.
summary	string	Summary of the achievement.
iconUrl	string	Relative URL of the icon url which will be displayed on MS Learn after earning the achievement. If the language is other than English, this icon is localized.
type	string	The type of achievement. Examples: badge, trophy .
sourceType	string	The source to which this type is mapped to. For example: progress .
sourceUid	string	The learning id to which this achievement is mapped and earned on completion.

RawJson

The raw Json file where the hierarchy is stored.

FAQ

1. Q: Who are the contact persons and/or what is the team alias to reach out to in case of lower priority issues/questions/clarifications?

A: The [Dev Relations Service Engineering team](#) is the best contact for issues or questions.

2. Q: Is there possibility of future breaking changes to any tables and views in the **userlearnreport** DB? We are pulling data from some tables/views and can absorb schema changes like addition of columns. But in case you are introducing data type changes or dropping tables/columns, we would like to be informed in advance so that we adequate reaction time.

A: We do not foresee us making breaking changes unless there are changes to the way MS Learn models content. Those will likely be quite large and will require more communication than just a schema change. We will use the **PX IDC Analytics Dev Team** (px_idc_analytics_dev@microsoft.com) team alias to communicate those type of changes. Please make sure you join to be up to date on schema changes.

3. Q: Can you please let our team know in advance in case you plan any changes to the grants for this group on the DB objects?

A: We will use the **PX IDC Analytics Dev Team** (px_idc_analytics_dev@microsoft.com) team alias

to contact you in case of this sort of change.

4. Q: Is there an 'ideal' time for pulling MS Learn data?

A: It's probably best to schedule your pull at a time other than the run times of our **Azure Data Factory (ADF)**:

OUR SCHEDULED RUN TIME	CURRENT MAX DURATION (MINUTES)	A GOOD START TIME TO SCHEDULE YOUR IMPORT
3:00 AM	90	4:30 AM
7:00 AM	90	8:30 AM
11:00 AM	90	12:30 PM
3:00 PM	90	4:30 PM
7:00 PM	90	8:30 PM
11:00 PM	90	12:30 AM

5. Q: We are not applying any NGP delete signals to this data since our understanding is that MS Learn periodically refreshes the data and the NGP deletes have already been applied upstream. Can you please confirm if this understanding is true – that the periodic MS Learn data refresh at your end ensures that deleted AAD users are automatically removed from the **userlearnreport** DB?

A: That is correct; every Saturday we rebuild the database from scratch which handles all deletes which we receive from our in-product experience and the central PCF.

IMPORTANT

You must do at least a monthly complete refresh of your copy with these data, or you may be violating GDPR requirements.

UserTesting

1/14/2022 • 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

Create a new Microsoft Learn GitHub repository

1/14/2022 • 6 minutes to read

Repos for Microsoft Learn content are created using the [Open Publishing Service portal](#), and are then heavily modified to make work with Learn features.

WARNING

Do not create new repos without approval from the Microsoft Learn team. Requests to onboard can be submitted here:

https://forms.office.com/Pages/ResponsePage.aspx?id=v4j5cvGGr0GRqy180BHBxenY_g_4FApVhlj1Jzq4RUQIFTT1FWNEo2N0JHWUkyU0RMS0pDVkRMWi4u

IMPORTANT

Before creating a new repo:

1. Verify that there is not already a repo for the content that you want to create. Review the [Microsoft Learn repo](#) article for the full list of existing repos.
2. Verify with the content team and localization team what the 'strategy' is for the new repo. Can we create one for the content development team, or does it need to be split to handle multiple localization teams or products, etc.?

This articles describes the procedure to create repos where the actual Learn content resides, such as modules and learning path. All the sample code used in the Learn content must be stored in individual code repos. To learn how to create code repos, refer to the [Learn code sample repos](#) article.

Create the repo in OPS

1. Go to [OPS](#) and select New docset.

Step 1 of 3: Select or create Git repository

X

Account/Organization [?](#)

MicrosoftDocs ▾

Repo Name [?](#)

learn-test-pr ▾

Private Repository [?](#)

[Show Locale Settings](#)

⚠ The following settings apply to every docset in this repo.

Notification Subscribers [?](#)



E.g. Alice@ms.com;Bob@ms.com

Build/publish notifications will be sent to these email addresses.

- Preview Build Pull Requests

[Hide More Repo Settings](#)

Next

Cancel

Choose the following options:

- **Account/Organization:** MicrosoftDocs
- **Repo Name:** learn-product-pr
- Select the checkbox for **Private Repository**.
- Expand **More repo settings** and select **Preview Build Pull Requests**.

2. After moving to the **Next** screen, define the following **Docset Information**.

Step 2 of 3: Docset information

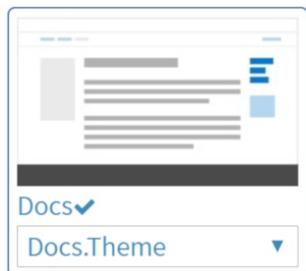
X

Docset Name ?

▼

Docset Folder ?

Site/Themes ?



Tenant ?

▼

Template ?

▼

Provisions a learn repo with necessary configuration and validation step.

Base Url ?

 learn

Show More Settings

Previous

Next

Cancel

Choose the following options:

- **Docset Name:** Duplicate the repo name
- **Docset Folder:** Duplicate the repo name
- **Site/Themes:** Docs
- **Tenant:** C+E
- **Template:** Microsoft Learn
- **Base URL:** learn

3. Finally, Start the process to build the new repo.

Step 3 of 3: Review your selections and click Start

X

Docset Name:	learn-test-pr
Docset Source Folder:	./learn-test-pr
Base Url:	https://docs.microsoft.com/{locale}/learn
Template:	Microsoft Learn

The base URL is shared with other docset(s). You can update the priority in Docset Settings.

Repository	Action	Progress	Result
MicrosoftDocs/learn-test-pr	Create + Provision	<div style="width: 0%; background-color: #ccc; height: 10px;"></div>	Not Started

[Previous](#)

[Start](#)

[Cancel](#)

Modify the Microsoft Learn repo

These files/folders should be created underneath the /docset-name/ folder.

1. Add the following files:

- /achievements/ folder
- /paths/ folder

2. Delete the following row from the `achievements.yml` file:

`achievements: []`

3. Update the `docfx.json` content mappings:

Product folders

```
{  
  "src": "<PRODUCT_NAME>",  
  "files": [  
    "**.yml"  
  ],  
  "dest": "modules"  
},
```

Exclude

Add "`<PRODUCT_NAME>/**`" to the exclude list.

```
"exclude": [
    "**/obj/**",
    "**/includes/**",
    "_themes/**",
    "_themes.pdf/**",
    "README.md",
    "LICENSE",
    "LICENSE-CODE",
    "ThirdPartyNotices",
    "<PRODUCT_NAME>/**"
]
```

File format Support

If you need to support additional file formats, such as `*.svg`, add them to the `resource` section of the build configurations.

```
"resource": [
{
    "files": [
        ...
        "**/*.{svg}",
        ...
    ]
}
```

Modify GitHub settings

IMPORTANT

The following settings only apply to primary Learn content repos (where modules, learning paths) are stored. Not LearnShared or code sample repos.

Learn content repos

1. Go to the GitHub repo settings: <https://github.com/MicrosoftDocs/<repo-name>/settings>.
2. In **Options**, only the following items should be checked. (Note: GitHub issues should not be turned on at this time!)
 - Features > **Restrict editing to users in teams with push access only**
 - Features > **Allow forking**
 - Merge button > **Allow merge commits**
 - Merge button > **Allow squash merging**
 - Merge button > **Allow rebase merging**
3. In **Collaborators & teams**, grant the **Pubdesk** team **Write** access.
4. In **Collaborators & teams**, grant the **MSFT Learn Repo Managers** team **Admin** access.
5. In **Collaborators & teams**, grant **Tyson Nevil Admin** access.
6. In **Branches**, add **two branch protection rules** for **master** and **live**. The following items should be checked. Make sure to **Save changes** for it to apply!
 - Require status checks to pass before merging
 - Require branches to be up to date before merging

- OpenPublishing.Build
- license/cla
- Restrict who can push to matching branches (Organization and repository administrators, MicrosoftDocs/pubdesk)

Go-live tasks

These tasks are one-time only tasks that should be done as a part of the initial launch process.

Enable the hierarchy service

IMPORTANT

You've prepared the repo, and now you're ready to reach out to the [APEX TripleCrown Backend](#) team to finalize onboarding.

1. Add the new Learn repo into backend sync configuration **when this repo is ready for content writers to contribute.**
 - a. Determine the depot name for your repo.
 - a. Go to one of the published URL for this new repo, such as a pull request build review URL.
 - b. View the page source and find the `depot_name` meta tag. You will use the value from this tag for the next step.
 - b. Submit a PR [editing the sync settings](#) to add this depot into "SourceDepots". (You can [review a sample pull request here](#)).
 - a. From the sync settings file in Azure DevOps, click the upper-right **Fork** button.
 - b. With the default repository name and branch option, click the **Fork** button. The repo name above the file section will show the new fork name showing you are now seeing your fork's version of this file.
 - c. Return to the `syncsettings.Prod.json` file, now in your fork.
 - d. Click the **Edit** button at the top of the file section.
 - e. Add the depot name from above as a new entry in the `SourceDepots` JSON array object. (Make sure to add a comma to the previously last item, if adding to the end of the array.)
 - f. Commit the changes to the file in your fork.
 - g. After the commit, click the **Create a pull request** link that appeared above the file section.
 - h. Give your pull request the title `Add new source depot: {depot-name} for learn repo`, replacing `{depot-name}` with the value you are using.
 - i. Click the **Create** button to submit your pull request.
 - c. Wait for the pull request to be approved and merged by the engineering team, since they need to double check to avoid hierarchy sync failures.
 2. Notify [APEX TripleCrown Backend](#) to add the new products list before go-live

The request you're making is for your repo to be added to the hierarchy service. This currently requires a manual code deployment from the team.

- For en-us repos, this will enable features across all branches. For international repos, the hierarchy services reads the master and live branches.
- Once the hierarchy service is enabled, the last remaining steps to go live are:
 - Merge master into live
 - Go to [OPS](#), where you'll soon be on the *highway to the Danger Zone*.

Danger Zone

Enable Go Live

Make Online

Make your content available to the public.

Current Status: Offline

Press the big red button to Go Live.

TIP

Some examples of the files you'll be working with are below. You can use these to do a review against the steps above if you don't actively have a repo to stage.

- [/learn-pr/learn-pr/docfx.json](#)
- [/learn-pr/openpublishing.publish.config.json](#)

Request scheduled publishing

The Content Production Services team (CPS - formerly known as Pubdesk) will perform on-demand or recurring publishing services for Docs repos.

1. Submit a recurring publishing request ([blank template](#), [copy an existing master to live](#), [copy an existing live to loc-live](#)) work item for a twice a day push from master->live (so that updates to master get pushed to the live site) and a one time per week push from live->loc-live (so that the localization team has a snapshot of the live branch to send off to vendors).

NOTE

Publishing does not occur on US holidays.

The work item details should be filled out as follows.

Twice a day request for master->live publishing:

```
Link to public repo: N/A
Link to private repo repo: https://github.com/microsoftdocs/<repo-name>
Branches to merge: master -> live
Recurring schedule: 10:00am PT, 4:00pm PT
Content Owner: Barry Langer (repo owner), Ashley Johnson (repo owner), Adrian Stevens (repo owner), Adam Patridge (repo owner)
Content Owner Email: <Learn team POC email>, learn-repo-managers@microsoft.com
Status mailing list/DG (if needed): learn-repo-managers@microsoft.com
```

NOTE

If the content is merged to live branch while go-live flag not enabled, content browse service will use the go-live flag tagged in hierarchy service to hide these not-go-live-yet contents.

Twice a week request for live->loc-live publishing:

Link to public repo: N/A
Link to private repo repo: <https://github.com/microsoftdocs/<repo-name>>
Branches to merge: live -> loc-live
Recurring schedule: Thursdays, 11AM PDT
Content Owner: Barry Langer (repo owner), Ashley Johnson (repo owner), Adrian Stevens (repo owner), Adam Patridge (repo owner)
Content Owner Email: barlan@microsoft.com, learn-repo-managers@microsoft.com
Status mailing list/DG (if needed): learn-repo-managers@microsoft.com

1. Add the link to the new scheduled publishing requests to the section below.

Links to recurring scheduled publishing requests

- [learn-pr: master->live, 2x/day](#)
- [learn-bizapps-pr: master->live, 2x/day](#)
- [learnshared: master->live, 2x/day](#)
- [learn-dynamics-pr: master->live, 2x/day](#)
- [learn-m365-pr: master->live, 2x/day](#)

What is interactivity in Learn?

1/14/2022 • 10 minutes to read

Interactivity in Microsoft Learn context is defined as instructional elements that are performed by the learner with the product they are learning about as they go through the content (sometimes commonly referred to as exercises). We require that every Learn module includes interactivity so that the learner can *do* the things they are learning about in a controlled environment that helps them be successful.

Exceptions are allowed if there's a compelling reason to exclude interactivity. An example of this might be a module where the purpose is to teach the learner how to determine which Microsoft product to use to build their solution when many could be used (a [choose module](#)). In that case, we aren't teaching about a specific task with a service but are instead trying to inform the learner which ones they should use, or the pros and cons of each.

Modules that include interactivity give the learner the opportunity to apply their new knowledge and gain hands-on experience. This active learning experience helps to cement the new concepts, fail and recover in a guided fashion, and improve comprehension and retention. Having at least one opportunity to use experience the product or service being taught is considered a best practice for Learn modules.

What is a lab on MS Learn?

A lab consists of a set of exercise instructions provided in one or more units of a module, combined with access to a Microsoft-provided lab environment which the learner uses to complete the exercises. Labs allow learners to simulate the hands-on practical application of the concepts taught in a given module in a realistic fashion at no cost to the learner.

IMPORTANT

A lab can span multiple units within a single Learn module, but can't be reused across multiple modules.

Creating a Lab is part of the [module planning & creation process](#). Content Authors should plan for their lab experiences when developing their overall Learn content plans.

Learn Lab Guidelines for content authors

When planning a lab, there are several guidelines to keep in mind.

- Learn modules should take a learner no more than 60 minutes to complete, including a lab experience. Factor your lab experience into your module design.
- The lab should teach the learner to do something useful, not just demo features.
- Leverage the conceptual content as part of the exercise - have the reader recall the details vs. just showing a step-by-step tutorial.
- Respect the learner's time. Consider whether the lab experience meaningfully contributes to the module and use other formats, such as screenshot walkthroughs or a short video, to demonstrate the most simple tasks.
- Labs should demonstrate methods that are consistent with best practices and standards.
- Content authors are accountable for maintaining their labs. Learners should not have to worry about the freshness, accuracy, or relevancy of the content on Learn. Learn will remove content from the catalog when needed.
- Take advantage of MS Learn functionality to enhance your learner's experience. For example, use the [Zone](#)

[Pivot functionality](#) to provide different versions of lab exercises within the same unit. For example:

- When a lab exercise can be completed from either the Cloud Shell or the Azure Portal GUI, consider creating separate versions of the lab exercises for each and allowing the Learner to select their preferred approach.
- Create "beginner" and "advanced" versions of the lab exercises, allowing more in-depth explanation and granular instructions for beginners while allowing advanced learners to focus on information more relevant to them.

For more tips, see [Guidance for creating technical exercise content](#)

Lab environments

There are several hosted lab environments that can be used to execute a lab in Microsoft Learn:

1. Azure sandbox
2. Dynamics 365 sandbox for Sales
3. Jupyter Notebooks sandbox
4. Labs on Demand (VM)
5. Try.NET C# REPL

All of these environments allow learners to try Microsoft products for free.

Azure sandbox

The Azure sandbox is the *preferred* environment for Azure-based labs. Learners sign into the Azure Sandbox environment with their Microsoft account and access a Microsoft-owned Azure subscription. There is no need to enter any personal information/credit card to complete the exercise. The lab can use an integrated Cloud Shell (a browser-based command-line for Azure) or the Azure portal to create and manipulate Azure resources such as databases, virtual machines, and websites.

Pros

- Free for the learner with no requirement to have an existing Azure account.
- The learner doesn't have to give Microsoft any personal information beyond email and name (taken from the logged on Microsoft account).
- Azure cost is billed at internal rates (AIRS).
- Can decide the resources the learner can create.
- Can validate what the learner has created.
- Can control how long the environment exists (1-4 hours).
- Each module gets a unique (clean) environment.

Cons

- Exercises cannot span modules as each module is given a new environment.
- Learners cannot (easily) save their work.
- Only have access to a single Azure Resource Group (RG) which is created as part of the environment. services that require multiple resource groups are currently not supported.
- When the Sandbox times-out, all resources are deleted and all work is lost.
- Learners can only activate 10 Sandboxes per day.
- Cannot use or create resources that require a service principal (a user in Active Directory).
- Cannot create or use resources that are not controlled by Azure Policy or don't exist in a resource group.

Read the [Add an Azure sandbox-based lab](#) article for more detail.

Dynamics 365 sandbox for Sales

MS Learn has an integrated Dynamics 365 experience for Sales and Customer Engagement products. This

environment is similar to the Azure sandbox. The user is placed into a D365 business group and has access to the environment with a pre-loaded set of sales data for 1-4 hours.

Read the [Add a Dynamics 365 Sales sandbox-based lab](#) article for more detail.

Jupyter Notebooks

Jupyter Notebooks has become a primary data science tool used to analyze, test, and make predictions from data sets. Jupyter is now a mainstream technology in academia, leveraged by many universities as a primary vehicle for data science training. MS Learn has an embedded Notebooks experience that enables exercise units to be written using a Jupyter Notebook.

This format dictates a slightly different flow for coding exercises which involves Markdown content and code cells be interleaved within the single unit. Currently, the only language available is Python - however the experience does support different environments:

- Python 3.6
- Python 3.6 with TensorFlow
- Python 3.6 with PyTorch
- Python 3.7
- Python 3.7 with TensorFlow
- Python 3.7 with PyTorch

Read the [Add a Jupyter Notebook sandbox-based lab](#) article for more detail.

Labs On Demand

Labs on Demand (LOD) is a third-party vendor that supplies a virtual machine environment over the Internet. Learn modules can use this approach when the service being explored is not supported in the Azure Sandbox environment. This includes labs for non-Azure products, labs that require a pre-installed environment, labs that require elevated Azure permissions, or labs that need a more sophisticated setup such as Azure Stack.

Choose a VM Lab if Learn's [Azure Sandbox](#) can't support your lab experience. VM Labs are typically required when:

- An Azure lab experience requires elevated permissions (higher than "contributor") to an Azure resource group, or access to more than one resource group.
- A lab experience is required for Microsoft products other than Azure (Microsoft 365, Business Applications, etc.)
- The lab experience requires pre-installed software, or a pre-configured environment.
- The lab experience requires the learner to interact with another technology stack which must also be virtualized.

LOD supports two interactivity options: Virtual Machine (VM) or Cloud Slice.

OPTION	DESCRIPTION
Virtual Machine	The VM-based lab enables lab scenarios that require a Windows or Linux environment with a set of pre-installed tools. For example, a Visual Studio Enterprise installation. In this model, the content team builds a VM image used as the basis for all lab environments. Activating a LOD VM-based experience opens a new browser tab with a prompt to login to the LOD environment with a set of provided credentials. Once signed in, the learner is presented with a desktop view of a remote computer along with the lab instructions on the right side of the screen. This approach uses Azure subscriptions that are owned by Microsoft.

OPTION	DESCRIPTION
Cloud Slice	The Cloud Slice environment provides access to a dynamically created Azure subscription using a set of supplied credentials. This approach leverages LOD's capability as an Azure EA reseller and enables lab scenarios that require subscription-level privileges. Activating a Cloud Slice experience opens a new browser tab that includes the username/password to log into Azure. There is no VM in this case - instead, the learner signs into the Azure portal using the provided credentials that provides access to an LOD-owned Azure subscription.

Pros

- Free for the learner with no requirement to have an existing Azure account.
- VM environment can provide installed tools such as SQL Server Management Studio or Visual Studio.
- Can control VM CPU, memory, disk space, etc.
- Enforces time limits to control costs (1-4 hours).
- Fraud detection and reporting monitored by LOD.
- Cloud Slice allows subscription-level privileges that enables support for AAD, etc.

Cons

- Exercises cannot span modules as each module is given a new environment.
- Experience isn't as integrated. Requires a new browser tab and unique login to their environment. Learner must switch back to the original tab to continue the module.
- Cloud Slice is priced at retail Azure costs + 20% markup.
- Learners cannot (easily) save their work.
- By default, only 30 concurrent users can launch VM-based labs for Azure. Teams can pay an additional cost to up this limit. Dynamics and PowerApps allow 100 concurrent users.

Check the [Add a Learn Lab to a unit using a VM lab experience](#) article for more detail on creating a VM-based lab.

Contact [Mark Smith](#) on the Learn product team for information on how to use Cloud Slice.

C# REPL

The Try.NET experience is supported in Learn content. Check the [Add a C# based lab to a unit](#) for more details.

What if I can't use any of the hosted environments?

In some cases, interactivity is desired, but cannot be enabled using the supported managed environments. In these instances, there are two available options:

1. Use an interactive "click-through" video
2. Use a personal subscription/account.

Interactive video

A second option is to record an interactive video and embed it into the content. This allows the learner to see how something works and potentially interact with the product in a simulation without requiring any installs or registration. We are currently exploring options for this type of interactivity. Please contact the Learn team if you are interested in exploring this option.

Personal subscriptions

Labs can direct the learners to install the product or use trials to learn the product. In these cases, care should be taken to ensure that the exercises always provide a no-cost route so we don't violate our 'Free Learning' tenant

of Microsoft Learn. This is not a preferred option as it adds more complexity to the learning, as well as a burden on the learner to install software or create an account.

IMPORTANT

Always make sure you have no other options before using a personal subscription. Having users create Azure accounts requires entering personal information - including credit cards, which can limit the audience significantly worldwide. In addition, it adds friction to the experience and can have a cost on the user which violates the "free learning" tenant of MS Learn.

Pros

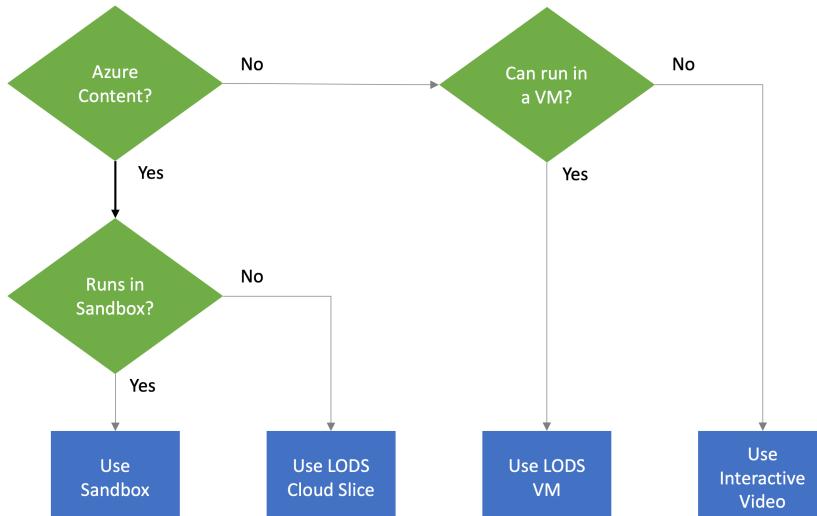
- Free to Microsoft.
- The work is all done in the learner subscription. They can do whatever they want with it and keep it as long as they want.
- Instructions are completely real – exactly what they'd have to do on their own.
- Easiest to write for since you can test using your own account.
- Work can carry forward from one module to the next since the learner controls the lifetime.
- You can do (almost) anything.

Cons

- It's not free. Might even have unexpected costs if the learner doesn't delete the resources.
- Misleading marketing messaging to learners – we promote that Microsoft Learn is free all over the site and marketing, but personal subscriptions and software are not.
- We can't (currently) use the embedded Cloud Shell environment because it's tied to the Sandbox today. That means the learner has to open a new tab to do exercise work or open a command-line prompt on their local computer.
- There are a few things that are limited to domain owners – particularly with Azure Active Directory.

Choosing the interactivity type to use

When authors are designing their content, they scope out what they want the learner to learn/do and design the best exercise/interactivity to support that. Next, they must confirm which interactivity option they can use to deliver this experience. They would follow a logic tree that is more expansive, but looks like this:



Need Help or have a question?

If you run into problems or have questions about labs, you can post them in the [Docs Support channel in Teams](#).

Reference guide for reviewing pull requests (PRs) in the Learn repo

1/14/2022 • 5 minutes to read

This reference guide is intended to help review pull requests (PRs) in the Learn repos.

For a list of the items checked during PR review, see [PR review checklist](#). All Learn PRs must meet the criteria applicable to *All* and *Learn* PRs.

Quick reference

The following sections include the items to be checked in each area during PR review.

Prerequisites

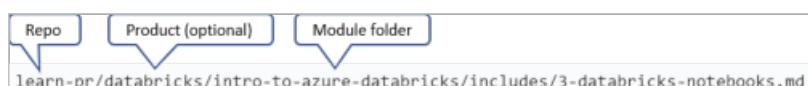
1. Close any pull request for a new module that is made against the main branch. New modules should be added against a release branch only.
2. New module PRs usually have 15-50+ new files in them that include files like:
 - index.yml
 - (multiple) unityml (look like 1-name.yml, 2-name.yml, and so on.)
 - (multiple) unit.md (look like 1-name.md, 2-name.md, and so on.)
 - Lots of image files

The PR title may include an indication that it's a new module PR.

Repo integrity

For new modules, no more than one module per pull request. Updates to existing content can go across modules.

To ensure that the PR includes only one module, check the breadcrumb for each file in the **Files Changed** section. All the files should be in the same module folder.



Naming

All the folders and file names should be lower case, separated by `-` (hyphen). There may be a few existing legacy files that don't follow this standard. In these cases, tag [Ashley Johnson](#).

Metadata

All Learn modules have two levels of metadata:

- `module-level` : Defined in the module `index.yml` file.
- `unit-level` : Defined in each `unit.yml` file.

Check all the YAML files to ensure that both, module and unit level metadata are defined correctly.

The metadata for `index.yml` should include:

- `title`

- `description`
- `ms.date`
- `author`
- `ms.author`
- `ms.topic` = `interactive-tutorial`
- `ms.prod` = `learning-(_product-name_). (_product-name_)`. This field can contain anything, but it should start with `learning`.

The metadata for all other (not `index`) YAML files should include:

- `title`
- `ms.date`
- `author`
- `ms.author`
- `ms.topic` = `interactive-tutorial`
- `ms.prod` = `learning-(_product-name_). (_product-name_)`. This field can contain anything, but it should start with `learning`.
-

Content

1. Module content should be for the product area scoped for the repository. To quickly check the product area for the module, look at the `ms.prod` metadata in the `index.yml` file. It may not be an exact one-to-one mapping. For example, instead of `learning-azure`, it could be `learning-ai`, but should be close enough. The following lists show all learn repos:

- `learn-pr` : Azure and everything that doesn't fit into one of the other repos.
- `learn-bizapps-pr` : Power BI, Flow, PowerApps, and Dynamics (this repo will move out of here soon)
- `learn-dynamics-pr` : Dynamics
- `learn-windows-pr` : Windows
- `learn-sqlserver-pr` : SQL Server
- `learn-m365-pr` : Microsoft 365
- `learn-xamarin-pr` : Xamarin

For a detailed mapping of content to repo, see the [Learn repositories](#) article. If content appears to be in the wrong repo, contact [learn-repo-managers](#).

2. In Markdown files, only H2 and lower headings are allowed. This style is enforced because the site automatically adds the H1 based off of the `title` property in the unit YAML file. If a content author adds it in the Markdown, it will appear twice.
3. Each new module must contain at least three units (YAML files), three corresponding Markdown files. One unit must be an **introduction**, one unit must be a **summary**, and the introduction unit must have *learning goals* and *prerequisites*. The module must include at least one knowledge check or task validation.
 - The knowledge checks can either be presented as stand-alone units or embedded with a learning unit. In either case, the corresponding unit YAML file should have specific metadata `quiz:` for questions and corrective feedback. See the [sample knowledge check](#) YAML file for detailed structure and metadata.
 - The task validations are defined via the `tasks:` metadata in the unit YAML file. For more information, see the [Add task validation to a Sandbox lab](#) article.
4. Don't use the word "quiz" anywhere in the content. The only acceptable value is the YAML metadata

property `quiz`:

Redirection

Redirections must be applied when files are renamed or deleted for published content. For Learn content, you may need to change the module or learning path folder names. If you need to make a change, redirections are needed for each YAML file within the folder. Redirections are needed because the folder name forms the part of the URL for each file. Changing the folder name will result in broken URL link for all files in the directory.

When to redirect

Redirections are required when any of the following items are renamed or deleted:

- **Module folder:** Renaming or deleting a module folder will need redirection for each YAML file within that folder.
- **Learning path folder:** Renaming or deleting a learning path folder will need redirection for the path `index.yml`.
- **YAML file:** Renaming or deleting any YAML file will require redirection for that specific file.

When *not* to redirect

For Learn content, the `includes` folder contains Markdown files. Any changes to these file names don't need a redirection. However, renaming or deleting the Markdown files will need changes to the file name reference in the `includes` tag under the `content` metadata of the specific unit YAML file.

In short:

- For renaming or deleting Markdown files, no redirection required.
- For renaming or deleting YAML files, redirection is required for each file.

Redirection file

All redirections must be added to the main redirection file in the repo. If the repo doesn't have a redirection file yet, one can be added from another repo. For more information, see the [learn-pr redirection file](#). All existing redirections must be deleted and new redirections added. Contact [Ashley Johnson](#) if you need assistance.

See detailed documentation on [implementing redirections](#) in the [Learn Contributor Guide](#).

Video

All embedded videos must be on RedTiger hosting. Any external video links must go to Microsoft channels, like a Microsoft-sponsored channel on YouTube. Links shouldn't reference YouTube directly.

Example of embedded video link

```
> [ !VIDEO https://www.microsoft.com/videoplayer/embed/RE2yuas ] The link should include Microsoft and not a YouTube.
```

Example of regular link

```
[Azure fundamentals]( https://www.microsoft.com/videoplayer/embed/RE2yuas )
```

It's ok to have YouTube links, as long as they are in a unit called 'summary'.

Images

All media and image files for a module are in the `media` folder within the module folder. However, the achievement files for badges and trophies are saved in a common `achievements` folder for the entire repo. See the [Request achievement images](#) article for guidelines about achievements images.

Other files

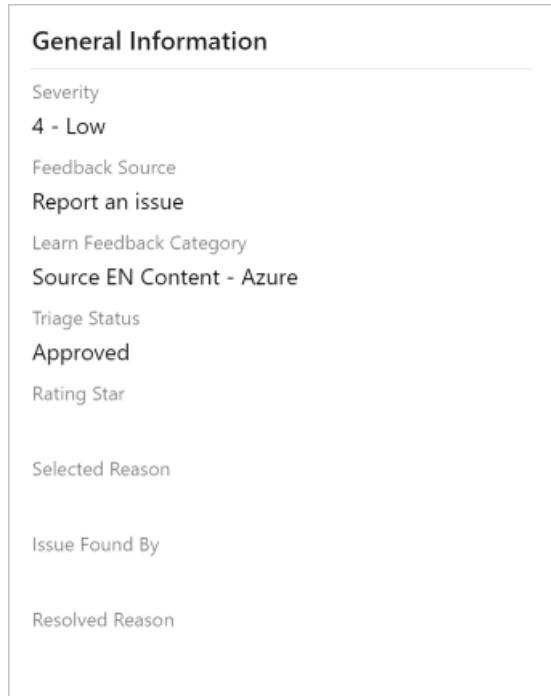
A PR will generally include `.yaml`, `.md`, or image files. However, sometimes other file types might be added.

- `PPTX` files are allowed in the repo as they can be a source for an image.
- `.svg` files are allowed as they're source files for conceptual and achievement images.
- `Issues.txt` file is **not** allowed.

Understanding the customer feedback work item

1/14/2022 • 2 minutes to read

Azure DevOps users provide feedback using a customer feedback work item. Customer feedback is relevant to many of our teams. Here's the *general information* section of a work item.



Work item entries

The following table describes the fields within the *general information* section of a work item.

	DESCRIPTION
Severity	The customer's assessment of issue severity, which is based on severity and SLA guidelines .
Feedback Source	The origin of a work item. Common values are Ratings or Report an issue . The team that handles initial feedback logs this information.
Learn Feedback Category	The issue category. Issues commonly fall into content, sandbox, or localization categories. The localization team uses this information.
Triage Status	This is initially blank, but is set to Approved when the feedback is ready for content author review. After reviewing, the content author sets a triage status, like Approved , Rejected , or Needs more info . The team that handles initial feedback tracks the work item's status. The localization team tracks the days spent from submitted to resolved.
Rating Star	The user's rating of 1 to 5 stars. The rating is used by the team that handles initial feedback.

	DESCRIPTION
Issue Found By	The type of user reporting the issue, which is usually Customer . In special cases, the user type is Partner , MVP , or Academia for issues originating from academic programs.
Resolved Reason	The outcome when the issue is closed, like Fixed , Duplicate , or As-designed . This value is set by the person who diagnoses or resolves the work item.
Cause of Issue	The likely cause of the issue. This value is set by the person who diagnoses or resolves the work item. If the reported issue is a duplicate of another work item, use the same cause of issue as the other work item. See Cause of issue for a list of allowable values.

Cause of issue

The following table describes the allowable values for *cause of issue*.

	DESCRIPTION
Cannot reproduce	You couldn't recreate the reported issue
Content needed clarity	The content wasn't incorrect, but needed rework to reduce user confusion.
Content problem	The content was incorrect or inaccurate
Covered product changed	A product change caused the user's issue
Covered product/service issue	A product outage or error caused the user's issue
Improvement suggestion	The content wasn't incorrect, but you adopted a user's suggestion to improve the user experience.
Learn platform changed	A change to the Learn platform caused the user's issue. Examples are navigation or YAML changes.
Learn platform/sandbox issue	A problem with the Learn platform caused the user's issue. Examples are server errors, sandbox issues, or failure to track user-progress.
Localization problem	A localization problem caused an issue for content in a language other than en-us English
Not enough info	You were unable to figure out the cause of issue from the work item
Other (please leave comment)	Any other issue. Add a descriptive comment to the issue to help us identify other root causes to add to this list.
Spam/Bot/Not feedback	No actionable items in this work item. Possibly caused by an automated tool hitting our feedback system.

	DESCRIPTION
User error	The user progressed through the content incorrectly or missed a crucial step. If this issue could occur for other users, select Content needed clarity and rework the content.
Verbatim-style feedback	The user's feedback is an evaluation and not actionable. For example, "Great content!".

The customer feedback work item uses standard field names and values for consistency between the Azure DevOps teams and [CEINTL](#) systems.

If you find inaccuracies on this page, you may submit a pull request with updated content.

Create the Learn Contributor Guide PDF for external sharing

1/14/2022 • 2 minutes to read

The Learn Contributor Guide provides guidance about creating Learn content. All the Microsoft FTEs and vendors with `v-` alias have direct access to the Learn Contributor Guide review site and can download a PDF version of the guide by clicking the **Download PDF** option located in the lower-left corner.

We maintain two versions of the Learn Contributor Guide:

- `Main` : A complete suite and is mostly intended for internal circulation.
- `Share` : A sub-set of the Main copy and includes only the parts of the Learn Contributor Guide that can be shared with stakeholders outside the Microsoft organization.

If you want to share the Learn Contributor guide with someone from outside of the Microsoft organization, use the Share version.

Update the Share version

The Main version of the Learn Contributor guide is always up to date and is built each time new updates are merged in the `learn-docs` repo `main` branch. However, since the Share version doesn't exactly replicate the Main copy, it needs to be updated manually.

We manually copy all new and updated files to the `share` branch and delete the files that are no longer required. The TOC is never copied from the `main` branch, as we maintain a customized TOC in the `share` branch to control the PDF content.

1. In GitHub, navigate to the `main` branch in the `learn-docs` repo using the following command.

```
git checkout main
```

2. Run the following command to get a list compare the `main` branch with the `share` branch.

```
git diff --name-status share
```

- `--name-status` : This parameter lists the name of the files with differences between the two branches. The parameter also includes the type of difference, for example *A(added)*, *M(modified)*, or *D(deleted)*.

3. Copy the added and modified files from the `main` branch to the `share` branch. Don't copy the `toc.yml` file.
4. Delete the deleted files from the `share` branch.
5. Navigate to the `share` branch.
6. Update the TOC in the `share` branch to include any new files that are appropriate to external sharing. If there are only modified and delete files, the TOC doesn't need to be updated.
7. Push changes to the upstream `share` branch.

Once the changes are merged in the upstream `share` branch, it may take few minutes for the build to run and

successfully update the branch. You can monitor the build status in the [Docs portal](#).

Create PDF from the `share` branch

You can create the PDF from the `share` branch once the build to update the branch has run successfully.

1. Navigate to the Learn guide on the [review site](#)
2. Select the `share` branch from the branches dropdown list at the top.
3. Create the PDF using the **Download PDF** option in the lower-right corner.

Planning your module

1/14/2022 • 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 [Teams](#).

This article lists the tasks of the planning phase of content contribution. It is one of a set of articles that are designed to help you develop and maintain quality content.

Planning helps a contribution effort off to an effective start by properly identifying the customer and clearly understanding content scope.

Planning is the initial phase of the content development process. It begins when a need for content is identified. This need can be based on upcoming releases or current gaps in content. The planning phase is complete when:

- a content plan is agreed upon,
- the customer is well-understood,
- stakeholders are established, and
- content publication timelines are set.

Content development plans can be tracked in Azure DevOps, Excel, GitHub, Teams, Planner, or Outlook (for example).

See also [Set up your content for success module](#) and [Microsoft Learn Content Development Process](#).

Module planning tasks

TIP

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

~	TASK	DESCRIPTION
□	Confirm stakeholders, Engage partners	Stakeholders may have been previously identified based on existing relationships and types of work. Make sure you've determined your best points of contact are that they have confirmed support.
□	Identify the customer, Propose new learning path for Learn	The product design phase is usually when the customer type is captured. Generate a customer intent statement that is agreed upon by stakeholders and content developers.

	TASK	DESCRIPTION
□	Define user scenarios, Write objectives for Learn	Consider customers, stakeholders, pain points, and marketing initiatives. You are most well prepared to draft content when you feel confident in your ability to describe a feature.
□	Browse existing content	Search for key terms and examine relevant libraries. Identify content to be included or added to.

Next steps

[Designing your module](#)

Designing your module

1/14/2022 • 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 [Teams](#).

Design Learn module

Learn modules teach things that our customers do at work. In short, our content is task-centered and not topic-centered. The tasks we teach won't be an exact match to what our customers are doing on-the-job. That's why we always include the facts and concepts that underlie what we teach. The combination of task-based instruction and underlying principles helps our customers transfer the new skills to their own work.

A module design document covers the basic components of Learn modules. The design document also includes details that can affect timelines or implementation decisions, such as interactivity choices or artwork. A good module design document helps us in a few ways:

- Guides the author to creating a task-centered module, which makes the content easier to learn.
- Simplifies the writing process since more planning is done up front.

Module design tasks

TIP

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

~	TASK	DESCRIPTION
□	Create module design document	Determine your learning objectives, prerequisites, units, interactive exercises, knowledge checks, and other content. A good module design document makes content both easier to learn and easier to create. You can capture these details in the Module design template .
□	Using a sandbox , Sandbox restrictions	Using the Azure Sandbox in a Learn module requires a few planning steps along with some administration. Once you've verified you can run your lab in the Sandbox, submit a Sandbox Module Onboarding Request Form .

	TASK	DESCRIPTION
□	Identify artwork	Microsoft Learn content generally follows the same guidance as the Docs content teams for adding art to articles . Achievement images are unique artwork to Microsoft Learn modules and learning paths, however and all Microsoft Learn achievements must follow the same branding, so all of these files are created by the Engineering Design team.
□	Add a lab, Add code samples	A lab consists of a set of exercise instructions provided in one or more units of a module, combined with access to a Microsoft-provided lab environment which the learner uses to complete the exercises. When designing labs, respect the learner's time and teach the learner to do something useful.

Next steps

[Write your module](#)

Writing your module

1/14/2022 • 2 minutes to read

Write Learn module

Modules are the building blocks of the Microsoft Learn experience. A module contains a collection of related units that teach a concept using textual content, videos, and labs. They are:

- Atomic chunks of training, like a course.
- The smallest reusable piece in the Microsoft Learn content model and can be shared across multiple Learning Paths.

Learners should have gained a new skill upon completion of a Microsoft Learn module. Learners can choose any modules to take, no matter the order. However, one of the exciting features of the platform is that content developers can string together modules into a recommended learning path for the learners who may not even know where to start or what they're looking for. Modules can be a part of a learning path, or can be standalone, "specialty" content.

A live example of an Microsoft Learn training module

Module writing tasks

TIP

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

~	TASK	DESCRIPTION
□	Create scaffold manual , Create scaffold template	Modules are a package of files (YAML, MD, and others) that drive the content that the learner interacts with. "Scaffolding" a module means creating all of the files and folders that are required for the module to successfully build. A scaffold may have no "real" learning content in it yet - but the structure, or shell is built.
□	Write an introduction , Write an introduction summary	A good introduction includes key technologies and subtasks while being brief. Introductions, along with titles and summaries, help customers decide if the Learn content will be useful to them.

	TASK	DESCRIPTION
□	Write concept and exercise units	Units in Microsoft Learn are effectively the "chapters" of a module. Each one contains a chunk of information that supports the module. As a content developer, you can think of them as a page that you can present content, quizzes, and/or interactivity with. Training has a different quality bar than other kinds of learning - the learner wants not only to become knowledgeable, but to become confident in actively applying their learning.
□	Write knowledge checks	Knowledge checks can be difficult to write. Should you use True/False questions? How many answers should you have? Should you write in the second or third person? Writing good questions is a well-studied problem and we've summarized the most common guidelines in this document.
□	Create achievements	One of the unique features of Microsoft Learn modules is that learners can earn achievement badges and trophies upon completion of the modules and learning paths. These achievements show up on their Docs user profile and can be shared via social media and other channels.

Next steps

[Publishing your module](#)

Publishing your module

1/14/2022 • 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 [Teams](#).

[Review Learn module](#), [Publish Learn module](#)

During the process of authoring, it's natural to stray from your original plan. The design doc is there to guide the process, not control the process completely. There may be things you forgot to add. As you wrote, you may have realized the module needed to be split, or content had to be added in order for the learner to be successful. Module creation should deviate from the plan when it makes sense.

During the review phase, you make sure all parts of your module are accurate and meet Learn quality standards prior to publishing. You and your peers should identify and address content quality items. You should also resolve build warnings. Once all reviews have been performed, the content is published on the Learn platform.

Module publishing tasks

TIP

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

~	TASK	DESCRIPTION
□	Submit pull request	You submit all your changes to Learn content via GitHub pull requests (PR). Depending upon the change, your PR is either merged automatically or by a human pull request reviewer after checking it against the basic quality check criteria. The PR reviewer checks each Learn pull request against a list of PR quality criteria .
□	Fix build errors and Acrolinx style issues, Guidance checklist for Learn	Tense, voice, formatting, and content types should all have the same look and feel across Learn modules. Detailed guidance includes point-based rubrics to evaluate your content. Additionally, ensure all content meets organizational level and Learn-specific compliance requirements .

	TASK	DESCRIPTION
□	Test Sandbox and module	Once the module has been published to master, make sure to test your exercise end-to-end. Testing ensures the module is registered properly and the environment activates (perform your tests on review.docs).
□	Request peer reviews and address feedback	You should request review for the technical, editorial, and instructional design aspects of your module. Final reviews are split because each is performed by different teams and have a different focus. Specialization ensures that all content has been thoroughly vetted through each lens.
□	#sign-off on pull request	Get approval after final testing, and verify that the content has published correctly on docs.microsoft.com

Next steps

[Maintaining modules](#)

Maintaining modules

1/14/2022 • 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 [Teams](#).

The maintenance phase for Learn is similar to the planning phase for Docs. Below is an excerpt from [Planning content](#):

This article identifies and defines the tasks that comprise content maintenance. 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 ensure content continues to meet customer expectations.

Maintenance is the final, ongoing phase of the content development process. Although it occurs after the content has been substantively developed, and it might involve content that wasn't yours to begin with, maintenance is as important as any other content development phase.

Content should be monitored indirectly, as through metrics or customer feedback, as well as directly, as by content review for up-to-date references and relevance. Content contributors should evaluate the fitness of content and decide how best to improve the content, including possibly standing the content down.

Finding opportunities to extend and improve content can trigger another round of planning and often loops you back to the beginning of the content development process.

See also [Maintain Learn module](#).

Module maintenance tasks

TIP

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

~	TASK	DESCRIPTION
□	Triage and address customer feedback	As internal and external feedback is received, issues should be organized and addressed by priority. After issues are addressed, you open a pull request to incorporate updates. Support options for Learn · Support Triage for Learn

	TASK	DESCRIPTION
□	Optimize content, SEO basics	Content publication is based on prioritization; you may have to defer specific content development until a later point. You can optimize existing content by identifying ways to increase its value to the customer.
□	Validate freshness	The longer content has been published on the live site, the more likely it should be reviewed for accuracy. Dependencies, brand names, and metadata can change over time.
□	Review non-performing content, BI/Reporting dashboards for Learn	Content might not receive feedback or views for various reasons. It might not be sufficiently targeted or detailed, or users might be looking for it somewhere else. Under-performing content should be addressed as too much of it damages customer trust in the effectiveness of the knowledge base.
□	Retire out-of-date content, Retiring content for Learn	Outdated content can make a customer spend time looking for answers in the wrong places. If content is for a product or feature that has been discontinued, or a process that is not valid in several steps, it may be best to retire the content.

Next steps

[Start contributing to Learn](#)

Working on Learn content in GitHub

1/14/2022 • 4 minutes to read

There's a series of tasks/commands to use when working in GitHub to submit new content and edits to existing content. The standard workflow can be roughly broken down to the following steps:

1. Fork the repo
2. Follow the workflow for working against `master` or release branches
3. Do the work in Markdown
4. Submit the work to GitHub

Terminology

For common GitHub terms, like **fork**, **remote**, and **repository**, check out <https://help.github.com/articles/github-glossary>.

Fork the repo

Create a fork of the repo on GitHub. Follow the steps in the [this Docs contributor guide doc](#) to create a fork and clone the forked repo locally.

Daily workflow

The first time you work on a given repo

COMMAND	WHAT IT DOES
<code>git remote -v</code>	Confirm that the remote repo locations are set up properly. This should output <code>fetch</code> and <code>push</code> lines for <code>origin</code> and <code>upstream</code> . Origin should have <code>your-git-username</code> in the URL and upstream should have <code>MicrosoftDocs</code> .
<code>git fetch upstream</code> or <code>git fetch upstream <release-branch></code> .	The addition of <code>upstream</code> or <code><release branch name></code> is to tell git to track against your release branch instead of master.

For working against release branches (e.g., `release-ignite18`)

COMMAND	WHAT IT DOES
<code>cd folder/repo</code>	Changes directory to wherever your repo is cloned to on your local system.
<code>git checkout master</code>	Check out <code>master</code> branch.
<code>git pull upstream master</code>	Pulls the current copies of all files, including new files, from <code>master</code> into <code>current-branch</code>
<code>git checkout -B <your new local branch name, like "barlan0511"> upstream/<release-branch-name></code>	Check out a new branch and set its upstream against the non-master branch you're interested in working with.

NOTE

If you get the error `fatal: 'upstream/branch-name' is not a commit and a branch 'branch-name' cannot be created from it`, you need to do `git fetch upstream` before you `git checkout`. This pulls down any upstream branches onto your local system that you can then check out.

At this point, you're ready to write your [content in Markdown](#).

After writing your content, you need to complete a set of commands to push the content from your local branch to the remote branch on GitHub.

COMMAND	WHAT IT DOES
<code>cd folder/repo</code>	Changes directory to wherever your repo is cloned to on your local system.
<code>git pull upstream <release branch name></code>	Confirm that you have the newest content on your local copy of the branch by pulling any new changes that have come in since you started working.
<code>git add -A</code>	Add all content you've been working on to your pull request.
<code>git commit -m "commit message"</code>	Write the first comment for your pull request and for this individual commit
<code>git push origin <branch name></code>	Submit your content to GitHub
Open https://github.com/MicrosoftDocs/repo-name	Create your PR at https://github.com .
Go to Pull Requests or look for the yellow notification bar	The base branch will show as master; change it to the release branch on the "Open a pull request" page. This will confirm that your content lands in the upstream branch where you were tracking changes from.

For working against master (e.g., ongoing changes)

COMMAND	WHAT IT DOES
<code>git checkout master</code>	Sets you up in your local copy of <code>master</code>
<code>git pull upstream master</code>	Pulls the current copies of all files, including new files, from remote <code>master</code> into local <code>master</code>
<code>git checkout -B <name of new branch></code>	Check out a new branch and sets its upstream against <code>master</code>
<code>git add -A</code>	Add all content you've been working on to your pull request.
<code>git commit -m "commit message"</code>	Write the first comment for your pull request and for this individual commit
<code>git push origin <branch name></code>	Submit your content to GitHub
Open https://github.com/MicrosoftDocs/repo-name	Create your PR at https://github.com

COMMAND	WHAT IT DOES
Go to Pull Requests or look for the yellow notification bar	The base branch will show as master; change it to the release branch on the "Open a pull request" page. This will confirm that your content lands in the upstream branch where you were tracking changes from.

At this point, you're ready to write your [content in Markdown](#).

After writing your content, you need to complete a set of commands to push the content from your local branch to the remote branch on GitHub. Note that the steps below are the same as those used in the release branch process.

COMMAND	WHAT IT DOES
<code>cd folder/repo</code>	Changes directory to wherever your repo is cloned to on your local system.
<code>git pull upstream <release branch name></code>	Confirm that you have the newest content on your local copy of the branch by pulling any new changes that have come in since you started working.
<code>git add -A</code>	Add all content you've been working on to your pull request.
<code>git commit -m "commit message"</code>	Write the first comment for your pull request and for this individual commit
<code>git push origin <branch name></code>	Submit your content to GitHub
Open https://github.com/MicrosoftDocs/repo-name	Create your PR at https://github.com .
Go to Pull Requests or look for the yellow notification bar	The base branch will show as master; change it to the release branch on the "Open a pull request" page. This will confirm that your content lands in the upstream branch where you were tracking changes from.

Cleaning up your branches

After your PR is merged, you can clean up the remote branch and local branch. To do this run the following git commands:

COMMAND	WHAT IT DOES
<code>git checkout <another_branch_name></code>	Switch to a different branch. You can't delete a branch you're actively checked out in.
<code>git push --delete origin <branch_name></code>	Delete remote branch. This can also be done via the delete branch button that appears in your PR on GitHub.
<code>git branch -d <branch_name></code>	Delete local branch

GitHub resources

- [GitHub commands cheat sheet](#)

- GitHub overview: Docs contributor guide

Module Design - Markdown

1/14/2022 • 2 minutes to read

```
# Title

Add the working title [(Title guidance)](/help/learn/id-guidance-title)

## Role(s)

Add the role(s) [Role guidance](/new-hope/information-architecture/metadata/taxonomies?branch=main#role)

## Level

Add the level [Level guidance](/new-hope/information-architecture/metadata/taxonomies?branch=main#level)

## Product(s)

Add the product(s)

## Prerequisites

List the prerequisites [(Prerequisite guidance)](/help/learn/id-guidance-prerequisites)

## Summary

Add the summary [(Summary guidance)](/help/learn/id-guidance-introductory-summaries)

## Learning objectives

1. Add numbered Learning Objectives [(Learning objective guidance)](/help/learn/id-guidance-learning-objectives)

## Chunk your content into subtasks

Identify the subtasks of module title.

| Subtask | What part of the introduction scenario does this subtask satisfy? | How will you assess it: Exercise or Knowledge check? | Which learning objective(s) does this help meet? | Does the subtask have enough learning content to justify an entire unit? If not, which other subtask will you combine it with? |
| ---- | ---- | ---- | ---- | ---- |
| TODO | TODO | TODO | TODO | TODO |
| TODO | TODO | TODO | TODO | TODO |
| TODO | TODO | TODO | TODO | TODO |

## Outline the units

Add more units as needed for your content

1. Introduction

Provide a scenario of a real-world job task that shows how the technology is used in practice:

Add your scenario [(Scenario guidance)](/help/learn/id-guidance-scenarios)

1. Learning-content unit title

List the content that will enable the learner to complete the subtask:



- Enabling objective
  - Information needed to accomplish the enabling objective
  - Information needed to accomplish the enabling objective
- Enabling objective
  - Information needed to accomplish the enabling objective

```

- Information needed to accomplish the enabling objective
- Enabling objective
 - Information needed to accomplish the enabling objective
 - Information needed to accomplish the enabling objective

Knowledge check

What types of questions will test the learning objective? [(Knowledge check guidance)](/help/learn/id-guidance-knowledge-check)

- Question type
- Question type

1. Exercise - exercise unit title

List the steps that apply the learning content from previous unit:

1. Step
1. Step
1. Step

1. Summary

How did you solve the problem in the initial scenario with the knowledge learned in the module?

Add your summary [(Summary guidance)](/help/learn/id-guidance-module-summary-unit)

Notes

Note any additional information that may be beneficial to this content such as links, reference material, and so on.

Definitions of Learning terms

1/14/2022 • 6 minutes to read

TERM	DEFINITION	EXAMPLE(S)
Achievements	A memento or badge awarded for successful completion of a course.	Digital trophies and badges awarded for completing learning paths and modules in Learn.
Assessments	Evaluates learner's accomplishment of intended learning outcomes for a specific course or training program.	Knowledge checks in Learn modules.
Blueprint	A list of the content areas included in a learning product. Also includes the relative importance of those areas within that content area. The blueprint is developed by subject matter experts (SMEs) through a survey in most cases.	Design documents for Learn modules.
Bootcamps	Intense deep-dive learning events to immerse the learner in the training event. Bootcamps are often instructor led with labs, activities, homework, and so on. Bootcamps are also frequently tied to certification preparation, but not always.	
Certificate	Printable representation of an accomplishment. For example, this certificate can be a PDF. It includes the person's name, date achieved, accomplishment, stamp, and organizational acknowledgment.	
Certificate of Completion	Provides evidence that the learner attended or passed a course, training program, or learning event.	<ul style="list-style-type: none">• edX MOOC course completion certificate• LaaS clients earn a Microsoft Certificate for passing a MOOC• MOC Course Completion Certificate• MOC On-Demand Course Completion Certificate

TERM	DEFINITION	EXAMPLE(S)
Certification	<p>Provides independent evaluation of specific skills and competencies that a person has. These skills and competencies are required for successful performance in a professional role or specific work-related tasks. Specifically, certifications:</p> <ul style="list-style-type: none"> • Validate the application of knowledge and skills separate from learning. • Ensure that the person is who they say they are (identity verification). • Ensure that person who submitted work <i>did</i> the work. 	Microsoft Certified Professional
Courses	Any learning event that has a clear beginning and end. Courses can be self-paced, or instructor led, in person, or virtual.	Microsoft Learn module.
Credential/Credentialing	Evidence that the learner accomplished or demonstrated something. Assessments, certifications, certificates, and badges are credentials	
Digital Achievements	A memento or badge awarded for successful completion of a course.	Digital trophies and badges awarded for completing learning paths and modules in Learn.
Facilitated	Learners are supported in some way during the learning event. Support can come from instructors, mentors, or tutors.	
How-tos	Step-by-step guide to complete a specific task or an in-depth education on a specific subject.	
Instructor-facilitated Training (IFT)	An instructor assists students as they complete labs to practice the skills they learned before the training event through self-study. IFTs are often done in a workshop format. Learning is done in advance and practiced in IFT workshop.	
Instructor-led Training (ILT)	An instructor leads the teaching of the course's learning goals. ILT can be in person or a virtual experience, sometimes referred to as Facilitated Virtual Training.	

TERM	DEFINITION	EXAMPLE(S)
Job Task Analysis (JTA)	The process of using focus groups and survey research to identify the tasks that individuals complete in a given job role. JTA is also used to refer to the list of tasks that results from this process.	
Just-in-Time Learning	Deliver training to learners when and where they need it. Rather than sitting through hours of traditional classroom training, learners can tap into online tutorials, interactive CD-ROMs, and other tools. These resources help to zero in on just the information they need to solve problems, complete specific tasks, or quickly update their skills.	
Knowledge	Theoretical understanding of a subject or concepts. Knowledge, by definition, is theoretical not practical.	
Knowledge Checks	Intended for learners to determine if they understand key concepts. Instructors use knowledge checks to determine if their students understand key concepts and modify learning materials as needed. Knowledge checks are a type of formative, non-graded assessment.	
Learning Paths	A series of skills and learning events that are linked together by a common end goal, ordered in a logical progression. One course or a series of courses could represent a learning path.	
Lesson	A subcomponent of a module. Sometimes referred to as unit. By design, Microsoft WWL uses lessons rather than units to describe this element of a course.	
Levels of Experience		
Mentoring	An experienced person, often called the mentor, assists the learner in developing specific skills and knowledge. Mentors usually work in a one-on-one relationship but tutors often work in a one-to-many relationship. As a result, mentoring relies heavily on relationship building to be effective.	

TERM	DEFINITION	EXAMPLE(S)
Macrolearning	Involves a larger time commitment and a focus on specified learning goals. Macrolearning is often used when choosing to engage with content that is largely unfamiliar.	
Microlearning	Short, focused learning nuggets (often 3-5 minutes or less) that are designed to meet a specific learning outcome. It involves short-term-focused strategies especially designed for skill-based understanding, learning, and education.	
Module	A subcomponent of a course. Courses consist of modules. Modules consist of lessons.	
MOOC	Massive open online course. An online course made available for little to no charge to an exceptionally large number of people.	
Objective Domains	A hierarchical outline of functional groups and goals defined to measure specific skills, abilities, and knowledge with an exam or course.	
Online Labs	Firsthands experience for either learning or assessment.	
Points		
Quick Start Guides	Allows learners to learn a task in 10 minutes or less through an interactive "try it" experience.	
Self-paced	Learners progress through the course at their own speed. If course is of limited duration, the learner must complete the requirements within that timeframe.	
Skill	The ability to complete a task or activity with an intended outcome. The ability is acquired through education, training, experience, or other means.	
Skills Validation	Process used to evaluate if someone has the required skills to complete a specific task.	
Snacks (learning)	Small bits of learning that are available on demand from any device at any time. Snacks are often associated to the concept of microlearning.	

TERM	DEFINITION	EXAMPLE(S)
Structured Learning	The learning event or path has a clear beginning, middle, and end. The path follows a logical progression throughout the learning journey and clearly guides the learner on what to do next.	
Task	Specific actions that having the appropriate skills allow you to complete.	
Tutorials	Method of transferring knowledge that is more interactive and specific than a book or a lecture. Tutorials usually teach by example and supply the information to complete a certain task. Tutorials often include specific steps to complete a task. While similar to How-tos, tutorials typically are more interactive and use a step-by-step process that ensures the learner is comprehending the material throughout the process.	
Tutoring	One-to-one or small group activity in which a person who has expertise in a specific content area or discipline provides help or clarification. Tutoring involves the teaching skills to either an individual or group of individuals. While it may be advantageous, it isn't necessary that the same person tutors an individual or a group over an extended period of time. Tutoring isn't based on a personal relationship between the person doing the tutoring and the person being tutored. Mentors, though, rely on the establishment and nurturing of a close, personal relationship with a learner over time.	
Unstructured Learning	Learning is open ended with no clear beginning, middle, or end. The learner can jump in and out and move around with little guidance.	
Units	A subcomponent of a module. Sometimes referred to as lesson.	
Validity	Checking the knowledge gained by learners after taking a course. This validation can be done via knowledge checks or hands-on exercises.	

TERM	DEFINITION	EXAMPLE(S)
Workshops	A team-based learning event with hands-on labs, whiteboarding activities, group projects, and case studies. Can also include team discussions led by an SME.	

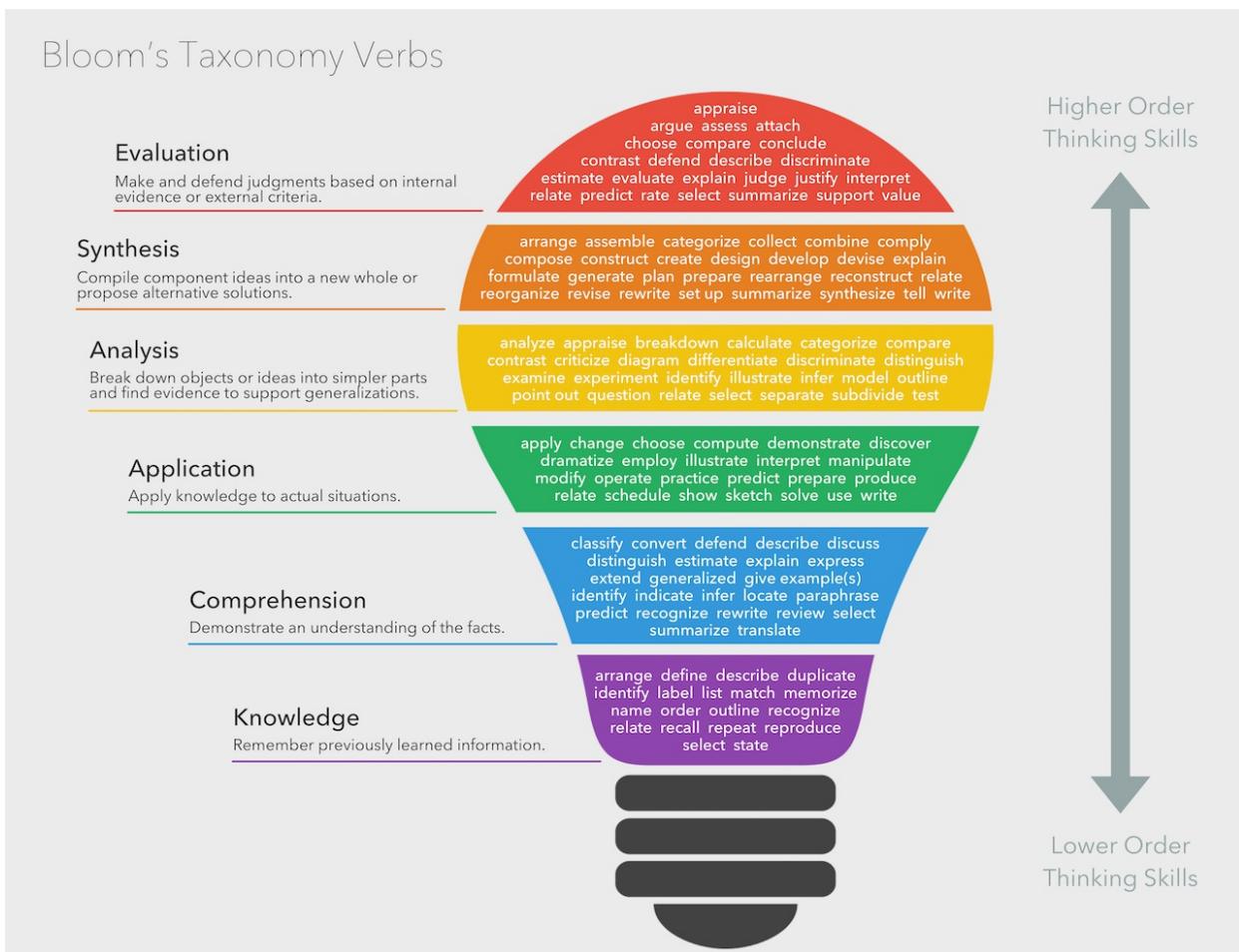
Learning theory resources

1/14/2022 • 2 minutes to read

Bloom's Taxonomy

Benjamin Bloom created a taxonomy of cognitive processes for learning goals. The processes go in order from lower order thinking to higher order. Higher-order thinking is more engaging for learners, so the closer you get to "create", the more the learner will be engaged.

- Remember: Memorize information.
- Understand: Understanding the meaning of content.
- Apply: Apply new knowledge.
- Analyze: Breaking down a subject into its parts and seeing how those parts relate.
- Evaluate: Make judgments about what you've learned.
- Create: Take new knowledge and put it together in a unique way.



[Download Blooms Verb Chart PDF.](#)

Adult learning

Malcom Knowles' theory of andragogy, or adult learning, is based on five assumptions about how adults learn:

- Self-concept: Being a self-directed and unique individual.
- Learner Experience: Adults have a lifetime of experience to draw upon.
- Readiness to Learn: Adults will decide to learn things that affect their social roles.

- Orientation to Learn: Adults want to learn things they can apply immediately.
- Motivation to Learn" Motivation for learning is driven by internal forces rather than external.

Adult learning principles

(Also from Knowles)

- Adults are self-directed. They should have a say in the content and process of their learning.
- Adults want to draw from their vast experiences. Their learning should focus on adding to what they've already learned in the past.
- Learning is relevant to their daily life. Content should focus on issues related to their work or personal life.
- Adults prefer higher order thinking to rote memorization. They enjoy solving problems.

Overview of UID

1/14/2022 • 3 minutes to read

UIDs are manually generated unique text-based identification codes that are used in Microsoft Learn to identify each learning path, module, and unit. A UID is one of the most important metadata. Though a UID is never visible to the public, it ties together most aspects of the Microsoft Learn platform. Apart from being the metadata used for identification purpose, learners' entire progress and achievements earned are tied to UIDs.

Currently in Microsoft Learn, the UIDs are used to identify the following:

- **Learning path:** Each learning path has a unique UID of its own.
- **Module:** Module UIDs are used to bind the modules to a learning path and are added to the learning path index.yaml. Add the module UIDs in the order that you want the modules to be presented to the learner.
- **Unit:** Unit UIDs are used to attach units to a module and should be added to the module index.yaml. Add the UIDs in the order that you want the units to display.
- **Achievement badge/trophy:** Achievement UID is used to identify specific badge or trophy icons earned for completing a module or a learning path. Add the badge UID to the module index.yaml and trophy UID to the learning path index.yaml files. Badges/Trophies should use the same UID as the module/path with the suffix ".badge" or ".trophy".

Naming standard

UIDs are manually generated and can be defined to any value, as long as it's unique across the entire repository. However, to ensure uniqueness and for ease of maintenance, UIDs should follow a specific format.

- **Learning path UID:** `{repo}.{team-or-content-area}.{learning-path-short-name}` Example:
`learn.azure.cloud-foundations`
- **Module UID:** `{repo}.{team-or-content-area}.{module-title-short-name}` Example:
`learn.data-ai.intro-to-azure-ml-service`
- **Unit UID:** `{moduleuid}.{unit-title-short-name}` Example:
`learn.azure.principles-cloud-computing.motivation`
- **Achievement badge UID for module:** `{moduleuid}.badge` Example:
`learn.azure.principles-cloud-computing.badge`
- **Achievement trophy UID for a learning path:** `{learningpathuid}.trophy` Example:
`learn.azure.cloud-foundations.trophy`

Where `{repo}` is the name of the repo without the "-pr" suffix. For example: learn, learn-dynamics, learn-xamarin, etc. `{team-or-content-area}` is the group managing the content (azure, data-ai, student-evangelism, iot, etc.), and `{xyz-name}` is a shortened version of the title replacing spaces with dashes and using acronyms for products. For example, "Create a machine learning experience in Azure Machine Learning Studio" might be "create-a-ml-experiment-in-ams".

IMPORTANT

Do not include any sequence number in the name portion of a UID. Unit filenames often have a numeric prefix to indicate ordering - this should be omitted in any UID for the unit so it can be re-ordered later on without changing the unique identifier in the system. For example, if the markdown file is `1-introduction.md` and the module UID is

`learn.azure.create-a-storage-account`, then the UID for the unit would be:

`learn.azure.create-a-storage-account.introduction`.

Updating UIDs

Changing the UID creates a new module in the system. People who had started the module with the old UID won't be able to complete it, and anyone who has *already* completed the module will see a new module that hasn't been started yet. For this reason, we **highly discourage** changing the UID of a published module *unless* the changes made are really creating something quite different than what was published before - where you *want* people who completed the prior version to go through this version.

Consider the act of changing the UID to be a replacement operation. If any of the YAML filenames are adjusted, or the folder is renamed, you'll need to add redirects to ensure people don't get a 404 trying to find the old URLs. In addition, a new badge should be applied to distinguish the new version of the module from the old one. Finally, you'll need to add an entry into the `achievements.yml` file with the original UID, title, summary, and URL to the trophy. Here's an example entry:

```
- uid: learn.make-your-virtual-machines-highly-available.badge
  type: badge
  title: Make your virtual machines highly available
  summary: Make your virtual machines highly available
  iconUrl: /learn/achievements/make-your-virtual-machines-highly-available.svg
```

This entry will ensure that people who have completed the module will continue to see the trophy they received.