MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
deployment-manager	Deployment Manager	(empty)	(empty)
devtest-lab	DevTest Labs	(empty)	(empty)
digital-twins	IoT Azure Digital Twins	(empty)	(empty)
dms	Data Migration Service	(empty)	(empty)
		connectivity	Connectivity
		migration-guide	Migration Guide
dns	DNS	(empty)	(empty)
documentdb	DocumentDB	(empty)	(empty)
dynamics365-accountant	Accountants	(empty)	(empty)
dynamics-365-ai	Dynamics 365 AI	(empty)	(empty)
dynamics365-business-central	Dynamics 365 Business Central	(empty)	(empty)
dynamics-365-commerce	Dynamics 365 Commerce	(empty)	(empty)
dynamics-365-connected-store	Connected Store	(empty)	(empty)
dynamics-365-cross-app	Dynamics 365 cross-applications	(empty)	(empty)
dynamics-365-customerservice	Dynamics 365 for Customer Service	(empty)	(empty)
		channel-integration-framework	Channel Integration Framework
		unified-service-desk	Unified Service Desk
dynamics-365-customervoice	Dynamics 365 Customer Voice	(empty)	(empty)
dynamics-365-field-service	Dynamics 365 Field Service	(empty)	(empty)
		common-scheduler	Common scheduler
		connected-field-service	Connected field service
		field-service-mobile	Field service mobile
		resource-scheduling-optimization	Resource scheduling optimization

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
dynamics-365-finance	Dynamics 365 Finance	(empty)	(empty)
dynamics365-financials	Dynamics 365 for Financials	(empty)	(empty)
dynamics-365-fraud-protection	Dynamics 365 Fraud Protection	(empty)	(empty)
dynamics-365-guides	Dynamics 365 Guides	(empty)	(empty)
dynamics-365-human-resources	Dynamics 365 Human Resources	(empty)	(empty)
dynamics-365-import-tool	Dynamics 365 Import Tool	(empty)	(empty)
dynamics-365-industry	Dynamics 365 Industry Solutions	(empty)	(empty)
		dynamics-365-industry-health	Dynamics 365 Healthcare
dynamics-365-intelligent-order-management	Dynamics 365 Intelligent Order Management	(empty)	(empty)
dynamics-365-linkedinconnector	Dynamics 365 Connector for LinkedIn	(empty)	(empty)
dynamics-365-marketing	Dynamics 365 for Marketing	(empty)	(empty)
dynamics-365-productvisualize	Dynamics 365 Product Visualize	(empty)	(empty)
dynamics-365-project-operations	Dynamics 365 Project Operations	(empty)	(empty)
dynamics-365-retail	Dynamics 365 for Retail	(empty)	(empty)
dynamics-365-sales	Dynamics 365 for Sales	(empty)	(empty)
dynamics-365-supply-chain	Dynamics 365 Supply Chain Management	(empty)	(empty)
dynamics-365-talent	Dynamics 365 for Talent	(empty)	(empty)
dynamics-ax-applications	Dynamics 365 for Operations Applications	(empty)	(empty)
dynamics-ax-platform	Dynamics 365 for Operations Platform	(empty)	(empty)
dynamics-lifecycle-services	Dynamics Lifecycle Services	(empty)	(empty)
dynamics-nav-2018	Dynamics NAV 2018	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
event-grid	Event Grid	(empty)	(empty)
		iot-edge	Event Grid on IoT Edge
		kubernetes	Event Grid on Kubernetes
event-hubs	Event Hubs	(empty)	(empty)
exchange-online	Exchange Online	(empty)	(empty)
exchange-powershell	Exchange PowerShell	(empty)	(empty)
expressroute	ExpressRoute	(empty)	(empty)
firewall	Azure Firewall	(empty)	(empty)
		networking	Networking
firewall-manager	Azure Firewall Manager	(empty)	(empty)
forms-pro	Forms Pro	(empty)	(empty)
fraud-protection	Fraud Protection	(empty)	(empty)
frontdoor	Azure Front Door Service	(empty)	(empty)
fxt-edge-filer	Azure FXT Edge Filer	(empty)	(empty)
gamification	Gamification	(empty)	(empty)
genomics	Microsoft Genomics	(empty)	(empty)
germany	Azure Germany	(empty)	(empty)
globalization	Globalization	(empty)	(empty)
governance	Azure Governance	(empty)	(empty)
groove-music	Groove Music	(empty)	(empty)
guest-configuration	Guest Configuration	(empty)	(empty)
guidance	Guidance	(empty)	(empty)
hdinsight	HDInsight	(empty)	(empty)
healthbot	Health Bot	(empty)	(empty)
healthcare-apis	Microsoft Healthcare APIs	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		fhir	FHIR
		iomt	IoMT connector
hotel-ads	Hotel Ads API	(empty)	(empty)
hotel-ads-hint-message	Hotel Ads Hint Message	(empty)	(empty)
hotel-ads-hotel-feed	Hotel Feed	(empty)	(empty)
hotel-ads-pos-feed	Hotel Ads POS Feed	(empty)	(empty)
hotel-ads-query-control-message	Hotel Ads Query Control Message	(empty)	(empty)
hotel-ads-query-message	Hotel Ads Query Message	(empty)	(empty)
hotel-ads-transaction-message	Hotel Ads Transaction Message	(empty)	(empty)
hpc-cache	HPC Cache	(empty)	(empty)
identity	Azure Active Directory	(empty)	(empty)
		verifiable-credentials	Verifiable Credentials
industrial-iot	Industrial IoT	(empty)	(empty)
industry	Industry	(empty)	(empty)
		financial	Financial Cloud
		healthcare	Healthcare Cloud
		manufacturing-cloud	Manufacturing Cloud
		nonprofit	Nonprofit Cloud
		retail	Retail Cloud
information-protection	Microsoft Information Protection	(empty)	(empty)
		aip	Azure Information Protection
		aiplabels	Azure Information Protection policy
		analytics	Analytics (central reporting)
		azurerms	Azure RMS

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		connector	RMS connector
		doctrack	Document tracking
		fci	Windows Server FCI
		hyok	HYOK
		kms	Key management (BYOK & MS-managed)
		labelmigrate	Label Migration
		migration	AD RMS migration
		mip	Microsoft Information Protection
		mipsdk	Microsoft Information Protection SDK
		prereqs	Requirements
		scanner	Scanner
		v1client	Classic client
		v2client	Unified labeling client
intelligent-systems	Intelligent Systems	(empty)	(empty)
internet-analyzer	Azure Internet Analyzer	(empty)	(empty)
internet-peering	Internet Peering	(empty)	(empty)
invoicing	Invoicing	(empty)	(empty)
iot-accelerators	IoT Solution Accelerators	(empty)	(empty)
iot-central	IoT Central	(empty)	(empty)
		iot-central-retail	IoT Central Retail
iot-develop	Azure IoT Device and Application Development	(empty)	(empty)
iot-dps	IoT Device Provisioning	(empty)	(empty)
iot-edge	IoT Edge	(empty)	(empty)
iot-fundamentals	IoT Fundamentals	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
iot-hub	IoT Hub	(empty)	(empty)
iot-hub-device-update	Device Update for IoT Hub	(empty)	(empty)
iot-pnp	IoT PNP	(empty)	(empty)
iot-suite	IoT Suite	(empty)	(empty)
jenkins	Jenkins on Azure	(empty)	(empty)
kaizala	Kaizala admin	(empty)	(empty)
key-vault	Key Vault	(empty)	(empty)
		certificates	Certificates
		general	General
		keys	Keys
		managed-hsm	Managed HSM
		secrets	Secrets
kusto	Kusto Query Language	(empty)	(empty)
lab-services	Lab Services	(empty)	(empty)
		classroom-labs	Classroom Labs
		custom-labs	Custom Labs
		trial-labs	Trial Labs
lighthouse	Lighthouse	(empty)	(empty)
linkedin	LinkedIn	(empty)	(empty)
linux-vms	Virtual Machines	(empty)	(empty)
load-balancer	Load Balancer	(empty)	(empty)
location-based-services	Location Based Services	(empty)	(empty)
log-analytics	Log Analytics	(empty)	(empty)
logic-apps	Logic Apps	(empty)	(empty)
m365-md	Microsoft 365 Managed Desktop	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
machine-learning	Machine Learning	(empty)	(empty)
		core	Core
		data-prep	Data prep
		studio-classic	Studio Classic
		team-data-science-process	Team Data Science Process
managed-applications	Managed Applications	(empty)	(empty)
managed-hsm	Managed HSM	(empty)	(empty)
managed-instance-apache-cassandra	Cassandra Managed Instance	(empty)	(empty)
mariadb	Azure Database for MariaDB	(empty)	(empty)
		migration-guide	Migration Guide
marketplace	Marketplace	(empty)	(empty)
		partnercenter-marketplace-publisher	PartnerCenterMarketplacePublisher
marketplace-customer	Commercial marketplace customer	(empty)	(empty)
mdm	Dynamics Marketing	(empty)	(empty)
media-services	Media Services	(empty)	(empty)
		lva-edge	Live Video Analytics on IoT Edge
		media-player	Azure Media Player
		rest-v2	Rest V2
		rest-v3	Rest V3
		video-indexer	Azure Media Services Video Indexer
mem	Microsoft Endpoint Manager	(empty)	(empty)
		fundamentals	Fundamentals
microsoft-365-education	Microsoft 365 Education	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
microsoft-academic-services	Microsoft Academic Services	(empty)	(empty)
microsoft-defender-for-identity	Microsoft Defender for Identity	(empty)	(empty)
microsoft-intune	Microsoft Intune	(empty)	(empty)
		apps	Apps
		configuration	Configuration
		developer	Developer
		education	Education
		end-user	End user help
		enrollment	Enrollment
		fundamentals	Fundamentals
		protect	Protect
		remote-actions	Remote actions
mobile-center	Mobile Center	(empty)	(empty)
mobile-engagement	Mobile Engagement	(empty)	(empty)
mobile-services	Mobile Services	(empty)	(empty)
movere	Movere	(empty)	(empty)
		movere	Movere
mse	Social Engagement	(empty)	(empty)
mssearch	Microsoft Search	(empty)	(empty)
msteams	Microsoft Teams	(empty)	(empty)
multi-factor-authentication	Multi-Factor Authentication	(empty)	(empty)
multiple	Multiple	(empty)	(empty)
mysql	Azure Database for MySQL	(empty)	(empty)
		flexible-server	Flexible Server

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		migration-guide	Migration Guide
mysql-database	MySQL	(empty)	(empty)
		single-server	Single Server
mysql-database-mc	MySQL-Mooncake	(empty)	(empty)
nat	NAT Gateway	(empty)	(empty)
network-watcher	Network Watcher	(empty)	(empty)
notification-hubs	Notification Hubs	(empty)	(empty)
o365-administration	Office 365 Administration	(empty)	(empty)
o365-proplus-itpro	Office 365 ProPlus (IT Pro)	(empty)	(empty)
O365-seccomp	O365 Security and Compliance	(empty)	(empty)
o365-solutions	Office 365 Solutions	(empty)	(empty)
odata	OData	(empty)	(empty)
office-scripts	Office Scripts	(empty)	(empty)
one-drive	One Drive	(empty)	(empty)
open-datasets	Azure Open Datasets	(empty)	(empty)
orbital	Orbital	(empty)	(empty)
partner-dashboard	Partner Center	(empty)	(empty)
		partnercenter-consumer	PartnerCenterConsumer
		partnercenter-csp	PartnerCenterCSP
		partnercenter-incentives	PartnerCenterIncentives
		partnercenter-mpn	PartnerCenterMPN
		partnercenter-sdk	PartnerCenter SDK
partner-services	Partner Services	(empty)	(empty)
peering-service	Peering Service	(empty)	(empty)
postgresql	Azure Database for PostgreSQL	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		flexible-server	Flexible Server
		hyperscale-citus	Hyperscale Citus
		migration-guide	Migration Guide
		single-server	Single Server
postgresql-database	PostgreSQL	(empty)	(empty)
powerapps	Power Apps	(empty)	(empty)
		canvas	Canvas apps
		cds	Common Data Service
		model	Model-driven apps
		pa-admin	PowerApps administrators
		pa-dev	PowerApps developers
		pa-user	PowerApps users
power-automate	Power Automate	(empty)	(empty)
		power-automate-desktop	Power Automate Desktop
powerbi	Power BI	(empty)	(empty)
		pbi-collaborate-share	Collaborate & Share
		pbi-cross-product	Cross-product integration
		pbi-dataflows	Dataflows in Power BI
		pbi-data-sources	Data sources for Power BI
		pbi-deployment	Deployment pipelines in Power BI
		pbi-enterprise	Enterprise for Power BI
		pbi-explore	Explore Power BI
		pbi-fundamentals	Fundamentals for Power BI
		pbi-natural-language	Natural Language and Q&A

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		pbi-reports-dashboards	Reports and dashboards in Power BI
		pbi-security	Security in Power BI
		pbi-transform-model	Transform, shape, and model in Power BI
		pbi-visuals	Visualizations in Power BI
		powerbi	Power BI all
		powerbi-admin	Power BI administration
		powerbi-consumer	Power BI Consumer
		powerbi-custom-visuals	Power BI Custom Visuals
		powerbi-desktop	Power BI Desktop
		powerbi-developer	Power BI developer
		powerbi-eim	Power BI Enterprise Information Management
		powerbi-gateways	Power BI gateways
		powerbi-mobile	Power BI mobile
		powerbi-premium	Power BI Premium
		powerbi-report-server	Power BI Report Server
		powerbi-service	Power BI service
		powerbi-template-apps	Power BI Template Apps
		report-builder	Power BI Report Builder
power-bi-embedded	Power BI Embedded	(empty)	(empty)
power-platform	Power Platform	(empty)	(empty)
powerquery	Power Query	(empty)	(empty)
power-virtual-agents	Power Virtual Agents	(empty)	(empty)
private-link	Azure Private Link	(empty)	(empty)
private-multi-access-edge-compute-mec	Azure private MEC	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
processrobot	ProcessRobot	(empty)	(empty)
product-insights	Product Insights	(empty)	(empty)
project-madeira	Financials	(empty)	(empty)
project-online	Project Online	(empty)	(empty)
project-operations	Project Operations	(empty)	(empty)
project-web	Project for the web	(empty)	(empty)
purview	Azure Purview	(empty)	(empty)
		purview-data-catalog	Azure Purview Data Catalog
remoteapp	RemoteApp	(empty)	(empty)
remote-connect-tool	Exchange Remote Connectivity Analyzer Tool	(empty)	(empty)
resource-graph	Azure Resource Graph	(empty)	(empty)
resource-health	Resource Health	(empty)	(empty)
resource-move	resource-move	(empty)	(empty)
resource-mover	Resource Mover	(empty)	(empty)
rights-management	Rights Management	(empty)	(empty)
role-based-access-control	Role-Based Access Control	(empty)	(empty)
		conditions	Conditions
route-server	Route Server	(empty)	(empty)
rtos	Microsoft Azure RTOS	(empty)	(empty)
scheduler	Scheduler	(empty)	(empty)
school-data-sync	School Data Sync	(empty)	(empty)
search	Azure Search	(empty)	(empty)
security	Security	(empty)	(empty)
		disk-encryption	Azure Disk Encryption
		security-develop	Security and Develop

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		security-fundamentals	Security-Fundamentals
security-center	Security Center	(empty)	(empty)
server-management	Server Management	(empty)	(empty)
service-bus	Service Bus	(empty)	(empty)
service-bus-messaging	Service Bus Messaging	(empty)	(empty)
service-bus-relay	Service Bus Relay	(empty)	(empty)
service-fabric	Service Fabric	(empty)	(empty)
service-fabric-mesh	Service Fabric Mesh	(empty)	(empty)
service-health	Service Health	(empty)	(empty)
sharepoint-online	SharePoint Online	(empty)	(empty)
sharepoint-pnp	SharePoint PnP	(empty)	(empty)
sharepoint-powershell	SharePoint PowerShell	(empty)	(empty)
signalr	SignalR Service	(empty)	(empty)
site-recovery	Site Recovery	(empty)	(empty)
		site-recovery	Site Recovery
site-reliability-engineering	Site Reliability Engineering	(empty)	(empty)
skype-for-business-online	Skype for Business Online	(empty)	(empty)
skypeforbusiness-powershell	Skype Powershell	(empty)	(empty)
spmt-powershell	SharePoint Migration Tool PowerShell	(empty)	(empty)
spring-cloud	Spring Cloud	(empty)	(empty)
sql-database	SQL Database	(empty)	(empty)
		backup-restore	Backup and restore
		business-continuity	business-continuity
		connect	Connect

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		data-movement	Data Movement
		deployment-configuration	Deployment Configuration
		development	Development
		elastic-jobs	Elastic Jobs
		elastic-pools	Elastic Pools
		high-availability	High Availability
		machine-learning	Machine Learning Services
		managed-instance	Managed Instance
		migration	Migration
		migration-guide	Migration guide
		monitor	Monitor
		operations	Operations
		performance	Performance
		replication	Replication
		scale-out	Scale Out
		scenario	Scenario
		security	Security
		service	Service
		service-overview	Service Overview
		single-database	Single Database
		sql-data-sync	SQL Data Sync
sql-database-edge	SQL Database Edge	(empty)	(empty)
		backup-restore	Backup and restore
		business-continuity	Business continuity
		data-movement	Data movement

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		development	Development
		features	Features
		high-availability	High Availability
		machine-learning	Machine Learning
		migrate	Migration
		monitor	Monitor
		performance	Performance
		security	Security
		service	Service
sql-data-warehouse	SQL Data Warehouse	(empty)	(empty)
		consume	Consume
		design	Design
		development	Development
		implement	Implement
		integration	Integration
		load-data	Load Data
		manage	Manage
		performance	Performance
		query	Query
		security	Security
		supportability	Supportability
		workload-management	Workload management
sql-db-mi	SQL DB and MI	(empty)	(empty)
		backup-restore	Backup and restore
		business-continuity	Business continuity

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		connect	Connect
		data-movement	Data movement
		deployment-configuration	Deployment Configuration
		development	Development
		features	Features
		high-availability	High Availability
		migrate	Migration
		monitor	Monitor
		performance	Performance
		security	Security
		service	Service
		service-overview	Service Overview
sql-edge	SQL Edge	(empty)	(empty)
		machine-learning	Machine Learning
sql-managed-instance	SQL Managed Instance	(empty)	(empty)
		backup-restore	Backup Restore
		business-continuity	business-continuity
		connect	Connect
		data-movement	Data Movement
		deployment-configuration	Deployment Configuration
		development	Development
		high-availability	High Availability
		instance-pools	Instance Pools
		machine-learning	Machine Learning Services
		migration	Migration

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		migration-guide	Migration guide
		monitor	Monitor
		operations	Operations
		performance	Performance
		replication	Replication
		scale-out	Scale Out
		security	Security
		service	Service
		service-overview	Service Overview
sql-server-stretch-database	SQL Server Stretch Database	(empty)	(empty)
		security	Security
		stretchDB	Stretch DB
sql-vms	Virtual Machines	(empty)	(empty)
staffhub-powershell	Staffhub Powershell	(empty)	(empty)
static-web-apps	App Service Static Web Apps	(empty)	(empty)
storage	Storage	(empty)	(empty)
		blobs	Blob storage
		common	Common
		data-lake-storage-gen2	Data Lake Storage Gen2
		disks	Disk Storage
		files	Azure Files
		partner	Partner integration
		queues	Queue storage
		tables	Table storage
storsimple	StorSimple	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
stream	Stream	(empty)	(empty)
stream-analytics	Azure Stream Analytics	(empty)	(empty)
synapse-analytics	Azure Synapse Analytics	(empty)	(empty)
		business-intelligence	Business Intelligence
		cicd	Continuous Integration/Continuous Deployment
		data-explorer	Data Explorer
		machine-learning	Machine Learning
		metadata	Metadata
		monitoring	Monitoring
		overview	Overview
		pipeline	Pipeline
		purview	Purview
		security	Security
		spark	Spark
		sql	SQL
		sql-dw	SQL DW
		synapse-link	Synapse Link
		troubleshooting	Troubleshooting
		workspace	Workspace
teams-powershell	Microsoft Teams PowerShell	(empty)	(empty)
terraform	Terraform	(empty)	(empty)
time-series-insights	IoT Time Series Insights	(empty)	(empty)
traffic-manager	Traffic Manager	(empty)	(empty)
turbos	Turbos	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
universal-print	Universal Print	(empty)	(empty)
		connector	UPConnector
		fundamentals	CoreContent
		print-devices	OemDevices
uxstypo	Typography and UX Solutions	(empty)	(empty)
virtual-desktop	Windows Virtual Desktop	(empty)	(empty)
virtual-machines	Virtual Machines	(empty)	(empty)
		automanage	Azure Automanage
		automatic-extension-upgrade	Automatic Extension Upgrade
		automatic-guest-patching	Automatic Guest Patching
		automatic-image-patching	Automatic Image Patching
		autoscale	Autoscale
		azure-hybrid-benefit	Azure Hybrid Benefit
		baremetal-infrastructure	BareMetal Infrastructure
		benchmark	Benchmark
		billing	Billing
		boot-diagnostics	Azure Boot Diagnostics
		classic-to-arm-migration	Classic to ARM Migration
		cloud-foundry	Cloud Foundry
		cloud-init	Cloud-init
		confidential-computing	Azure Confidential Computing
		dedicated-hosts	Dedicated Hosts
		disks	Disks
		extensions	Extensions

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		flexible-scale-sets	Flexible scale sets
		generation-2-vm	Generation 2 Virtual Machines
		hotpatch	Hotpatch
		hpc	HPC
		image-builder	Image Builder
		imaging	Imaging
		mainframe-rehosting	Mainframe Rehosting
		maintenance-control	Maintenance Control
		metadata-service	Metadata Service
		monitoring	Monitoring
		networking	Networking
		openshift	OpenShift
		oracle	Oracle
		proximity-placement-groups	Proximity Placement Groups
		redhat	Redhat
		reserved-instances	Reserved Instances
		scheduled-events	Scheduled Events
		security	Security
		shared-image-gallery	Shared Image Gallery
		sizes	Sizes
		spot	Azure Spot Virtual Machines
		trusted-launch	Trusted Launch
		vm-sizes-compute	Compute Optimized VM Sizes
		vm-sizes-fpga	FPGA Optimized VM Sizes

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		vm-sizes-general	General Purpose VM Sizes
		vm-sizes-gpu	GPU Optimized VM Sizes
		vm-sizes-hpc	High Performance Compute VM Sizes
		vm-sizes-memory	Memory Optimized VM Sizes
		vm-sizes-storage	Storage Optimized VM Sizes
virtual-machine-scale-sets	Virtual Machine Scale Sets	(empty)	(empty)
		automatic-guest-patching	Automatic VM Guest Patching
		automatic-instance-repairs	Automatic Instance Repairs
		automatic-os-upgrade	Automatic-Os Upgrade
		autoscale	Autoscale
		availability	Availability
		azure-hybrid-benefit	Azure Hybrid Benefit
		dedicated-hosts	Dedicated Hosts
		disks	Disks
		extensions	Extensions
		hotpatch	Hotpatch
		hpc	HPC
		imaging	Imaging
		instance-protection	Instance Protection
		maintenance-control	Maintenance Control
		management	Management
		monitoring	Monitoring
		networking	Networking
		oracle	Oracle

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		proximity-placement-groups	Proximity Placement Groups
		redhat	RedHat
		scale-in-policy	Scale-In Policy
		security	Security
		shared-image-gallery	Shared Image Gallery
		sizes	Sizes
		spot	Spot
		terminate-notification	Terminate Notification
virtual-machines-linux	Linux Virtual Machines	(empty)	(empty)
		disks	Disk storage
		extensions	Extensions
		imaging	Imaging
		monitoring	Monitoring
		networking	Networking
		recovery	Recovery
		security	Security
		sizes	Sizes
		workloads	Workloads
virtual-machines-sap	Virtual machines SAP	(empty)	(empty)
		baremetal-sap	SAP HANA on BareMetal
virtual-machines-sql	SQL Virtual Machines	(empty)	(empty)
		backup	SQL VM Backup
		deployment	SQL VM Deployment
		hadr	SQL VM HADR
		management	SQL VM Management

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
		migration	SQL VM Migration
		migration-guide	SQL VM Migration Guide
		performance	SQL VM Performance
		resource-manager	SQL VM Resource Manager
		rhel	Red Hat Enterprise Linux
		security	SQL VM Security
		service-management	SQL VM Service Management
		service-overview	Service Overview
		sql-vm	SQL VMs
		suse	SUSE Enterprise Linux
		ubuntu	Ubuntu
virtual-machines-windows	Windows Virtual Machines	(empty)	(empty)
		disks	Disk storage
		extensions	Extensions
		imaging	Imaging
		monitoring	Monitoring
		networking	Networking
		recovery	Recovery
		security	Security
		sizes	Sizes
		workloads	Workloads
virtual-network	Virtual Network	(empty)	(empty)
		ip-services	IP Services
		nat	Virtual Network NAT
virtual-wan	Virtual WAN	(empty)	(empty)

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
vmware-virtustream	Azure VMware Solution by Virtustream	(empty)	(empty)
vnf-manager	VNF- Manager	(empty)	(empty)
vpn-gateway	VPN Gateway	(empty)	(empty)
vs-appcenter	App Center	(empty)	(empty)
		app-center-analytics	Analytics
		app-center-auth	Auth
		app-center-build	Build
		app-center-data	Data
		app-center-diagnostics	Diagnostics
		app-center-distribute	Distribute
		app-center-general	General
		app-center-push	Push
		app-center-test	Test
		app-center-transition	Transition
vs-appcenter-sdk	App Center SDK	(empty)	(empty)
		app-center-analytics	Analytics
		app-center-auth	Auth
		app-center-build	Build
		app-center-data	Data
		app-center-diagnostics	Diagnostics
		app-center-distribute	Distribute
		app-center-general	General
		app-center-push	Push
		app-center-test	Test
		app-center-transition	Transition

MS.SERVICE SLUG	MS.SERVICE LABEL	MS.SUBSERVICE SLUG	MS.SUBSERVICE LABEL
web-app-container	Web App for Containers	(empty)	(empty)
web-application-firewall	Web Application Firewall	(empty)	(empty)
websites	Websites	(empty)	(empty)
web-sites	Websites	(empty)	(empty)
whiteboard-powershell	Whiteboard PowerShell	(empty)	(empty)
windows-vms	Virtual Machines	(empty)	(empty)
yammer	Yammer	(empty)	(empty)

ms.topic

The `ms.topic` value indicates the type of content.

To request a new ms.topic value, email docsmetamanager@microsoft.com.

For more details, see the full taxonomy Detail View.

SLUG
archived
article
best-practice
callback
checklist
class
conceptual
contributor-guide
design-pattern
end-user-help
enum
enumeration
error-reference
example-scenario

SLUG
function
github-sample
guide
how-to
hub-page
include
interactive-tutorial
interface
ioctl
kb-support
landing-page
language-reference
macro
managed-reference
method
overview
portal
quickstart
reference
reference-architecture
retired
sample
struct
structure
troubleshooting

SLUG
tutorial

Docs Markdown reference

6/1/2021 • 18 minutes to read

> **TIP**
>
> You may refer external docs contributors to this public contributor guide page for similar information. Keep the information in sync between these two pages as things change.

This article provides an alphabetical reference for writing Markdown for docs.microsoft.com (Docs).

Markdown is a lightweight markup language with plain text formatting syntax. Docs supports CommonMark compliant Markdown parsed through the Markdig parsing engine. Docs also supports custom Markdown extensions that provide richer content on the Docs site.

You can use any text editor to write Markdown, but we recommend Visual Studio Code with the Docs Authoring Pack. The Docs Authoring Pack provides editing tools and preview functionality that lets you see what your articles will look like when rendered on Docs.

Alerts (Note, Tip, Important, Caution, Warning)

Alerts are a Markdown extension to create block quotes that render on docs.microsoft.com with colors and icons that indicate the significance of the content. The following alert types are supported:

```
> [!NOTE]
> Information the user should notice even if skimming.

> [!TIP]
> Optional information to help a user be more successful.

> [!IMPORTANT]
> Essential information required for user success.

> [!CAUTION]
> Negative potential consequences of an action.

> [!WARNING]
> Dangerous certain consequences of an action.
```

These alerts look like this on docs.microsoft.com:

> **NOTE**
>
> Information the user should notice even if skimming.

> **TIP**
>
> Optional information to help a user be more successful.

> **IMPORTANT**
>
> Essential information required for user success.

Caution

Negative potential consequences of an action.

> **WARNING**
>
> Dangerous certain consequences of an action.

Angle brackets

If you use angle brackets in text in your file--for example, to denote a placeholder--you need to manually encode the angle brackets. Otherwise, Markdown thinks that they're intended to be an HTML tag.

For example, encode `<script name>` as `<script name>` or `\<script name>`.

Angle brackets don't have to be escaped in text formatted as inline code or in code blocks.

Apostrophes and quotation marks

If you copy from Word into a Markdown editor, the text might contain "smart" (curly) apostrophes or quotation marks. These need to be encoded or changed to basic apostrophes or quotation marks. Otherwise, you end up with things like this when the file is published: Itâ€™s

Here are the encodings for the "smart" versions of these punctuation marks:

- Left (opening) quotation mark: `“`
- Right (closing) quotation mark: `”`
- Right (closing) single quotation mark or apostrophe: `’`
- Left (opening) single quotation mark (rarely used): `‘`

> **TIP**
>
> To avoid "smart" characters in your markdown files, rely on the Docs Authoring Pack - smart quote replacement feature. For more information, see smart quote replacement.

Blockquotes

Blockquotes are created using the `>` character:

```
> This is a blockquote. It is usually rendered indented and with a different background color.
```

The preceding example renders as follows:

> This is a blockquote. It is usually rendered indented and with a different background color.

Bold and italic text

To format text as **bold**, enclose it in two asterisks:

```
This text is **bold**.
```

To format text as *italic*, enclose it in a single asterisk:

```
This text is *italic*.
```

To format text as both ***bold and italic***, enclose it in three asterisks:

```
This text is both ***bold and italic***.
```

Code snippets

Docs Markdown supports the placement of code snippets both inline in a sentence and as a separate "fenced" block between sentences. For more information, see How to add code to docs.

Columns

The **columns** Markdown extension gives Docs authors the ability to add column-based content layouts that are more flexible and powerful than basic Markdown tables, which are only suited for true tabular data. You can add up to four columns, and use the optional `span` attribute to merge two or more columns.

The syntax for columns is as follows:

```
:::row:::
   :::column span="":::
      Content...
   :::column-end:::
   :::column span="":::
      More content...
   :::column-end:::
:::row-end:::
```

Columns should only contain basic Markdown, including images. Headings, tables, tabs, and other complex structures shouldn't be included. A row can't have any content outside of column.

For example, the following Markdown creates one column that spans two column widths, and one standard (no `span`) column:

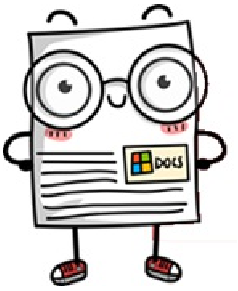
```
:::row:::
   :::column span="2":::
      **This is a 2-span column with lots of text.**

      Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vestibulum mollis nunc
      ornare commodo. Nullam ac metus imperdiet, rutrum justo vel, vulputate leo. Donec
      rutrum non eros eget consectetur.
   :::column-end:::
   :::column span="":::
      **This is a single-span column with an image in it.**

      ![Doc.U.Ment](media/markdown-reference/document.png)
   :::column-end:::
:::row-end:::
```

This renders as follows:

This is a 2-span column with lots of text.

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec vestibulum mollis nunc ornare commodo. Nullam ac metus imperdiet, rutrum justo vel, vulputate leo. Donec rutrum non eros eget consectetur.

This is a single-span column with an image in it.



Headings

Docs supports six levels of Markdown headings:

```
# This is a first level heading (H1)

## This is a second level heading (H2)

...

###### This is a sixth level heading (H6)
```

- There must be a space between the last `#` and heading text.
- Each Markdown file must have one and only one H1 heading.
- The H1 heading must be the first content in the file after the YML metadata block.
- H2 headings automatically appear in the right-hand navigating menu of the published file. Lower-level headings don't appear, so use H2s strategically to help readers navigate your content.
- HTML headings, such as `<h1>`, aren't recommended, and in some cases will cause build warnings.
- You can link to individual headings in a file via bookmarks.

HTML

Although Markdown supports inline HTML, HTML isn't recommended for publishing to Docs, and except for a limited list of values will cause build errors or warnings.

For more information, see HTML allowlist in the Docs Admin Guide.

Images

The following file types are supported by default for images:

- .jpg
- .png

To support additional image types, such as .gif, you must add them as resources in docfx.json:

```
"resource": [
  {
    "files" : [
      "**/*.png",
      "**/*.jpg,
      "**/*.gif"
    ],
```

Standard conceptual images (default Markdown)

The basic Markdown syntax to embed an image is:

```
![<alt text>](<folderPath>)

Example:
![alt text for image](../images/Introduction.png)
```

Where `<alt text>` is a brief description of the image and `<folder path>` is a relative path to the image. Alternate text is required for screen readers for the visually impaired. It's also useful if there's a site bug where the image can't render.

Underscores in alt text aren't rendered properly unless you escape them by prefixing them with a backslash (`\_`). However, don't copy file names for use as alt text. For example, instead of this:

```
![ADextension_2FA_Configure_Step4](./media/bogusfilename/ADextension_2FA_Configure_Step4.PNG)
```

Write this:

```
![Active Directory extension for two-factor authentication, step 4: Configure]
(./media/bogusfilename/ADextension_2FA_Configure_Step4.PNG)
```

Standard conceptual images (Docs Markdown)

The Docs custom `:::image:::` extension supports standard images, complex images, and icons.

For standard images, the older Markdown syntax will still work, but the new extension is recommended because it supports more powerful functionality, such as specifying a localization scope that's different from the parent topic. Other advanced functionality, such as selecting from the shared image gallery instead of specifying a local image, will be available in the future. The new syntax is as follows:

```
:::image type="content" source="<folderPath>" alt-text="<alt text>":::
```

If `type="content"` (the default), both `source` and `alt-text` are required.

Complex images with long descriptions

You can also use this extension to add an image with a long description that is read by screen readers but not rendered visually on the published page. Long descriptions are an accessibility requirement for complex images, such as graphs. The syntax is the following:

```
:::image type="complex" source="<folderPath>" alt-text="<alt text>":::
   <long description here>
:::image-end:::
```

If `type="complex"` , `source` , `alt-text` , a long description, and the `:::image-end:::` tag are all required.

Automatic borders

The `:::image:::` extension also supports the `border` property, which automatically adds a 1-pixel gray border around your image. This is the equivalent of the `mx-imgBorder` style described in Gray borders. The `border` property is `true` by default for `content` and `complex` images, so you'll get the border automatically unless you explicitly add the property with a value of `false`. The `border` property is `false` by default for `icon` images.

The `border` property is the recommended way to add a border - please don't create your own borders manually.

Creating an expandable image

The optional `lightbox` property allows you to create an expanded image, as described in Create an expandable screenshot (lightbox). The value of `lightbox` is the path to the expanded image.

Specifying loc-scope

Sometimes the localization scope for an image is different from that of the article or module that contains it. This can cause a bad global experience: for example, if a screenshot of a product is accidentally localized into a language the product isn't available in. To prevent this, you can specify the `loc-scope` attribute. The `loc-scope` attribute can be specified on any image of types `content` and `complex`, and is *required* for screenshots that show a product with a different localization scope than the article or module that contains it.

For more information, see How to specify the localization scope for an image.

Icons

The image extension supports icons, which are decorative images and should not have alt text. The syntax for icons is:

```
:::image type="icon" source="<folderPath>":::
```

If `type="icon"`, `source` should be specified but `alt-text` shouldn't be.

The `border` property is `false` by default for icons. If your decorative image requires the standard image border, explicitly add `border="true"` to the `:::image:::` tag.

The `lightbox` property works the same for `icon` images as for standard `content` images.

Other image file types

You can add support for other image types by adding them as resources to the docfx.json file for your docset. For example, add .gif to enable animated .gif files.

For more information about creating and using images, see Create a screenshot, Create an expandable screenshot, Create conceptual art, and Create an animated GIF.

Included Markdown files

Where markdown files need to be repeated in multiple articles, you can use an include file. The includes feature instructs Docs to replace the reference with the contents of the include file at build time. You can use includes in the following ways:

- Inline: Reuse a common text snippet inline with within a sentence.
- Block: Reuse an entire Markdown file as a block, nested within a section of an article.

An inline or block include file is a Markdown (.md) file. It can contain any valid Markdown. Include files are typically located in a common *includes* subdirectory, in the root of the repository. When the article is published, the included file is seamlessly integrated into it.

Includes syntax

Block include is on its own line:

```
[!INCLUDE [<title>](<filepath>)]
```

Inline include is within a line:

```
Text before [!INCLUDE [<title>](<filepath>)] and after.
```

Where `<title>` is the name of the file and `<filepath>` is the relative path to the file. `INCLUDE` must be capitalized and there must be a space before the `<title>`.

Here are requirements and considerations for include files:

- Use block includes for significant amounts of content--a paragraph or two, a shared procedure, or a shared section. Do not use them for anything smaller than a sentence.
- Includes won't be rendered in the GitHub rendered view of your article, because they rely on Docs extensions. They'll be rendered only after publication.
- Write all the text in an include file in complete sentences or phrases that don't depend on preceding or following text in the article that references the include. Ignoring this guidance creates an untranslatable string in the article.
- Don't embed include files within other include files.
- `/Includes` folders are excluded from build. Therefore, images stored in `/includes` folders and referenced in included files won't be displayed in published content. Store images in a `/media` folder outside the `/includes` folder.
- As with regular articles, don't share media between include files. Use a separate file with a unique name for each include and article. Store the media file in the media folder that's associated with the include.
- Don't use an include as the only content of an article. Includes are meant to be supplemental to the content in the rest of the article.

For more information, see Include reusable content in articles.

Links

For information on syntax for links, see Links.

Lists (Numbered, Bulleted, Checklist)

Numbered list

To create a numbered list, you can use all 1s. The numbers are rendered in ascending order as a sequential list when published. For increased source readability, you can increment your lists manually.

Don't use letters in lists, including nested lists. They don't render correctly when published to Docs. Nested lists using numbers will render as lowercase letters when published. For example:

```
1. This is
1. a parent numbered list
   1. and this is
   1. a nested numbered list
1. (fin)
```

This renders as follows:

1. This is
2. a parent numbered list
   a. and this is
   b. a nested numbered list
3. (fin)

Bulleted list

To create a bulleted list, use `-` or `*` followed by a space at the beginning of each line:

```
- This is
- a parent bulleted list
  - and this is
  - a nested bulleted list
- All done!
```

This renders as follows:

- This is
- a parent bulleted list
  - and this is
  - a nested bulleted list
- All done!

Whichever syntax you use, `-` or `*`, use it consistently within an article.

Checklist

Checklists are available for use on Docs via a custom Markdown extension:

```
> [!div class="checklist"]
> * List item 1
> * List item 2
> * List item 3
```

This example renders on Docs like this:

- List item 1
- List item 2
- List item 3

Use checklists at the beginning or end of an article to summarize "What will you learn" or "What have you learned" content. Do not add random checklists throughout your articles.

Next step action

You can use a custom extension to add a next step action button to Docs pages.

The syntax is as follows:

```
> [!div class="nextstepaction"]
> [button text](link to topic)
```

For example:

```
> [!div class="nextstepaction"]
> [Learn about adding code to articles](code-in-docs.md)
```

This renders as follows:

Learn about adding code to articles

You can use any supported link in a next step action, including a Markdown link to another web page. In most cases, the next action link will be a relative link to another file in the same docset.

Non-localized strings

You can use the custom `no-loc` Markdown extension to identify strings of content that you would like the localization process to ignore.

All strings called out will be case-sensitive; that is, the string must match exactly to be ignored for localization.

To mark an individual string as non-localizable, use this syntax:

```
:::no-loc text="String":::
```

For example, in the following, only the single instance of `Document` will be ignored during the localization process:

```
# Heading 1 of the Document

Markdown content within the :::no-loc text="Document":::.  The are multiple instances of Document, document,
and documents.
```

> **NOTE**
>
> Use `\` to escape special characters:
>
> ```
> Lorem :::no-loc text="Find a \"Quotation\""::: Ipsum.
> ```

You can also use metadata in the YAML header to mark all instances of a string within the current Markdown file as non-localizable:

```
author: cillroy
no-loc: [Global, Strings, to be, Ignored]
```

> **NOTE**
>
> The no-loc metadata is not supported as global metadata in *docfx.json* file. The localization pipeline doesn't read the *docfx.json* file, so the no-loc metadata must be added into each individual source file.

In the following example, both in the metadata `title` and the Markdown header the word `Document` will be ignored during the localization process.

In the metadata `description` and the Markdown main content the word `document` is localized, because it does not start with a capital `D`.

```
---
title: Title of the Document
author: author-name
description: Description for the document
no-loc: [Title, Document]
---
# Heading 1 of the Document

Markdown content within the document.
```

Selectors

Selectors are UI elements that let the user switch between multiple flavors of the same article. They are used in some doc sets to address differences in implementation across technologies or platforms. Selectors are typically most applicable to our mobile platform content for developers.

Because the same selector Markdown goes in each article file that uses the selector, we recommend placing the selector for your article in an include file. Then you can reference that include file in all your article files that use the same selector.

There are two types of selectors: a single selector and a multi-selector.

Single selector

```
> [!div class="op_single_selector"]
> - [Universal Windows](../articles/notification-hubs-windows-store-dotnet-get-started/)
> - [Windows Phone](../articles/notification-hubs-windows-phone-get-started/)
> - [iOS](../articles/notification-hubs-ios-get-started/)
> - [Android](../articles/notification-hubs-android-get-started/)
> - [Kindle](../articles/notification-hubs-kindle-get-started/)
> - [Baidu](../articles/notification-hubs-baidu-get-started/)
> - [Xamarin.iOS](../articles/partner-xamarin-notification-hubs-ios-get-started/)
> - [Xamarin.Android](../articles/partner-xamarin-notification-hubs-android-get-started/)
```

... will be rendered like this:

Multi-selector

```
> [!div class="op_multi_selector" title1="Platform" title2="Backend"]
> - [(iOS | .NET)](./mobile-services-dotnet-backend-ios-get-started-push.md)
> - [(iOS | JavaScript)](./mobile-services-javascript-backend-ios-get-started-push.md)
> - [(Windows universal C# | .NET)](./mobile-services-dotnet-backend-windows-universal-dotnet-get-started-push.md)
> - [(Windows universal C# | Javascript)](./mobile-services-javascript-backend-windows-universal-dotnet-get-started-push.md)
> - [(Windows Phone | .NET)](./mobile-services-dotnet-backend-windows-phone-get-started-push.md)
> - [(Windows Phone | Javascript)](./mobile-services-javascript-backend-windows-phone-get-started-push.md)
> - [(Android | .NET)](./mobile-services-dotnet-backend-android-get-started-push.md)
> - [(Android | Javascript)](./mobile-services-javascript-backend-android-get-started-push.md)
> - [(Xamarin iOS | Javascript)](./partner-xamarin-mobile-services-ios-get-started-push.md)
> - [(Xamarin Android | Javascript)](./partner-xamarin-mobile-services-android-get-started-push.md)
```

... will be rendered like this:

Subscript and superscript

You should only use subscript or superscript when necessary for technical accuracy, such as when writing about mathematical formulas. Don't use them for non-standard styles, such as footnotes.

For both subscript and superscript, use HTML:

```
Hello <sub>This is subscript!</sub>
```

This renders as follows:

Hello $_{\text{This is subscript!}}$

```
Goodbye <sup>This is superscript!</sup>
```

This renders as follows:

Goodbye $^{\text{This is superscript!}}$

Tabbed conceptual

Tabbed conceptual is a Markdown extension for docs.microsoft.com that allows us to present different versions of content, such as procedural steps to accomplish the same task on different platforms, in a tabbed format. See Tabbed Conceptual in the Docs Admin Guide for more information.

Tables

The simplest way to create a table in Markdown is to use pipes and lines. To create a standard table with a header, follow the first line with dashed line:

```
|This is   |a simple   |table header|
|----------|-----------|------------|
|table     |data       |here        |
|it doesn't|actually   |have to line up nicely!|
```

This renders as follows:

THIS IS	A SIMPLE	TABLE HEADER
table	data	here
it doesn't	actually	have to line up nicely!

You can align the columns by using colons:

```
| Fun                  | With                 | Tables          |
| :------------------- | -------------------: |:---------------:|
| left-aligned column  | right-aligned column | centered column |
| $100                 | $100                 | $100            |
| $10                  | $10                  | $10             |
| $1                   | $1                   | $1              |
```

The Markdown above renders as follows:

FUN	WITH	TABLES
left-aligned column	right-aligned column	centered column

FUN	WITH	TABLES
$100	$100	$100
$10	$10	$10
$1	$1	$1

> **TIP**
>
> The Docs Authoring Extension for VS Code makes it easy to add basic Markdown tables!
>
> You can also use an online table generator.

Line breaks within words in any table cell

Long words in a Markdown table might make the table expand to the right navigation and become unreadable. You can solve that by allowing Docs rendering to automatically insert line breaks within words when needed. Just wrap up the table with the custom class `[!div class="mx-tdBreakAll"]`.

Here is a Markdown sample of a table with three rows that will be wrapped by a `div` with the class name `mx-tdBreakAll`.

```
> [!div class="mx-tdBreakAll"]
> |Name|Syntax|Mandatory for silent installation?|Description|
> |-------------|----------|---------|---------|
> |Quiet|/quiet|Yes|Runs the installer, displaying no UI and no prompts.|
> |NoRestart|/norestart|No|Suppresses any attempts to restart. By default, the UI will prompt before restart.|
> |Help|/help|No|Provides help and quick reference. Displays the correct use of the setup command, including a list of all options and behaviors.|
```

It will be rendered like this:

NAME	SYNTAX	MANDATORY FOR SILENT INSTALLATION?	DESCRIPTION
Quiet	/quiet	Yes	Runs the installer, displaying no UI and no prompts.
NoRestart	/norestart	No	Suppresses any attempts to restart. By default, the UI will prompt before restart.
Help	/help	No	Provides help and quick reference. Displays the correct use of the setup command, including a list of all options and behaviors.

Line breaks within words in second column table cells

You might want line breaks to be automatically inserted within words only in the second column of a table. To limit the breaks to the second column, apply the class `mx-tdCol2BreakAll` by using the `div` wrapper syntax as shown earlier.

Inconsistent column widths between tables

You may notice that the column widths of the tables in your articles look odd or inconsistent. This behavior occurs because the length of text within the cells determines the layout of the table. Unfortunately, there's no way to control how the tables render. This is a limitation of Markdown. Even though it would look nicer to have the width of table columns be consistent, this would have some disadvantages too:

- Interlacing HTML code with Markdown makes topics more complicated and discourages community contributions.
- A table that you make look good for a specific screen size may end up looking completely unreadable at different screen sizes as it preempts responsive rendering.

Data matrix tables

A data matrix table has both a header and a weighted first column, creating a matrix with an empty cell in the top left. Docs has custom Markdown for data matrix tables:

```
|                  |Header 1 |Header 2|
|------------------|---------|--------|
|**First column A**|Cell 1A  |Cell 2A |
|**First column B**|Cell 1B  |Cell 2B |
```

The example renders as:

	HEADER 1	HEADER 2
First column A	Cell 1A	Cell 2A
First column B	Cell 1B	Cell 2B

Every entry in the first column must be styled as bold (`**bold**`); otherwise the tables won't be accessible for screen readers or valid for Docs.

> **TIP**
> The Docs Authoring Pack for VS Code includes a function to convert a regular Markdown table into a data matrix table. Just select the table, right-click, and select **Convert to data matrix table**.

HTML Tables

HTML tables aren't recommended for docs.microsoft.com. They aren't human readable in the source - which is a key principle of Markdown.

Videos

For more information, see Embedding video in your content.

How to include code in docs

4/13/2021 • 18 minutes to read

> **TIP**
>
> You may refer external docs contributors to this public contributor guide page for similar information. Keep the information in sync between these two pages as things change.

There are several ways to include code in an article published on docs.microsoft.com:

- Individual elements (words) within a line.

  Here's an example of `code` style.

  Use code format when referring to named parameters and variables in a nearby code block in your text. Code format may also be used for properties, methods, classes, and language keywords. For more information, see Code elements later in this article.

- Code blocks in the article Markdown file.

  ````
  ```csharp
  public static void Log(string message)
  {
      _logger.LogInformation(message);
  }
  ```
  ````

  Use inline code blocks when it's impractical to display code by reference to a code file. For more information, see Code blocks later in this article.

- Code blocks by reference to a code file in the local repository.

  ```
  :::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26":::
  ```

  For more information, see In-repo snippet references later in this article.

- Code blocks by reference to a code file in another repository.

  ```
  :::code language="csharp" source="~/samples-durable-functions/samples/csx/shared/Location.csx" highlight="2,5":::
  ```

  For more information, see Out-of-repo snippet references later in this article.

- Code blocks that let the user run code in the browser.

  ```
  :::code source="PowerShell.ps1" interactive="cloudshell-powershell":::
  ```

  For more information, see Interactive code snippets later in this article.

Besides explaining the Markdown for each of these ways to include code, the article provides some general guidance for all code blocks.

Code elements

A "code element" is a programming language keyword, class name, property name, and so forth. It's not always obvious what qualifies as code. For example, NuGet package names should be treated as code. When in doubt, see Text formatting guidelines.

Inline code style

To include a code element in article text, surround it with backticks (`) to indicate code style. Inline code style shouldn't use the triple-backtick format.

MARKDOWN	RENDERED
By default, the Entity Framework interprets a property that's named `Id` or `ClassnameID` as the primary key.	By default, the Entity Framework interprets a property that's named `Id` or `ClassnameID` as the primary key.

When an article is localized (translated into other languages), text styled as code is left untranslated. If you want to prevent localization without using code style, see Non-localized strings.

Bold style

Some older style guides specify bold for inline code. Bold is an option when code style is so obtrusive as to compromise readability. For example, a Markdown table with mostly code elements might look too busy with code styling everywhere. If you choose to use bold style, use non-localized strings syntax to make sure that code is not localized.

Links

A link to reference documentation may be more helpful than code format in some contexts. If you use a link, you can apply code format to the link text, but there's no difference in the rendered appearance compared to link text that isn't styled as code.

If you use a link and refer to the same element later in the same context, make the subsequent instances code format rather than links.

Placeholders

If you want the user to replace a section of displayed code with their own values, use placeholder text marked off by angle brackets. For example:

```
az group delete -n <ResourceGroupName>
```

You may note that the brackets must be removed when substituting real values. The Microsoft Writing Style Guide calls for italics, which you may format within angle bracketed inline code:

> `<ResourceGroupName>` is the resource group where...

Curly braces { } are discouraged for use as syntactical placeholders. They may be confused with the same notation used in replaceable text, format strings, string interpolation, text templates, and similar programming constructs.

Placeholder names can be separated by hyphens ("kebab case"), with underscores, or not separated at all (Pascal case). Kebab case may generate syntax errors and underscores may conflict with underlining. All-caps may conflict with named constants in many languages, though it may also draw attention to the placeholder name.

> `<Resource-Group-Name>` or `<ResourceGroupName>`

Code blocks

The syntax for including code in a doc depends on where the code is located:

- In the article Markdown file
- In a code file in the same repository
- In a code file in a different repository

Following are guidelines that apply to all types of code blocks:

- Screenshots
- Automate code validation.
- Highlight key lines of code.
- Avoid horizontal scroll bars.

Screenshots

All of the methods listed in the preceding section result in usable code blocks:

- You can copy and paste from them.
- They're indexed by search engines.
- They're accessible to screen readers.

These are just a few of the reasons why IDE screenshots aren't recommended as a method of including code in an article. Use IDE screenshots for code only if you're showing something about the IDE itself, like IntelliSense. Don't use screenshots just to show colorization and highlighting.

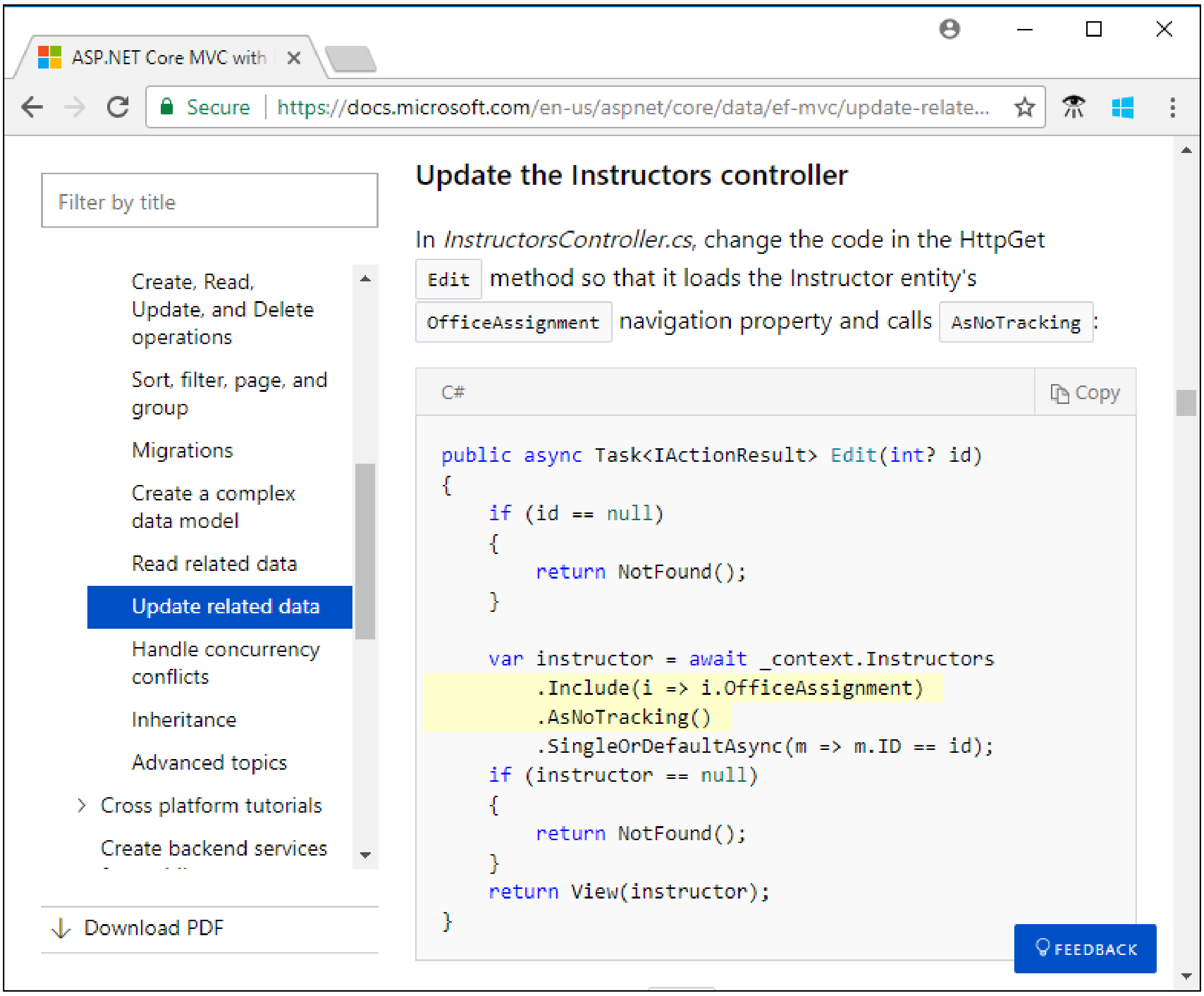
Code validation

Some repositories have implemented processes that automatically compile all sample code to check for errors. The .NET repository does this. For more information, see Learn how to contribute to the .NET docs repositories.

If you are including code blocks from another repository, work with the owners on a maintenance strategy for the code so that your included code does not break or go out of date as new versions of the libraries the code uses are shipped.

Highlighting

Snippets typically include more code than necessary in order to provide context. It helps readability when you highlight the key lines that you're focusing on in the snippet, as in this example:



You can't highlight code when you include it in the article Markdown file. It works only for code snippets included by reference to a code file.

Horizontal scroll bars

Break up long lines to avoid horizontal scroll bars. Scroll bars on code blocks make code hard to read. They're especially problematic on longer code blocks, where it may be impossible to see the scroll bar and the line you want to read at the same time.

A good practice for minimizing horizontal scroll bars on code blocks is to break up code lines longer than 85 characters. But keep in mind that the presence or absence of a scroll bar isn't the only criterion of readability. If breaking a line before 85 hurts readability or copy-paste convenience, feel free to go over 85.

Inline code blocks

Use inline code blocks only when it's impractical to display code by reference to a code file. Inline code is more difficult to test and keep up to date compared to a code file that is part of a complete project. And inline code may omit context that could help the developer to understand and use the code. These considerations apply mainly to programming languages. Inline code blocks can also be used for outputs and inputs (such as JSON), query languages (such as SQL), and scripting languages (such as PowerShell).

There are two ways to indicate a section of text in an article file is a code block: by *fencing* it in triple-backticks (```) or triple tildes (~~~), or by indenting it. Fencing is preferred because it lets you specify the language. Avoid using indentation because it's too easy to get wrong and it may be hard for another writer to understand your intent when they need to edit your article.

Language indicators are placed immediately after the opening triple-backticks or triple-tildes, as in the following example:

Markdown:

````
```json
{
    "aggregator": {
        "batchSize": 1000,
        "flushTimeout": "00:00:30"
    }
}
```
````

Rendered:

```json
{
    "aggregator": {
        "batchSize": 1000,
        "flushTimeout": "00:00:30"
    }
}
```

> **TIP**
>
> GitHub Flavored Markdown supports delimiting code blocks with tildes (~) as well as with backticks (`). The symbol used to open and close the code block must be consistent within the same code block.

For information about the values you can use as language indicators, see Supported languages.

If you use a language or environment word after the triple-backticks (```) that isn't supported, that word appears in the code section title bar on the rendered page. Whenever possible, use a language or environment indicator in your inline code blocks.

> **NOTE**
>
> If you copy and paste code from a Word document, make sure it has no "curly quotes," which aren't valid in code. If it does, change them back to normal quotes (`'` and `"`). Alternatively, rely on the Docs Authoring Pack, smart quotes replacement feature.

In-repo snippet references

The preferred way to include code snippets for programming languages in docs is by reference to a code file. This method gives you the ability to highlight lines of code and makes the wider context of the snippet available on GitHub for developers to use. You can include code by using the triple colon format (:::) either manually or in Visual Studio Code with the help of the docs.microsoft.com Authoring Pack.

1. In Visual Studio Code, click **Alt + M** or **Option + M** and select Snippet.
2. Once Snippet is selected, you will be prompted for Full Search, Scoped Search, or Cross-Repository Reference. To search locally, select Full Search.
3. Enter a search term to find the file. Once you've found the file, select the file.
4. Next, select an option to determine which line(s) of code should be included in the snippet. The options are: **ID**, **Range**, and **None**.
5. Based on your selection from Step 4, provide a value if necessary.

Display entire code file:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs":::
```

Display part of a code file by specifying line numbers:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26":::
```

Display part of a code file by specifying a snippet name:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" id="snippet_Create":::
```

The following sections explain these examples:

- Use a relative path to the code file
- Include only selected line numbers
- Refer to a named snippet
- Highlight selected lines

For more information, see Snippet syntax reference later in this article.

Path to code file

Example:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26":::
```

The example is from the ASP.NET docs repo, aspnetcore/data/ef-mvc/crud.md article file. The code file is referenced by a relative path to aspnetcore/data/ef-mvc/intro/samples/cu/Controllers/StudentsController.cs in the same repository.

The preceding example uses the folder structure of the ASP.NET Core docs repository. If your repo doesn't already have guidelines for where to locate code files that you access through snippet references, use the folder structure recommended for the .NET docs:

- In the same folder as the article that shows the snippets, create a *snippets* folder.
- In the *snippets* folder, create a folder that has the same name as the article file, but without the *.md* extension. If snippet files are shared among two or more articles in the same folder, name the folder *shared* instead of using an article file name.
- If your doc set shows multiple languages for the same snippets, create a folder for each language. Name the folders using the language code used to identify snippets. For example, language-specific folders could be named *csharp* and *vb*.
- Put the code file that has snippets in the folder named for a language. If language-named folders aren't used, put the code file in the folder that is named after the article or *shared*.

For an example, see Example folder structure.

Selected line numbers

Example:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26":::
```

This example displays only lines 2-24 and 26 of the *StudentController.cs* code file.

Prefer named snippets over hard-coded line numbers, as explained in the next section.

Named snippet

Example:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" id="Create":::
```

Use only letters and underscores for the name.

The example displays the `Create` section of the code file. The code file for this example has snippet tags in comments in the C# code:

```
// code excluded from the snippet
// <Create>
// code included in the snippet
// </Create>
// code excluded from the snippet
```

Named code snippets can be nested, as shown in the following example:

```
// <Method>
public static SomeMethod()
{
    // <Line>
    // Single line of code.
    // </Line>
}
// </Method>
```

When the `snippet_Method` code snippet is rendered, the `snippet_Line` tags aren't included in the rendered output.

Whenever you can, refer to a named section rather than specifying line numbers. Line number references are brittle because code files inevitably change in ways that make line numbers change. You don't necessarily get notified of such changes. Your article eventually starts showing the wrong lines and you have no clue anything has changed.

Highlighting selected lines

Example:

```
:::code language="csharp" source="intro/samples/cu/Controllers/StudentsController.cs" range="2-24,26"
highlight="2,5":::
```

The example highlights lines 2 and 5, counting from the start of the displayed snippet. Line numbers to highlight don't count from the start of the code file. In other words, lines 3 and 6 of the code file are highlighted.

Out-of-repo snippet references

If the code file you want to reference is in a different repository, set up the code repository as a *dependent repository*. When you do that, you specify a name for it. That name then acts like a folder name for purposes of code references.

Dependent repositories metadata

The data structure to set up dependent repositories for cross-repo references (CRRs) is a list that contains path, repository, and branch information. The following `dependent_repositories` list property is required in the *.openpublishing.publish.config.json* file.

```json
{
    "dependent_repositories": [
        {
            "path_to_root": "<relative path to repository root>",
            "url": "<referenced repository url>",
            "branch": "<branch name of referenced repository>",
            "branch_mapping": {
                "<source repository branch>": "<referenced repository branch>",
                "<source repository branch>": "<referenced repository branch>"
            }
        },
        {
            "path_to_root": "token",
            "url": "https://github.com/Microsoft/token",
            "branch": "master",
            "branch_mapping": {
                "master": "master",
                "develop": "test"
            }
        }
    ]
}
```

METADATA	MEANING	PARAMETER REQUIRED
dependent_repositories	The CRR relationship list name	Yes
path_to_root	A relative folder path to repository root. The folder doesn't have to have been created yet.	Yes
url	URL of the reference repository that the current repository is dependent on.	Yes
branch	The branch of the reference repository that you want to use when doing builds.	Yes
branch_mapping	**Advanced feature**: A branch map of current repository and reference repository. Key is current repository's branch and value is reference repository's branch. This property allows you to use different reference branches when you build source from different branches of the current repository. **If the source build branch isn't specified in this property, the `branch` value will be used.**	No

For sample JSON that you can add to the *.openpublishing.publish.config.json* file and reference in CRR snippets, see Cross Repository Reference.

Snippet reference syntax

You use the same triple-colon snippet syntax for CRRs as for in-repo snippets. Only the path to the snippet is different.

A path to a CRR snippet must either start with a tilde (`~`) if it's relative, or else it must path out of the repo with

`..` syntax. The tilde signifies the root folder. For example, in the SQL repository, `~` represents "docs". To point to the repository root, use `~/..`. `..` denotes the parent directory one level back, or 'up,' relative to the current directory. This sequence can be used multiple times, for example `../../` denotes a directory 2 levels back. If a relative path requires several levels of backing out, consider the complexity of the path and the user experience; you may decide to use a different, friendlier link syntax.

Note that relative file paths cannot be directly copied and pasted because the folder they point to is relative to your working directory. If your environment differs from the source code environment, the path will be invalid.

The following snippet reference uses a relative file path:

`~/../xamarin-forms-samples/WebServices/TodoREST/TodoAPI/TodoAPI/Startup.cs`

> **NOTE**
> The name you assign to the dependent repository is relative to the root of the main repository, but the tilde (~) refers to the root of the doc set. The doc set root is determined by `build_source_folder` in *.openpublishing.publish.config.json*. The path to the snippet in the preceding example is correct for the AspNetCore.Docs repo because `build_source_folder` in that repository's *.openpublishing.publish.config.json* file is set to the *aspnetcore* folder.
>
> ```
> "build_source_folder": "aspnetcore"
> ```
>
> Suppose the AspNetCore.Docs repo had the following setting in its *.openpublishing.publish.config.json* file:
>
> ```
> "build_source_folder": "."
> ```
>
> In that case, the path would start with `~/xamarin-forms-samples` instead of `~/../xamarin-forms-samples`.

> **IMPORTANT**
> Updating an external code snippet won't automatically trigger a content build. You need to trigger the build by either changing something in the doc repo or manually starting a build.

Snippets in a Jupyter notebook

You can reference a cell in a Jupyter notebook as a code snippet. In order to reference the cell:

1. Add cell metadata to the notebook for the cells you wish to reference.
2. Set up access to the repository.
3. Use Jupyter notebook snippet syntax in your markdown file.

Add metadata to notebook

1. Name the cell by adding cell metadata in the Jupyter notebook.
   - In Jupyter, you can edit cell metadata by first enabling the cell toolbar: **View > Cell Toolbar > Edit Metadata**.
   - Once the cell toolbar is enabled, select **Edit Metadata** on the cell you wish to name.
   - Or you can edit metadata directly in the notebook's JSON structure.
2. In the cell metadata, add a "name" attribute:

   ```
   "metadata": {"name": "<name>"},
   ```

   For example:

```
"metadata": {"name": "workspace"},
```

> **TIP**
>
> You can add any other metadata you'd like to help you track where the cell is being used. For example:
>
> ```
>     "metadata": {
>       "name": "workspace",
>       "msdoc": "how-to-track-experiments.md"
>     },
> ```

Set up repository access

If the notebook file you want to reference is in a different repository, set up the code repository as a dependent repository.

Jupyter notebook snippet syntax reference

Once your notebook contains the required metadata, reference it in your markdown file. Use the `<cell-name-value>` you added to the notebook, and the `<path>` you set up as your dependent repository.

```
[!notebook-<language>[] (<path>/<notebook-name.ipynb>?name=<cell-name-value>)]
```

For example:

```
[!notebook-python[] (~/MachineLearningNotebooks/train-on-local.ipynb?name=workspace)]
```

> **IMPORTANT**
>
> This syntax is a block Markdown extension. It must be used on its own line.

Use any of the supported languages for the `<language>` identifier.

Interactive code snippets

Inline interactive code blocks

For the following languages, code snippets can be made executable in the browser window:

- Azure Cloud Shell
- Azure PowerShell Cloud Shell
- C# REPL

When interactive mode is enabled, the rendered code boxes have a **Try It** or **Run** button. For example:

````
    ```azurepowershell-interactive
    New-AzResourceGroup -Name myResourceGroup -Location westeurope
    ```
````

renders as follows:

```
New-AzResourceGroup -Name myResourceGroup -Location westeurope
```

And

`````
```csharp-interactive
var aFriend = "Maria";
Console.WriteLine($"Hello {aFriend}");
```
`````

renders as:

```
var aFriend = "Maria";
Console.WriteLine($"Hello {aFriend}");
```

To turn on this feature for a particular code block, use a special language identifier. The available options are:

- `azurepowershell-interactive` - Enables the Azure PowerShell Cloud Shell, as in the preceding example
- `azurecli-interactive` - Enables the Azure Cloud Shell
- `csharp-interactive` - Enables Try .NET

For the Azure Cloud Shell and PowerShell Cloud Shell, users can run commands against only their own Azure account.

Code snippets included by reference

You can enable interactive mode for code snippets included by reference. To turn on this feature for a particular code block, use the `interactive` attribute. The available attribute values are:

- `cloudshell-powershell` - Enables the Azure PowerShell Cloud Shell, as in the preceding example
- `cloudshell-bash` - Enables the Azure Cloud Shell
- `try-dotnet` - Enables Try .NET
- `try-dotnet-class` - Enables Try .NET with class scaffolding
- `try-dotnet-method` - Enables Try .NET with method scaffolding

Here are some examples:

```
:::code source="PowerShell.ps1" interactive="cloudshell-powershell":::
```

```
:::code source="Bash.sh" interactive="cloudshell-bash":::
```

For the Azure Cloud Shell and PowerShell Cloud Shell, users can only run commands against their own Azure account.

Snippet syntax reference

Syntax:

```
:::code language="<language>" source="<path>" <attribute>="<attribute-value>":::
```

> **IMPORTANT**
>
> This syntax is a block Markdown extension. It must be used on its own line.

- `<language>` (*optional*)

- Language of the code snippet. For more information, see the Supported languages section later in this article.

- `<path>` (*mandatory*)
  - Relative path in the file system that indicates the code snippet file to reference.

- `<attribute>` and `<attribute-value>` (*optional*)
  - Used together to specify how the code should be retrieved from the file and how it should be displayed:
    - `range`: `1,3-5` A range of lines. This example includes lines 1, 3, 4, and 5.
    - `id`: `snippet_Create` The ID of the snippet that needs to be inserted from the code file. This value cannot coexist with range.
    - `highlight`: `2-4,6` Range and/or numbers of lines that need to be highlighted in the generated code snippet. The numbering is relative to the lines displayed (as specified by range or id), not the file.
    - `interactive`: `cloudshell-powershell`, `cloudshell-bash`, `try-dotnet`, `try-dotnet-class`, `try-dotnet-method` String value determines what kinds of interactivity are enabled.
    - For details about tag name representation in code snippet source files by language, see the DocFX guidelines.

To enable Try .NET with class scaffolding:

```
:::code language="csharp" source="ClassLevel.cs" interactive="try-dotnet-class":::
```

This will render a code block with a **Try It** button. When the **Try It** button is clicked, the interactive window only displays code inside the class.

```
public static void Main()
{
    // Specify the data source.
    int[] scores = new int[] { 97, 92, 81, 60 };        // Define the query expression.

    IEnumerable<int> scoreQuery =
        from score in scores
        where score > 80
        select score;

    // Execute the query.
    foreach (int i in scoreQuery)
    {
        Console.Write(i + " ");
    }
}
```

To enable Try .NET with method scaffolding:

```
:::code language="csharp" source="MethodLevel.cs" interactive="try-dotnet-method":::
```

This will render a code block with a **Try It** button, that when selected, activates an interactive window that only displays code inside the method.

```
// Create a new dictionary of strings, with string keys.
//
Dictionary<string, string> openWith =
    new Dictionary<string, string>();

// Add some elements to the dictionary. There are no
// duplicate keys, but some of the values are duplicates.
openWith.Add("txt", "notepad.exe");
openWith.Add("bmp", "paint.exe");
openWith.Add("dib", "paint.exe");
openWith.Add("rtf", "wordpad.exe");

// The Add method throws an exception if the new key is
// already in the dictionary.
try
{
    openWith.Add("txt", "winword.exe");
}
catch (ArgumentException)
{
    Console.WriteLine("An element with Key = \"txt\" already exists.");
}
```

To enable Try .NET with full code experience:

```
:::code language="csharp" source="Example.cs" interactive="try-dotnet":::
```

This will render a code block with a **Try It** button, that when selected, activates an interactive window that displays full code including the using statement.

```
using System;

namespace Calculator
{
    class Program
    {
        static void Main(string[] args)
        {
            // Declare variables and then initialize to zero.
            int num1 = 1; int num2 = 2;

            Console.WriteLine("num1 + num2 = " + (num1 + num2));
            Console.WriteLine("num1 - num2 = " + (num1 - num2));
            Console.WriteLine("num1 * num2 = " + (num1 * num2));
            Console.WriteLine("num1 / num2 = " + (num1 / num2));
        }
    }
}
```

Supported languages

The Docs Authoring Pack includes a feature to provide statement completion and validation of the available language identifiers for code fence blocks. Available languages are pulled from the Docs **devlang** taxonomy, available here. Each devlang slug corresponds to a standard label and accurate colorization for the language. Only languages in the list will be rendered correctly on Docs.

To request a new language, follow the process in How to request metadata changes.

TIP

The Docs Authoring Pack, Dev Lang Completion feature uses the first valid alias when multiple aliases are available.

Next steps

For information on text formatting for content types other than code, see Text formatting guidelines.

Add links to articles

5/27/2021 • 18 minutes to read

This article describes how to use hyperlinks from pages hosted at Docs. Links are easy to add into Markdown with a few varying conventions. Links point users to content in the same page, other neighboring pages, or to external sites and URLs.

By order of preference, links hosted on Docs should be Relative if they are in the same repository and docset. If they are in a different docset, even if in the same repository, they should be Site Relative. Links to content hosted on Docs shouldn't use a complete URL, otherwise known as Fully Qualified Domain Names (FQDN). Using a complete URL from Docs to other content on Docs will cause that link to be non-functional in air-gap 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 a lot of 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 is not 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 be 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 is 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 is 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. This is in contract to a file relative path, which begins with the current file as the point of origin, and not the root directory.
- A **fully qualified domain name (FQDN)** includes the top-level domain, sub-domain, 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 a FQDN, a URL also include the transmission protocol, such as *http://* (the hypertext protocol, for example).

Link priorities and examples

LINK TYPE	PRIORITY	EXAMPLE	LAY EXAMPLE
File relative path	1st	`another-file.md`	
Root relative path	2nd	`~/different-directory/another-file.md`	
Site relative URL	3rd	`/windows/uwp/get-started/get-set-up`	
FQDN / Full URL	Avoid	`https://doc.microsoft.com/help/contribute/links-how-to`	
Cross repo reference	1st (if supported)	See Cross repo reference	

Link validation

Build validation in the PRs (just-in-time) give Suggestion, Warning, Error, or Validate messages that can allow or prevent the merging of PRs into many of the Docs repos.

The table below outlines the levels of validation support depending on the link type.

LINK TYPE	PR	BOOKMARK	AIR GAPPED CLOUD (AGC)
FQDN / Full URL	N	N	N
Site relative URL	N	N	Y
Root relative path	Y	Partial	Y
File relative path	Y	Partial	Y
Reference	Partial	Partial	Y
HTML URL	N	N	Partial
Cross reference	Y	--	Y

Links to articles in the same docset

To link to an article in the same docset, use a *relative path* link. A relative path is the path to the target file relative to the current file. Use the following syntax to build a relative path:

- **file.md** or **./file.md** specifies a file that's in the same directory (folder) as the current file
- **ide/file.md** or **./ide/file.md** specifies a file that's in a child directory (subfolder) named **ide**
- **../file.md** specifies a file that's in the parent directory of the current directory
- **../../file.md** specifies a file that's two directories above the current directory
- **../ide/file.md** specifies a file that's in a directory named **ide** that's a peer directory of the current directory

For example:

```
[link text](../../folder/filename.md)
```

Instead of constructing a relative path from the current file to the target file, you can start the file path at the root of the docset. This type of link starts with **~**. For example, if the docset's root folder is **docs**, the current file is **docs/ide/current-file.md**, and the target file is **docs/test/unit-test/load-tests.md**, the link is as follows:

```
[load tests](~/test/unit-test/load-tests.md)
```

> **TIP**
>
> Starting links with ~ produces invalid links when navigating source repositories on GitHub.

Include the **.md** file extension in all types of relative link. If you don't, the link may still work on the docs.microsoft.com site, but it won't work when you view the file on GitHub. Also, a build warning is generated at build time.

Advantages

Relative path links, that is, links that don't start with **/** and that do end with **.md**, are validated at build time. Build 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 air-gapped environments like JEDI, and in offline books (if that docset is published as an offline book). For example, to link to the article at **https://docs.microsoft.com/windows/uwp/get-started/get-set-up**, the link syntax is **/windows/uwp/get-started/get-set-up** as shown here:

```
For more information, see [Get set up](/windows/uwp/get-started/get-set-up).
```

Don't append the file extension *.md* on a link to an article in another repo or docset.

Site-relative links aren't validated at build time like file-relative links. If you have a typo in your link, no warning is generated. Site-relative links are also not click-navigable on the staging server `review.docs.microsoft.com`. To verify that the link works, replace the branch name in the link's generated URL with `master`. You can also check the broken links report for your docset on ops.microsoft.com after you publish your changes.

Links to external sites

Use a complete URL to link to an external site, including non-docs.microsoft.com Microsoft sites.

```
[Microsoft](https://www.microsoft.com)
```

All links must be secure (`https` instead of `http`) whenever the target supports it.

Locale codes in external links

Don't include locale codes such as `en-us` in your links to MSDN, TechNet, microsoft.com, and Azure.com articles. Hard-coded locale codes prevent localized content from displaying, which is a bad customer experience for users in other locales. When you copy a URL from a browser, delete the locale code from it when you create your link.

For example:

```
[OneDrive](https://developer.microsoft.com/onedrive)
```

And not:

```
[OneDrive](https://developer.microsoft.com/en-us/onedrive)
```

In some cases, the locale code is necessary for the link to work. For example, links to videos on Microsoft Virtual Academy require a locale code. Always test your link without the locale code to be sure it functions.

Third-party site link guidance

Minimize links that send users to a non-Microsoft site. If you must link to a third-party site, use the following guidance:

- **Accountability**: Link to third-party content when it's the third party's information to share. For example, it's not Microsoft's place to tell people how to use Android developer tools—that is 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

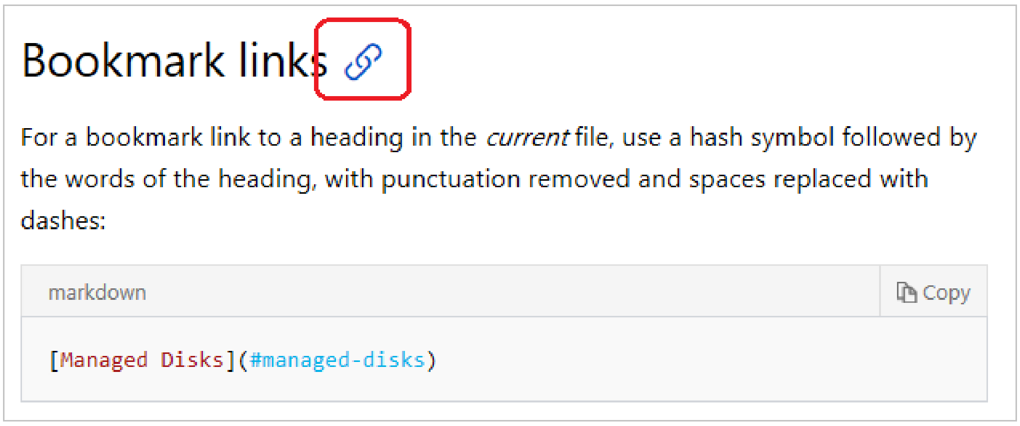
For a bookmark link to a heading in the *current* file, use a hash symbol followed by the lowercase words of the heading. Remove punctuation from the heading and replace spaces with dashes:

```
[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 dashes:

```
[Managed Disks](../../linux/overview.md#managed-disks)
```

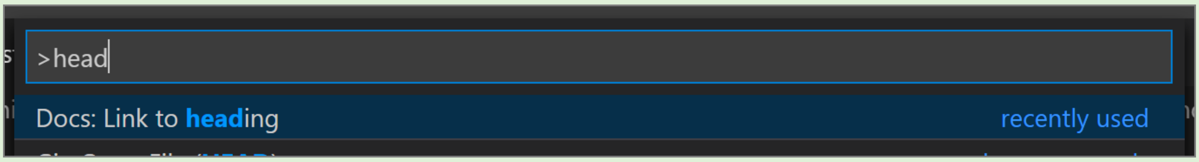
You can also copy the bookmark link from the URL. To find the URL, hover your mouse over the heading line on docs.microsoft.com. You should see a link icon appear:



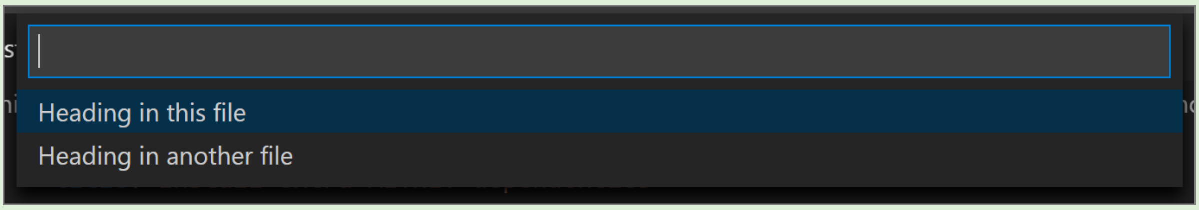
Click the link icon and then copy the bookmark anchor text from the URL (that is, the part after the hash).

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 auto-generated bookmarks as described in bookmark links. For hub and landing pages, declare anchors as follows:

```
## <a id="anchortext">Header text</a>
```

or

```
## <a name="anchortext">Header text</a>
```

And the following to link to the anchor:

```
To go to a section on the same page:
[text](#anchortext)

To go to a section on another page.
[text](filename.md#anchortext)
```

> **NOTE**
>
> Anchor text must always be lowercase and not contain spaces.

Links to specific versions

To hard code a link so it always goes to a specific version of an article, add the `view` parameter with its value set to the moniker (product and version) you want to link to, and then add `&preserve-view=true` so Docs knows you've set the version explicitly and it shouldn't be overridden. For example, the following URL will always resolve to the SQL Server 2019 (v15) version of the article:

```
https://docs.microsoft.com/sql/t-sql/functions/cast-and-convert-transact-sql?view=sql-server-ver15&preserve-view=true
```

If you set the view parameter without adding `&preserve-view=true`, Docs won't know that the view was set explicitly, and the version may be overridden based on the default version and the user's selections.

XRef (cross reference) links

XRef links are the recommended way to link to APIs, because they're validated at build time. Before using XRef links in your article, make sure that XRef-style links are enabled for your docset.

To link to auto-generated API reference pages in the current docset or other docsets, use XRef links with the unique ID (UID) of the type or member. UIDs are case sensitive.

Check if the API you want to link to is on docs.microsoft.com by typing all or some of its full name in the .NET API browser or Windows UWP search box. If you don't see any results displayed, the type isn't yet on docs.microsoft.com.

You can use one of the following syntaxes:

- Auto-links

  ```
  <xref:UID>
  <xref:UID?displayProperty=[chosen display property]>
  ```

  Example: `<xref:System.String>`

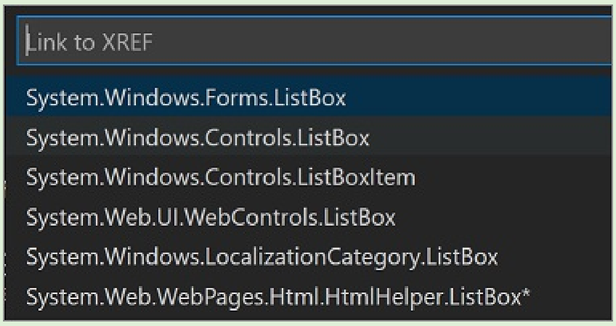
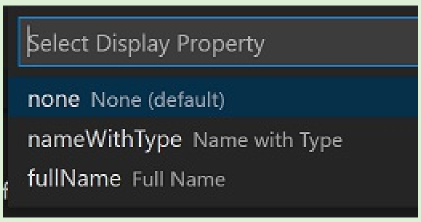
For more information about the `displayProperty` parameter, see Display properties.

- Markdown-style links

  Use Markdown-style xref links when you want to customize the link text that's displayed beyond the qualified or unqualified API name.

  ```
  [link text](xref:UID)
  ```

  Example: `[String class](xref:System.String)`

> **TIP**
>
> You can use the Docs Markdown extension for VS Code (part of the Docs Authoring Pack) to insert .NET XRref links (that is, those surfaced in the .NET API Browser) into Markdown files.
>
> 1. Press **F1** in Visual Studio Code to open the command palette, then type in **xref**. Select the **Docs: Link to Xref** option.
>
> 
>
>
> 2. Type in part or all of the API's name, for example, **listbox**, and then press **Enter**. A list of matching APIs is shown.
>
> 
>
>
> 3. Select the desired API or type in some text to further filter the list.
>
> 4. Select a display property.
>
> 
>

Display properties

By default, link text for auto-links shows only the unqualified API name. Add the optional `displayProperty` query parameter to partially or fully qualify the displayed API name.

DISPLAY PROPERTY	DESCRIPTION	EXAMPLE
`fullName`	Displays the fully qualified API name.	System.String.Trim()
`nameWithType`	Displays **namespace.type** for types and **type.member** for type members.	String.Trim()

XRef build warnings and incremental builds

An incremental build only builds files that have changed or been affected by a change. If you see a build warning about an invalid XREF link, but the link is actually 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**:

```
<meta name="page_type" content="dotnet" />
<meta name="page_kind" content="class" />
<meta name="description" content="Represents text as a sequence of UTF-16 code units. " />
<meta name="toc_rel" content="_splitted/System/toc.json" />
<meta name="source_url" content="" />
<meta name="ms.assetid" content="System.String" />
<meta name="pdf_url_template" content="https://docs.microsoft.com/pdfstore/en-us/VS.dotnet-api-docs/{branchName}{pdfName}" />
<meta name="search.mshattr.devlang" content="csharp vb cpp" />
```

- Use the autocomplete site by appending some or all of the name of the type to the URL. For example, entering `https://xref.docs.microsoft.com/autocomplete?text=Writeline` in the address bar of your browser displays all the types and methods that contain **Writeline** in their name, along with their UID.

Verify the UID

To test if you have the correct UID, replace **System.String** in the following URL with your UID, and then paste it into the address bar of a browser:

https://xref.docs.microsoft.com/query?uid=System.String

> **TIP**
> The UID in the URL is case-sensitive, and if you're checking a method overload UID, don't include spaces between the parameter types.

If you see something like the following, you have the correct UID:


```
[{"uid":"System.String","name":"String","fullName":"System.String","href":"https://docs.microsoft.com/dotnet/api/system.string","tags":",/dotnet,netframework-4.5.1,netframework-4.5.2,netframework-4.5,...xamarinmac-3.0,public,","vendor":null,"hash":null,"commentId":"T:System.String","nameWithType":"System.String"},
{"uid":"System.String","name":"String","fullName":"System.String","href":"https://docs.microsoft.com/dotnet/api/system.string","tags":",/dotnet,netframework-4.5.1,netframework-4.5.2,netframework-4.5,netframework-4.6,netframework-4.6.1,...,xamarinmac-3.0,public,","vendor":null,"hash":null,"commentId":"T:System.String","nameWithType":"System.String"}]
```


If you just see [] displayed on the page, you have the wrong UID.

HTML encoding

Special characters in the UID need to be HTML encoded as follows:

CHARACTER	HTML ENCODING
`	%60
#	%23

CHARACTER	HTML ENCODING
*	%2A

See a full list of HTML character codes.

Encoding examples:

- **System.Threading.Tasks.Task`1** encodes as `System.Threading.Tasks.Task%601` (see the section on generic types)
- **System.Exception.#ctor** encodes as `System.Exception.%23ctor`
- **System.Object.Equals*** encodes as `System.Object.Equals%2A`

Generic types

Generic types are those types such as **System.Collections.Generic.List<T>**. If you browse to this type in the .NET API browser and look at its URL, you see that **<T>** is written as **-1** in the URL, which actually represents **`1**:

https://docs.microsoft.com/dotnet/api/system.collections.generic.list-1

To link to a generic type such as **List<T>**, encode the ` backtick character as **%60**, as shown in the following example:

```
<xref:System.Collections.Generic.List%601>
```

Methods

To link to a method, you can either link to the general method page by adding a * after the method name, or to a specific overload. For example, use the general page when you want to link to the Object.Equals method without specific parameter types. The asterisk character * is encoded as **%2A**. For example:

<xref:System.Object.Equals%2A?displayProperty=nameWithType> links to Object.Equals

To link to a specific overload, add parenthesis after the method name and include the full type name of each parameter. Do not put a space character between the type names or the link won't work. For example:

<xref:System.Object.Equals(System.Object,System.Object)?displayProperty=nameWithType> links to Object.Equals(Object, Object)

Reference-style links

You can use reference-style links to make your source content easier to read. Reference-style links replace inline link syntax with simplified syntax that allows you to move the long URLs to the end of the article. Here's Daring Fireball's example:

Inline text:

```
I get 10 times more traffic from [Google][1] than from [Yahoo][2] or [MSN][3].
I start my morning with a cup of coffee and [The New York Times][NY Times].
```

Link references at the end of the article:

```
<!--Reference links in article-->

[1]: http://google.com/
[2]: http://search.yahoo.com/
[3]: http://search.msn.com/
[ny times]: http://www.nytimes.com/
```

Make sure that you include the space between the colon and the link. If you forget to include the space, links to other technical articles are broken.

Links inside HTML markup

If your Markdown file includes HTML markup, you can't use Markdown syntax for links inside that HTML markup. Instead, use HTML anchor tags.

Correct:

```
<a href="https://www.microsoft.com">link text</a>
```

Incorrect:

```
[link text](https://www.microsoft.com)
```

Links to Azure PowerShell reference content

The Azure PowerShell reference content has been through several changes since November 2016. Use the following guidelines for linking to this content:

Structure of the URL

- For cmdlets:

  ```
  /powershell/module/<module-name>/<cmdlet-name>[?view=<moniker-name>]
  ```

- For conceptual topics:

  ```
  /powershell/azure/<topic-file-name>[?view=<moniker-name>]
  /powershell/azure/<service-name>/<topic-file-name>[?view=<moniker-name>]
  ```

The <moniker-name> portion is optional. If it's omitted, you'll be directed to the latest version of the content. The <service-name> portion is one of the examples shown in the following base URLs:

- Azure Active Directory (AzureAD) PowerShell content: https://docs.microsoft.com/powershell/azure/active-directory
- Azure Service Fabric PowerShell: https://docs.microsoft.com/powershell/azure/service-fabric
- Azure Information Protection PowerShell: https://docs.microsoft.com/powershell/azure/aip/overview
- Azure Elastic DB Jobs PowerShell: https://docs.microsoft.com/powershell/azure/elasticdbjobs

When you use these URLs, you're redirected to the latest version of the content. This way, you don't have to specify a version moniker, and you don't have to update the link when the version changes.

To create the correct link, find the page that you want to link to in your browser, and copy the URL. Then, remove "https://docs.microsoft.com" and the locale info.

When you're linking from a table of contents, you must use the full URL without the locale information.

Example links:

```
[Get-AzureRmResourceGroup](/powershell/module/azurerm.resources/get-azurermresourcegroup)
[Get-AzureRmResourceGroup](/powershell/module/azurerm.resources/get-azurermresourcegroup?view=azurermps-4.1.0)
[New-AzureVM](/powershell/module/azure/new-azurevm?view=azuresmps-4.0.0)
[New-AzureRmVM](/powershell/module/azurerm.compute/new-azurermvm)
[Install Azure PowerShell](/powershell/azure/install-azurerm-ps)
```

Backlinks from user interfaces and product portals

In order to link to Docs pages from web portals and other in-product user interfaces, there are special considerations such as using FWLinks and campaign codes for tracking. For more information, see Link to articles from the user interface.

FWLinks and aka.ms links

FWLinks and aka.ms links (known collectively as ShortLinks) both shorten and redirect URLs. You can use these links in web portals and other in-product user interfaces, in some situations. For more information, see Add the right type of link.

Caution

Don't use FWLinks and aka.ms links in docs.microsoft.com content. Automation periodically converts links like this to full URLs to improve SEO and be verifiable by platform tools such as the OPS Broken Links report. Use FWLinks only as a last resort to point to content outside of docs.microsoft.com (for example, when you link to an external page that doesn't yet have a URL, or when you link to an external page and you know its URL will change). If you must use an FWLink, include the "/p/" parameter to make it a permanent redirect, as in the following example: `[https://go.microsoft.com/fwlink/p/?LinkId=389595](https://go.microsoft.com/fwlink/p/?LinkId=389595)`. Using this parameter doesn't affect your ability to later update the FWLink.

Link to content from the user interface

5/17/2021 • 13 minutes to read

This article describes the process of creating context-sensitive help (CSH) links for a user interface (UI). For other types of contextual help, such as tooltips and help bubbles, see the Cloud Style Guide section on UX/UI content design.

If you need help with your content integration strategy, contact Sarah Luck or Miriam Lottner.

Overview

Some users may need help with a concept or procedure while using a product or service. To address this need for just-in-time assistance, add a CSH link to the UI. The link targets core content that typically opens in a new browser tab. This type of integrated help is designed to minimize on-page explanation while providing a resource that helps users move forward with a task.

The type of link you choose depends on where you're using the link:

- For most products and services, you'll use either an FWLink or a hard-coded full link to `https://docs.microsoft.com/<path/to/your/content>`. For more information about which option to choose, see Add the right type of link later in this article.
- For Microsoft 365, use the MARVELCSH system for all context-sensitive help links. For more information, see the published onboarding guidance.

To implement a CSH linking strategy for your product, service, or feature, follow these steps:

1. Understand the context in which the link appears.
2. Choose the best content as your help source.
3. Determine the best link text to use, considering accessibility requirements.
4. Add the right type of link and reference it in code.
5. Use campaign IDs for tracking.
6. Monitor traffic and satisfaction to ensure the links are working well.

Understand context

Links that target content are intended to provide help for the current user context. We can't yet customize the content experience for the user context based on factors such as tenant lifecycle, subscription type, or last action. Today, context is limited to the location in the UI and our intuition of the user's mindset and intent.

When in the UI, users are in task-completion mode. Our goal is to keep them on task and in the UI, so don't interrupt that flow with unnecessary clicks out of the workspace. Think of help resources in the UI as a safety net – it's great it's there to catch the users who stumble, but even better if they don't need help at all. If a user clicks on a help link, it's likely they're confused, unsure, or blocked by not understanding next steps or how features work together.

Linked help content should immediately deliver the information the user needs to complete the task. Use the smallest, most unobtrusive type of help for the current context. Most of the time, you don't need to link to content. You can use placeholder text or a help bubble to explain correct formatting or a tooltip to introduce an unfamiliar term. Well-written strings coupled with intuitive workflows go a long way toward reducing the need for additional help.

Example links in context

The following screenshots illustrate some examples of link placement in the UI.

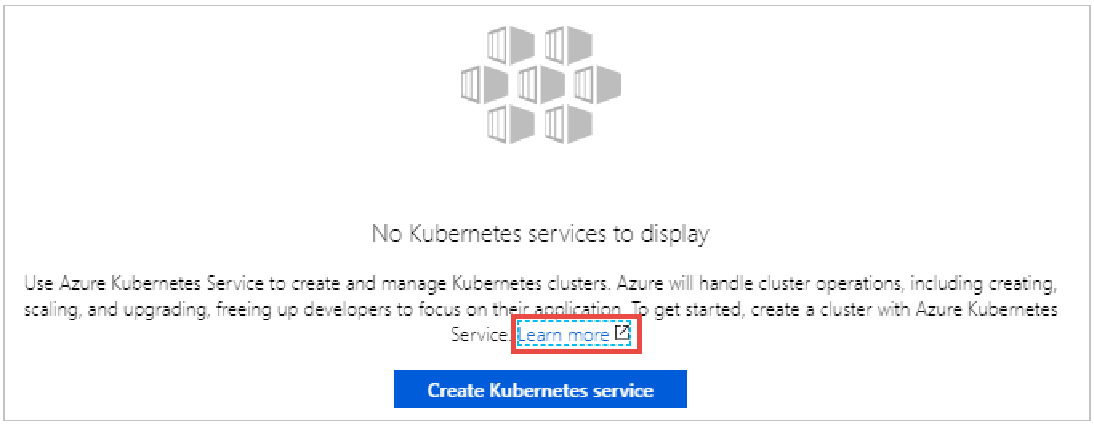


Figure 1: Learn more link in Azure service empty state

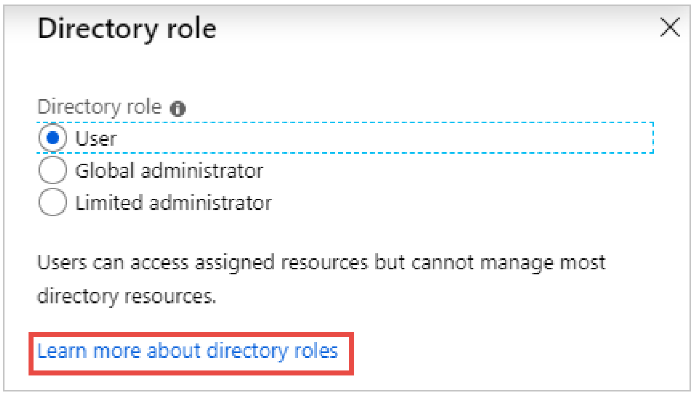


Figure 2: Learn more link in Azure form footer

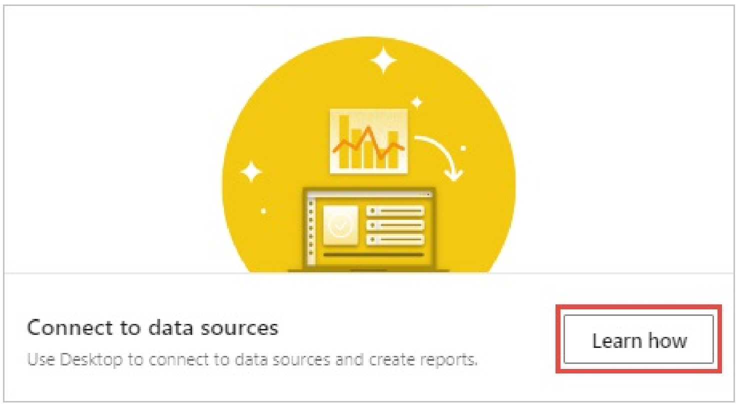


Figure 3: Learn how link in Power BI getting started

Choose the best content

Links embedded in the UI should be specific, targeted, and task-oriented. When possible, choose content on docs.microsoft.com, the authoritative site for product-related technical content.

When you choose a link target, consider the location in the UI and the user intent. Why did you want the user to click there and what information do they need right now? If you're linking to an article in documentation, it's often best to link to a specific section to go right to what the user needs to know. Linking to a specific section minimizes scrolling and unnecessary overview and instruction. Follow these general guidelines:

DO	DON'T
Work with writers, designers, and engineering teams to design the contextual help strategy and select link targets.	Duplicate content or create one-off processes.
Link to articles, article sections, Learn modules, and other content on docs.microsoft.com	Link to destinations that aren't maintained to follow the product lifecycle.
Link to azure.com if connecting to Azure pricing information.	Link to azure.com for product overviews or technical information.

Determine the best link text

Links in the UI must follow accessibility guidelines and be optimized for screen readers. There are three elements to link text:

- The visible label on the page
- A tooltip that appears on hover
- ARIA text, from a suite of web standards ("Web Accessibility Initiative - Accessible Rich Internet Applications") to make the Web accessible to people with disabilities

Screen readers always narrate the tooltip and append it to the other labels. If no ARIA text is present, the visible label is narrated. When ARIA text is present, screen readers read that text instead of the visible label.

Visible label

Use a short phrase as link text that provides some indication of the link destination. For a sighted user, the phrase "learn more" when appended to a string is fine. The string provides the necessary context. However, a vision-impaired user relies on a screen reader to identify links on the page. If there's more than one link on the page that's labeled "learn more", it can be confusing. It's a best practice to include a phrase that says what the user will learn more about. Follow these guidelines for visible link text:

- Keep it short, but give a clue as to where the link will take you.
- Use "Learn more" as stand-alone text when targeting generic content, like an overview.
- Use "Learn more about *something*" when targeting intent-focused content, like a tutorial or a Learn module.
- Only use the "learn more" phrase when the link targets technical content.
- If there's more than one "learn more" link on the page, use link-specific text.

For more information, see "Link text" in the Microsoft Cloud Style Guide.

Tooltip

Tooltips DO NOT repeat the visible label. They provide helpful information for all users. Include a tooltip with link text to let readers know how the link will behave. The tooltip is shown on hover to sighted users and is appended to either the visible label or the ARIA text for screen reader narration. The tooltip can read "Content opens in a new window" or "Content opens in a new tab".

For more information, see "Tooltip guidelines" in the Microsoft Cloud Style Guide.

ARIA text

ARIA text doesn't display in the UI, but it's an important part of meeting accessibility standards. ARIA text is read by screen readers instead of the visible label. Think of ARIA text as "screen reader text." When ARIA text is present, a screen reader will narrate the ARIA text + the tooltip. To author ARIA text, follow these guidelines:

- Avoid language that assumes you can see, including directional terms and phrases such as "show" or "take a look".

- Avoid language that assumes you can use your hands.
- Use end punctuation to cause the screen reader to pause briefly and to avoid narrated text that runs together.
- Use text instead of symbols, which may be narrated differently by different screen readers.
- Don't just repeat the visible label. If your visible label says "learn more", the context may be understood by a sighted user because of what surrounds the text. Don't make that assumption for ARIA text – always include a full expression, such as "Learn more about securing resources with role-based access control."

Add the right type of link

We've always recommended FWLinks for in-product linking because they were designed for this purpose. Hard-coding full URLs (`https://docs.microsoft.com/<path/to/your/content>`) can result in issues that require a product or service update to correct. When you use an FWLink, you can change the link target easily and quickly, with new destinations live within minutes. Making changes to hard-coded URLs is definitely quicker in services than on-premises products, but it can still take time and typically requires a developer. For more information, see FWLinks and aka.ms links.

The introduction of air-gapped clouds (AGC) like the one used for Joint Enterprise Defense Infrastructure (JEDI) presents a problem because there is currently no redirection service available in the AGC to handle FWLinks or aka.ms links. There is a version of Docs in the AGC, but the FWLinks or aka.ms links don't resolve to the targets. In other words, these links return a 404.

There is now a plan to address this issue, but no ETA at this time (May 2021). In addition to the issue with FWLinks and aka.ms links, full URLs work only if the service handles them correctly. See the following table for details, and contact Sarah Luck or Michael Blythe if you or your engineering team needs more information.

PRODUCT OR SERVICE	CONSIDERATIONS / RECOMMENDATIONS
On-premises products	Use FWLinks
Services that aren't in the AGC	Use FWLinks
Services that are in the AGC	Given the plan for a redirection service in AGC, consider continuing to use FWLinks. If you want links to work today, the service needs to do two things: • Use full URLs • Rewrite the domain of all docs.microsoft.com links to the appropriate air-gapped domain. The Azure portal already does this, and we're checking with other services. Carefully evaluate whether the effort to update current FWLink and aka.ms links makes sense given the plan for AGC.
Microsoft 365	Use the MARVELCSH system

Use campaign IDs for tracking

A campaign ID is a query string that is appended to a URL to track sessions from a particular traffic source. Campaign ID query strings use this syntax: ?wt.mc_id=*your_unique_campaignID_here*, and they can be added to all CSH links - full URLs, FWLinks, and aka.ms links.

When applied to CSH links, a campaign ID is used to filter data for reporting, to track user actions for the session, and includes information about where the link is located. **We strongly recommend using campaign**

IDs to get detailed reporting about referral traffic from products and services. To assign a campaign ID to your link target, first generate the ID using the In-Product Tag Generator. For more information about campaign IDs, contact James M. Kim. To plan out large-scale use of campaign IDs, contact Sarah Luck or Miriam Lottner.

Monitor and maintain

After your CSH links are live, track performance to make sure you're achieving the results you want. Make sure content is maintained or updated based on where the link appears and what you think the customer needs.

Track performance

Use content reports to track performance. To isolate traffic, SAT, and verbatim to CSH views, filter the referrer to your product or service, like `portal.azure.com`. You can also look for a campaign ID in the destination URL. These reports can help you figure out whether you've selected the best content for this link. A high number of clicks on a CSH link as a percentage of users to the UI could indicate a problem with the user experience. You can also export traffic data for selected FWLinks and aka.ms links in the link manager tool.

For more information about contextual help metrics and the actions that data drives, see this presentation about CSH.

Maintain link health

Broken links, especially in user experiences, aren't just annoying, they look unprofessional. A resource that was supposed to help the user has become a source of frustration. As a link owner, monitor the occurrence of 404s and make corrections as quickly as possible.

To avoid broken links, update associated FWLinks before retiring content and when changing content titles or section labels. If you're not sure whether content is the target of a UI link, search the FWlink and aka.ms link system for the target URL to identify the associated links to update. If you're using a full URL to the content, update the URL in the product or service and add a redirect in docs.microsoft.com, as you would for any retired content.

FWLinks and aka.ms links

Both FWLinks and aka.ms links shorten and redirect URLs, but they were designed for different purposes. Aka.ms links provide an easy-to-remember URL for marketing and social-media campaigns. FWLinks were designed for product-to-web linking to avoid hard-coding the destination URL within a product's UI. FWlinks provide additional functionality for localization, but we typically don't use this functionality anymore for links to docs.microsoft.com. Either type of link will work from the UI, but we'll focus on FWLinks in this article.

Before you get started, read Add the right type of link to understand the implications of using FWlinks in some services. Also consider how you'll manage the set of FWLinks for your product or service. We've seen many instances of out-of-date and duplicate links; work with your partners to make sure your approach to linking and link ownership is maintainable. For more information about link maintenance, contact Kelly Pittman.

Create an FWLink

An FWLink has the following format: `https://go.microsoft.com/p/fwlink/?linkid=` `<number ID>` (where the system generates the numerical ID when you create the FWLink). Follow the instructions below to create an FWLink. **Always include the "/p/" parameter to make the redirect permanent**. Using this parameter doesn't affect your ability to later update the FWLink.

1. Go to the AKA Link Manager tool.
2. Set **Host** to `go.microsoft.com`, and then select **New Link**.
3. Complete the link details as shown in the tables below.
4. Select **Create**. If you get an error about a duplicate link, add details to the optional fields to make the link

unique.

5. The new link is displayed in the summary of links owned by you. Use the **copy to clipboard** icon to copy the link.

> **TIP**
>
> If you need to create or edit many links, download the Excel bulk tool add-in from the AKA Link Manager tool.

Link details

Asterisk denotes required field.

FWLINK FIELD	VALUE	NOTE
Target URL*	The URL that should open when the link is clicked.	Make sure that the target doesn't include **en-us** (or any other language locale). You can link to a bookmark to target the relevant content section.
CLCID (Country Locale ID)	Leave as default	
PRD (Product)*	Select the appropriate product, like Microsoft Azure (12177), Dynamics 365 (10965), or Office 365 (12413)	
PVER (Product Version)*	Leave as default	
PLCID (Product Locale ID)	Leave as default	
AR (Area)*	CSH	Describes the property of where the FWLink will be used (for example, *Settings app*).
SBA (Sub Area)*	General type of link (for example, Learn more)	Describes the topic or page where the FWLink will be used (for example, *Bluetooth settings* or *Turn on Bluetooth*)

Link owners

FWLINK FIELD	VALUE	NOTE
Security Group Alias (as a Link Owner)	Use a common security group if you have one, like azureportallinks, wdglinks, or ddfwlink. If you don't know of a relevant security group, check with other writers in your area or the product team.	The Link Manager tool lets you view only links owned by you or by security groups in which you have membership. If your team's group is the link owner, you limit who can view and maintain the link.
Co-Owner Aliases	User aliases separated by semicolon.	The feature PM and the documentation owner are recommended. Owners are automatically notified via email whenever an FWLink is updated.
Make this link Active	Enabled	

FWLINK FIELD	VALUE	NOTE
Allow Parameters	Optional	If you turn on this feature, you can forward query string parameters from your FWLink to your target URL.

Optional fields

Field usage is not validated. Use as many of the optional fields as you want to leave an historical trail for the link and to provide options for reporting.

FWLINK FIELD	VALUE	NOTE
Category	CSH-Static	
Description	Optional	Recommended to include some text like "link from portal VM create to procedural doc."
SBP (Sub Product)	Name of your product or service	
OS (Operating System)	Leave as default	
OLCID (OS Locale ID)	Blank	
OVER (OS Version)	Blank	
Mobile URL	Optional	Use only if the mobile URL is different from the target URL.
Optional Field 1	Documentation	
Optional Fields 2-5	Optional	You can use these fields to provide additional information about the link, such as the link text or whether it's part of an experiment.

Related resources

For more information about integrating content with user interfaces, see these resources:

- Microsoft Cloud Style Guide
- Microsoft 365 Context Sensitive Help

If you need help with your content integration strategy, contact Sarah Luck or Miriam Lottner.

Guidelines for using bullets and numbered lists in content

5/10/2021 • 2 minutes to read

For basic elements of grammar, like numbered and bulleted lists, we follow the Microsoft Writing Style Guide and Microsoft Cloud Style Guide.

In addition to that guidance, there are a limited number of exceptions or clarifications for numbered lists.

When should I use numbered lists?

You should use numbered lists for:

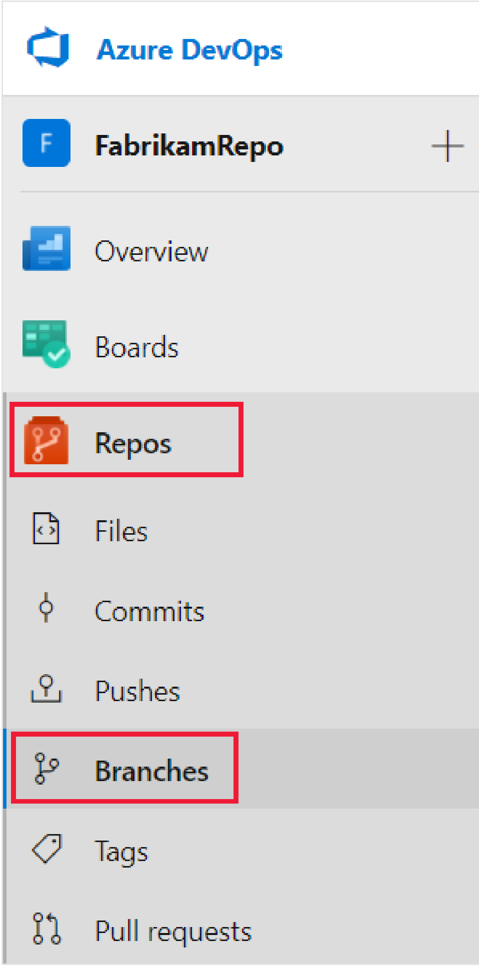
- Portal-based procedures
- Procedures that rely heavily on screenshots

The following is an example of using numbered lists for portal-based procedures or procedures that rely heavily on screenshots.

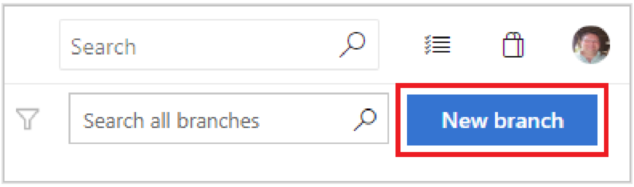
```
1. View your repo's branches by selecting **Repos** > **Branches** while viewing your repo on the web.

   :::image type="content" source="media/contribute-how-to-mvc-bullet-list/repos-branches.png" alt-text="Screenshot that shows how to view your branches.":::

1. Select the **New branch** button in the upper right corner of the page.

   :::image type="content" source="media/contribute-how-to-mvc-bullet-list/create-branch.png" alt-text="Screenshot that shows the New branch button.":::
```

This example renders as follows:

1. View your repo's branches by selecting **Repos** > **Branches** while viewing your repo on the web.



2. Select the **New branch** button in the upper right corner of the page.



When should I not use numbered lists?

Don't use numbered lists for:

- CLI-based procedures
- Any other procedure where each step is a block of code

The following is an example of document steps in a CLI-based procedure.

````
Use the `branch` command to create the branch and `checkout` to swap to that branch.

    ```console
    git branch feature1
    git checkout feature1
    ```
````

This example renders as:

Use the `branch` command to create the branch and `checkout` to swap to that branch.

Console

```
git branch feature1
git checkout feature1
```

General

- You should only number tutorials in the TOC when they are sequential.
- You shouldn't number article headings (H2 or H3 headings).

Tutorials

Tutorials include a special list to outline what you will learn in the article. This list requires specific format that is found in the templates.

Sample bulleted list for tutorials:

- OS disks and temporary disks
- Data disks
- Standard and Premium disks
- Disk performance

Include reusable content in articles

5/21/2021 • 4 minutes to read

You can embed other Markdown files within a Markdown file by using the Markdown INCLUDE syntax. This is useful for shared content, such as a note or warning you want to include in multiple files. You can include other file content either as a block or inline with other text.

Includes syntax

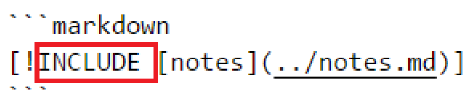
```
Block include is on its own line:

[!INCLUDE [<title>](<filepath>)]

Inline include is within a line: [!INCLUDE [<title>](<filepath>)]
```

- The `<filepath>` is the relative path to the file.
- The `<title>` is a label to provide context to the author. It's arbitrary and isn't visible to the reader, but could be useful for the content author to note context of the include file.

Here's an example of a block included file:

```
[!INCLUDE [notes](../notes.md)]
```

`INCLUDE` must be capitalized and there must be a space before the `<title>`:



Guidelines for using includes

Includes allow you to write, review, and localize content once, then use it in multiple places. This practice can save time and money and help ensure consistent wording across articles.

Do use include files:

- For boilerplate content such as notes and warnings that apply in many scenarios.
- For steps that apply in many procedures.
- For announcements that apply to an entire content set, such as deprecation notifications or other breaking changes.
- For content that requires legal review and must have precise wording.

Don't create include files:

- Before you're ready to use them. Uncalled includes add clutter to a repo and may be removed.
- To be called from outside documentation, for example from a product UI. A new feature, chromeless zones, is being developed for this scenario.
- For content that only needs to be in one file.
- For content that only needs to be in a few files, unless there's a specific need for exact consistency, as in the

case of some legal text.
- For content that is longer than one or two paragraphs.
- For content that is shorter than one sentence.
- For single words or phrases, such as product names. This practice results in many small Markdown files that are difficult to manage.
- As the only content of an article. Includes are meant to be supplemental to the content in the rest of the article.

Other requirements and considerations:

- Includes won't be rendered in the GitHub rendered view of your article because they rely on Docs extensions. They'll be rendered only after publication.
- Write all the text in an include file in complete sentences or phrases that don't depend on preceding or following text in the article that references the include. Ignoring this guidance creates an untranslatable string in the article.
- Don't embed include files within other include files.

Folder structure for included files

All included Markdown files should be stored in an `/includes` folder within the docset. This way they can easily be identified and exempted from build functionality that doesn't apply to included files. Be sure to add the `/includes` folder to the excluded content array in the docfx.json file for the docset:

```
"content": [
  {
   ...
    "exclude": [
      "**/obj/**",
      "**/includes/**",
          "README.md",
          ...
    ]
  }
]
```

Content guidelines for included files

To render properly on the published page, included files must meet the following requirements:

- Can't contain an H1 heading (`#`).
- If included inline, can't contain any block formatting, such as headings or alerts.
- Can't use an absolute path for `<filepath>`.

Using images in included files

Because `/includes` folders are excluded from build as described above, images stored in `/includes` folders and referenced in included files won't be displayed in published content. Store images in a `/media` folder outside the `/includes` folder.

As with regular articles, don't share media between include files. Use a separate file with a unique name for each include and article. Store the media file in the media folder that's associated with the include

Includes metadata

Included files should contain a subset of the standard article metadata. For more information, see Metadata for included files.

Cross-repo includes

Using cross-repo reference (CRR), you can configure dependent repositories. This configuration allows .md files from an outside repo to be built or used as includes as if they are in the same local repo.

1. Locate the **.openpublishing.publish.config.json** file in the root of the local repo to edit it.

2. Add the following JSON configuration to point to the outside repo to consume the include file(s) from. Choose a friendly name as the identifier for the repo. Set the URL of the outside repo, and the branch name to pull the files from.

   ```
   "dependent_repositories": [
     {
       "path_to_root": "friendly_name",
       "url": "https://github.com/MicrosoftDocs/outside_repo/",
       "branch": "master",
       "branch_mapping": {}
     }
   ]
   ```

   For more examples, see azure-docs-pr.

   For more information on the configuration format, see Cross-Repository Reference.

   > **WARNING**
   >
   > If the dependent repository is moved, the branch is renamed, or the include file is moved, the builds of the docset will show a build warning. It can be confusing to interpret the warning since the problematic missing file location is not local to the docset that was being built.

3. In the local repo, edit the .md file where you want to use the included outside file. Use the `~/friendly-name/` token to reference the dependent repo.

   ```
   [!INCLUDE[My included file](~/friendly_name/folder/include-file.md)]
   ```

Tabbed Conceptual

5/12/2021 • 5 minutes to read

Tabs enable conceptual content that is multi-faceted. They allow sections of a document to contain variant content renderings, each covering the same core concept or task, but allowing for customer reading preferences on tooling, platform, etc. This feature provides richer information to customers and should also improve writer productivity, reducing maintenance by eliminating duplicate content.

In this article you will:

- See the tab experience and learn how to use it in your content.
- Discover how to configure dependencies between different tab groups.
- Understand the limitations and future plans for tabs.
- Receive guidance for how to appropriately leverage tabs.
- Learn about approved tabs and the proposal process.

Basic Usage

Here's an example of the tab experience:

- Linux
- Windows

Content for Linux...

The above tab group was created with the following syntax:

```
# [Linux](#tab/linux)

Content for Linux...

# [Windows](#tab/windows)

Content for Windows...

---
```

Tabs are indicated by using a specific link syntax within a Markdown header. The syntax can be described as follows:

```
# [Tab Display Name](#tab/tab-id)
```

A tab starts with a Markdown header, `#`, and is followed by a Markdown link `[]()`. The text of the link will become the text of the tab header, displayed to the customer. In order for the header to be recognized as a tab, the link itself must start with `#tab/` and be followed by an ID representing the content of the tab. The ID is used to sync all same-ID tabs across the page. Using the above example, when a customer selects a tab with the link `#tab/windows`, all tabs with the link `#tab/windows` on the page will be selected. Try it out below and compare to the tab selection above.

- Linux
- Windows

Content for Linux...

The first instance of a link-header with the above syntax not only defines a tab, but defines the start of the group of tabs. Each successive link-header with this format, starts a new tab within the current group. The group itself is terminated with a Markdown horizontal rule `---`.

> **NOTE**
> When a customer selects a tab, the url's query string will be updated to capture the selection and when a customer navigates to a page with these query string values, the appropriate tabs will be auto-selected. This means that you can link to an article with a particular set of tabs pre-selected and customers can bookmark particular combinations as well.

Limitations of Tabs

1. You **can** use includes inside of a tab's content, but you **cannot** include a Markdown file within the content of a tab that itself contains tabs.
2. You **cannot** use H1s or H2s inside the content of a tab.
3. You **cannot** use tabs *inside of* nested includes.

Dependent Tabs

It's possible to make the selection in one set of tabs dependent on the selection in another set of tabs. Here's an example of that in action:

- Azure CLI
- PowerShell
- Portal

Azure CLI content for Linux...

Notice how changing the Linux/Windows selection above changes the content in the Azure CLI and PowerShell tabs. This is because the tab group defines two versions for each Azure CLI and PowerShell, where the Windows/Linux selection above determines which version is shown for Azure CLI/PowerShell. Here's the markup that shows how this is done:

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

Azure CLI content for Linux...

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

Azure CLI content for Windows...

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

PowerShell content for Linux...

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

PowerShell content for Windows...

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

Azure Portal content, independent of platform...

---
```

Dependent tabs are indicated by using an extension of the original tab link syntax. The syntax can be described as follows:

```
# [Tab Display Name](#tab/tab-id/dependency-tab-id)
```

To set up a dependent tab, you would create multiple versions of the header and add a second URL segment to the tab link, targeting the ID of the dependency. For example, both `#tab/azure-cli/linux` and `#tab/azure-cli/windows` represent the singular Azure CLI tab. Which one is displayed is based on whether linux or windows is selected above.

> **IMPORTANT**
>
> Remember that the segments in the tab link are IDs. In order for selection synchronization to work, the IDs bust be the same across tab groups and dependent tab groups.

Limitations of Dependencies

1. For dependent tab groups to function properly, the dependency must appear in the document prior to the dependent group.
2. You can only have one level of dependencies.

Guidance

Tabs are intended to represent multiple variations of the same content, based on different ways of approaching the task or problem being discussed. For example, if the article is "Create a Virtual Machine", then there are many ways to do this. Is it a Linux or Windows VM? Do you want to create it with Azure CLI, PowerShell, or directly through the Portal?

Here are a few things to use tabs for:

- **Do** use tabs for showing differences between platforms for the same task.
- **Do** use tabs for showing different tooling approaches for the same task.

> **NOTE**
>
> Make sure to order your tabs so that the tab you want to be the default selection is always first.

Here are a few things **not** to use tabs for:

- **Do not** use tabs to show ways to accomplish a task for different versions of the same product. Use the native versioning support of docs.
- **Do not** use more than four tabs. This will cause a terrible mobile experience. We hope to address this in the future.
- **Do not** use tabbed conceptual in combination with the `selector` or `multi-selector` Markdown extensions. While we are not deprecating the selector extensions, we think you'll find that the tab experience handles these scenarios better in most cases, and we encourage you to investigate migrating content over to this new approach.

Here is a **soft rule**:

- **Try not to** use tabs to show different programming language examples for the same task. First, consider using the native dev_langs selector for this.

Approved Tabs

Although we don't currently validate tab values, we recommend that you use existing values if possible. See the tab allowlist for existing values.

If none of these values are appropriate you can create new values and add them to the list via a PR. However, please do so sparingly. The strategy for content switchers is under review and is likely to change in the future.

Combining Rule

In some scenarios, content may be the same for multiple tabs. An example of this is iOS and macOS or Browser and Node.js. Rather than inventing new tabs for these scenarios, you can combine approved tabs together. The rule can be described as follows:

```
# [Display Name / Display Name / Display Name](#tab/tab-id+tab-id+tab-id)
```

- The display names should be concatenated using a `/`. There should be a single space on each side of the `/`.
- The tab IDs should be concatenated with a `+`. No spaces should be found anywhere.
- The order of concatenation should always follow the order that the items appear in the approved list.
- You cannot combine the `Other` tab.

Examples

- `[macOS / iOS](#tab/macos+ios)`
- `[Browser / Node.js](#tab/browser+nodejs)`

Converting SMS content to markdown

5/12/2021 • 2 minutes to read

Prerequisites:

Install pandoc on your machine http://pandoc.org/installing.html

Steps:

1. Log into SMS and switch to batch mode
2. Select each of the articles in your target dev center using the left navigation pane
3. Click “Export” and save the resulting CSV to a file.
4. Open PowerShell ISE and paste the following script into the script editor. This will output HTML files to a new directory called MSDN.

```
function getHTMLFromSmsExport($file) {
    cd $PSScriptRoot
    $export = (import-csv $file)

    new-item -ItemType Directory -Force -Path MSDN

    foreach ($row in $export) {
        $url = "https://msdn.microsoft.com/en-US/library/" + $row.ContentId
        echo $url

        $out = "MSDN\"+$row.ContentId+ "-" + $row.ContentName+".html"

        $httpContent = Invoke-WebRequest -URI $url -TimeoutSec 15
        $htmlContent = $httpContent.ParsedHtml.getElementById("content")
        if ($htmlContent -ne $null) {
            $html = $htmlContent.OuterHTML
            $html | Out-File -FilePath $out -Encoding utf8

            $images = $httpContent.ParsedHtml.getElementById("content").getElementsByTagName("img")

            foreach ($image in $images) {
                $chunks = $image.src.Split("/")
                $outName = "MSDN\" + $chunks[$chunks.Length -1]

                Invoke-WebRequest -URI $image.src -OutFile $outName -TimeoutSec 15
            }
        }
    }
}
getHTMLFromSmsExport("smsexport.csv") #Update this to point to your export CSV
```

5. Next, you can convert the html content (which is just the page content without any chrome) to Markdown, also using PowerShell

```
cd MSDN
$files = dir

foreach ($file in $files) {
    if ($file.Extension -eq ".html") {
        $out = $PSScriptRoot + "markdown\" + $file.BaseName + ".md"
        echo "pandoc $file -f html -t markdown -o $out"
        pandoc $file -f html -t markdown-raw_html-native_divs-native_spans+startnum-link_attributes-
header_attributes+hard_line_breaks --atx-headers -o $out
    }
}
```

6. Now that your content is converted into markdown, it can be added to your git repository.

Text formatting guidelines

4/20/2021 • 4 minutes to read

TIP

You may refer external docs contributors to this public contributor guide page for similar information. Keep the information in sync between these two pages as things change.

Consistent and appropriate use of bold, italic, and code style for text elements improves readability and helps avoid misunderstandings.

Bold

Use bold for UI elements, such as menu selections, dialog box names, and input field names.

Examples

- **This**: In **Solution Explorer**, right-click the project node, and then select **Add** > **New Item**.
- **Not this**: In Solution Explorer, right-click the project node, and then select Add > New Item.
- **Not this**: In *Solution Explorer*, right-click the project node, and then select *Add* > *New Item*.

Italics

Use italics for:

- Introducing new terms along with a definition or explanation.
- File names, folder names, paths.
- User input. See also Placeholders later in this article.

Examples

- **This**: In App Service, an app runs in an *App Service plan*. An App Service plan defines a set of compute resources for a web app to run on.
- **Not this**: In App Service, an app runs in an "App Service plan." An App Service plan defines a set of compute resources for a web app to run on.
- **This**: Replace the code in *HttpTriggerCSharp.cs* with the following code.
- **Not this**: Replace the code in `HttpTriggerCSharp.cs` with the following code.
- **This**: Enter *ContosoUniversity* for the **Name**, and then select **Add**.
- **Not this**: Enter "ContosoUniversity" for the **Name**, and then select **Add**.

Code style

Use code style for:

- Code elements, such as method names, property names, and language keywords.
- SQL commands
- NuGet package names
- Command-line commands
- Database table and column names
- Resource names that should not be localized (such as virtual machine names)

- URLs that you don't want to be clickable

Why? Older style guides specify bold for many of these text elements. However, most articles are localized, and code style tells the translator to leave that part of the text untranslated.

Code style can be *inline* (surrounded by `) or *fenced* code blocks (surrounded by ``` or ~~~) that span multiple lines. Put longer code snippets and paths in fenced code blocks.

Examples using inline styles

- **This**: By default, the Entity Framework interprets a property that's named `Id` or `ClassnameID` as the primary key.
- **Not this**: By default, the Entity Framework interprets a property that's named *Id* or *ClassnameID* as the primary key.
- **This**: The `Microsoft.EntityFrameworkCore` package provides runtime support for EF Core.
- **Not this**: The *Microsoft.EntityFrameworkCore* package provides runtime support for EF Core.

Examples of fenced code blocks

- **This**: No commands are sent to the database by statements that just change an `IQueryable`, such as the following code:

  ````
  ```csharp
  var students = context.Students.Where(s => s.LastName == "Davolio")
  ```
  ````

- **Not this**: No commands are sent to the database by statements that just change an **IQueryable**, such as **var students = context.Students.Where(s => s.LastName == "Davolio")**.

- **This**: For example, to run the `Get-ServiceLog.ps1` script in the `C:\Scripts` directory, type:

  ````
  ```powershell
  C:\Scripts\Get-ServiceLog.ps1
  ```
  ````

- **Not this**: For example, to run the **Get-ServiceLog.ps1** script in the **C:\Scripts** directory, type: "C:\Scripts\Get-ServiceLog.ps1."

- **All** fenced code blocks must have an approved language tag. For a list of support language tags, see How to include code in docs.

Placeholders

If you want the user to replace part of an input string with their own values, use placeholder text marked off by angle brackets. For example:

```
az group delete -n <ResourceGroupName>
```

You can note that the brackets must be removed when substituting real values. The use of italics is recommended for user input, and you can format italics within angle bracketed inline code:

> *<ResourceGroupName>* is the resource group where...

If you use italics without code format, escape the starting angle bracket by putting a back-slash in front of it (`\<`).

Curly braces { } are discouraged for use as syntactical placeholders. They may be confused with the same notation used in replaceable text, format strings, string interpolation, text templates, and similar programming constructs.

Placeholder names can be separated by hyphens ("kebab case"), with underscores, or not separated at all (Pascal case). Kebab case may generate syntax errors and underscores may conflict with underlining. All-caps may conflict with named constants in many languages, though it may also draw attention to the placeholder name.

> `<Resource-Group-Name>` or `<ResourceGroupName>`

Headings and link text

Don't apply an inline style such as italics, bold, or code to headings or hyperlink text.

Why?

People rely on standard hyperlink text to identify text elements as clickable links. Styling a link as code, for example, can obscure the fact that the text is a link. Headings have their own styles and mixing other styles in them looks bad.

Examples

- **This**: The *function.json* file is generated by the NuGet package Microsoft.NET.Sdk.Functions.
- **Not this**: The *function.json* file is generated by the NuGet package `Microsoft.NET.Sdk.Functions`.
- **This**:

```
### The Microsoft.NET.Sdk.Functions package
```

- **Not this**:

```
### The `Microsoft.NET.Sdk.Functions` package
```

Keys and keyboard shortcuts

When referring to keys or key combinations, follow these conventions:

- Capitalize the first letter of key names.
- Don't apply an inline style such as italics, bold, or code.
- Use "+" to join keys that are pressed at the same time.

Examples

- **This**: Select Alt+Ctrl+S.
- **Not this**: Press **ALT+CTRL+S**.
- **Not this**: Hit `ALT+CTRL+S`.

For more information, see Describing interactions with UI.

Exceptions

Style guidelines aren't rigid rules. In contexts where they harm readability, do something different. For example, an HTML table with mostly code elements might look too busy with code styling everywhere. You might choose bold styling in that context.

If you choose an alternate text style where code is normally called for, make sure it's okay for the text to be translated in localized versions of the article. Code is the only style that automatically prevents translation. For scenarios where you want to prevent localization without using code style, see Non-localized strings.

GitHub workflow for MicrosoftDocs

3/5/2021 • 7 minutes to read

MicrosoftDocs uses GitHub to host our Git repositories (repos). As a contributor you'll create and update files in GitHub hosted repos, and the publishing system reads the files from the GitHub repos.

This article describes some basic GitHub concepts and the GitHub workflow specific to docs.microsoft.com. The process becomes easier to follow when you understand the location, or where the content resides. The GitHub workflow is illustrated below:

Git

Git is a distributed version control system and is a command-line tool. A set of content and associated files is known as a repository. Git helps manage and track content changes made by various authors and allows for collaboration.

GitHub

GitHub is a cloud-based Git repository hosting platform. It provides a web interface to manage repositories and collaborate.

Repository

A repository, also known as a repo, houses a content set for an organization. For example, MicrosoftDocs, is the organization and docs-help-pr is one of the repositories. Repos are the main/default/published/current version of the content. In a repository, you can store and organize the necessary files for a content set. Within a repository, there are folders and files, images, and link to videos – everything the content project needs. The main/default branch is also referred to as Upstream, especially when using commands in the command-line interface.

Markdown source files are stored in Git repositories. *Git* is a version control system for tracking changes and coordinating work in files among multiple users. A Git *repository*, also known as a *repo*, is a collection of related files. For example, there is one repo for almost all the Azure documentation and one repo for all of the ASP.NET Core documentation.

A Git repo that publishes to docs.microsoft.com contains the following kinds of files:

- Article source files. In Markdown, with YAML metadata.
- Image files for diagrams and screenshots in articles. Usually PNG format.
- Code files for use in articles and as accompanying sample projects.
- Table-of-Contents source files. Mostly in YAML, some legacy Markdown.
- Config files. For example, one config file provides URL redirection rules for deleted articles.

The repo folder structure corresponds to URL structure. For example, the App Service Environments introduction is in the following repo folder and at the following URL:

```
articles/app-service/environment/intro.md
```

```
https://docs.microsoft.com/azure/app-service/environment/intro
```

Private and public GitHub repos

Most teams use a private repo for internal contributors. Private repos aren't visible to the public. If an unauthorized person goes to a private repo URL, they'll receive a `404 Not Found` error. Most private repos are identified with a -pr. For example, the Docs Platform Help Documentation repository is docs-help-pr.

Most private repos have a corresponding public repo that people outside of Microsoft can use to propose content changes. The public repo is regularly synced with the private one.

Some product teams work with a public repo only. For example, there are no private repos for ASP.NET. And some content is hosted in a private repo only, such as legal documentation that isn't open to external contributions or feedback.

See Should I work in a private or a public repository? for information on why and how we work with private repos.

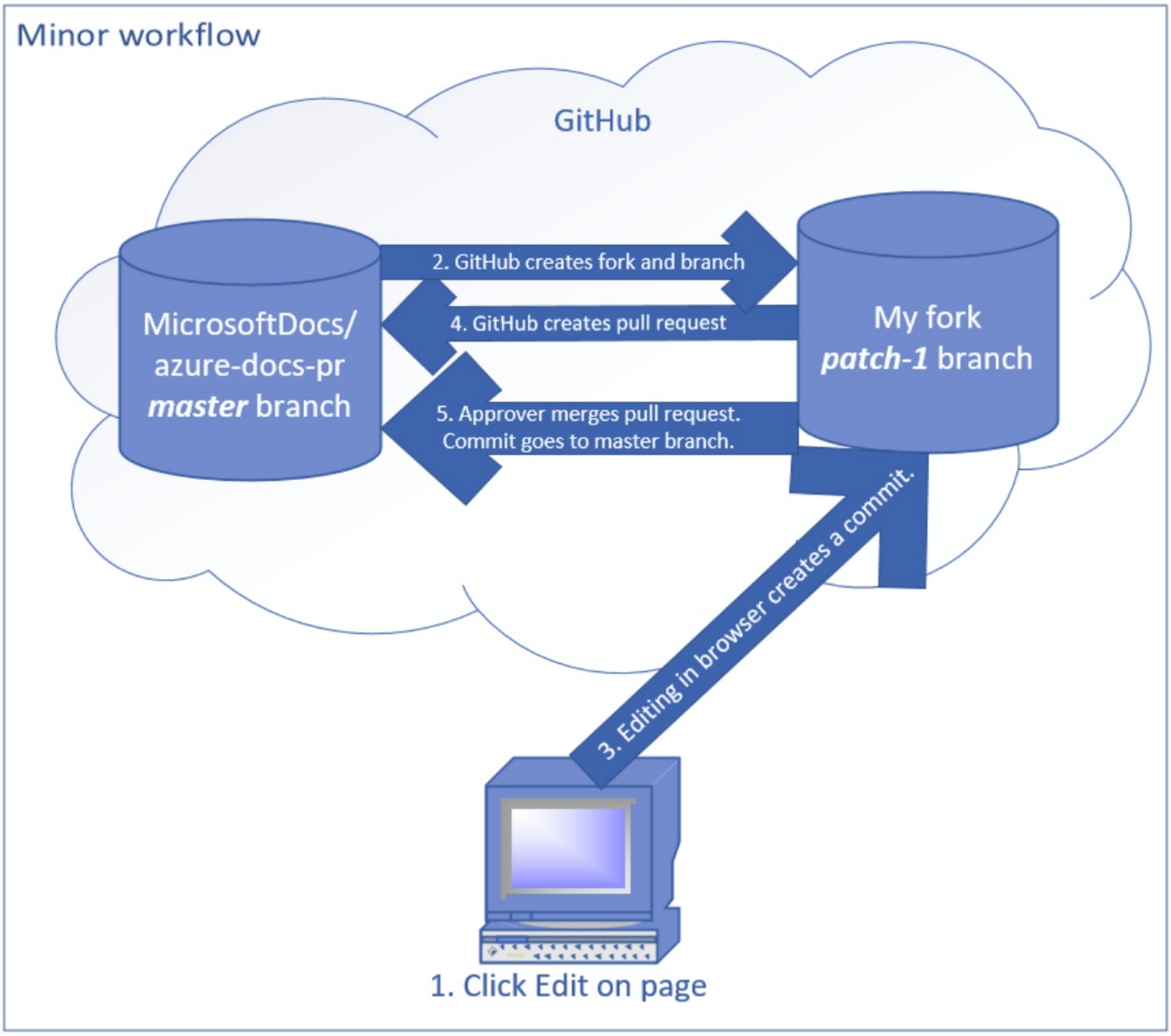
Fork

To edit an article, you must make changes in the GitHub repo it is published from. But in most cases, you don't have write permissions to that production repo. So, you make a copy of the repo in your own GitHub account and make your changes there. That copy of the production repo is called a *fork* in GitHub. A fork is a complete copy of the repository and is a snapshot in time. If changes have been made to a repo after you have forked, you need to pull the latest copy to your fork. You can fork a repository from within the GitHub UI.

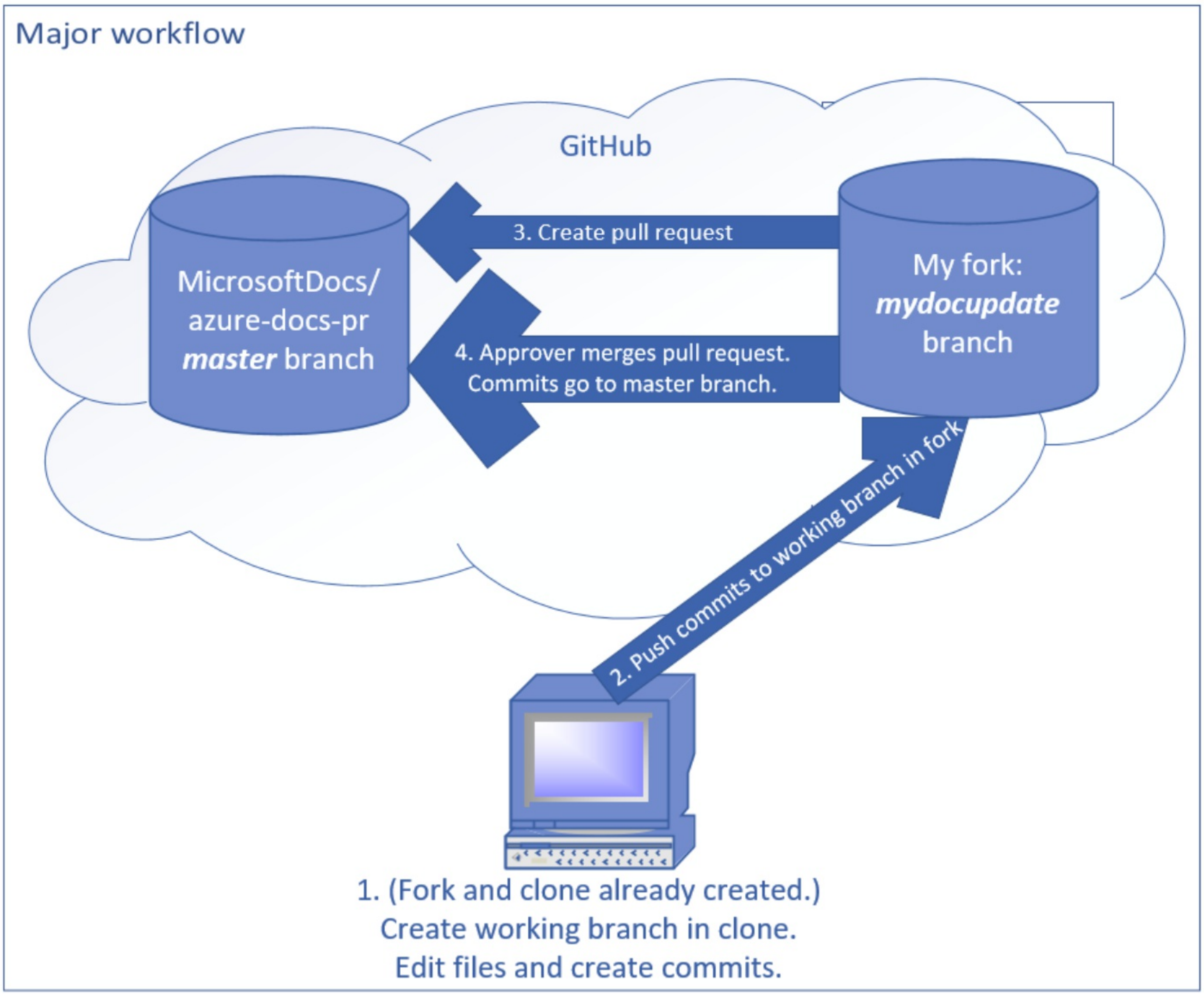
For example, the production repo (which we call the *upstream* repo) for this contributor guide it is MicrosoftDocs/docs-help-pr. When you fork this repo, you will have a full copy in your GitHub account: **YourUserName**/docs-help-pr.

Minor and major workflows

If all you want to do is fix a typo or make some similarly trivial change, you can edit a doc directly in the browser. For this *minor workflow*, you don't need a local clone repo, and a fork repo will be created for you automatically. For more about the minor workflow, see Edit an article on Docs. The following diagram illustrates the minor workflow.



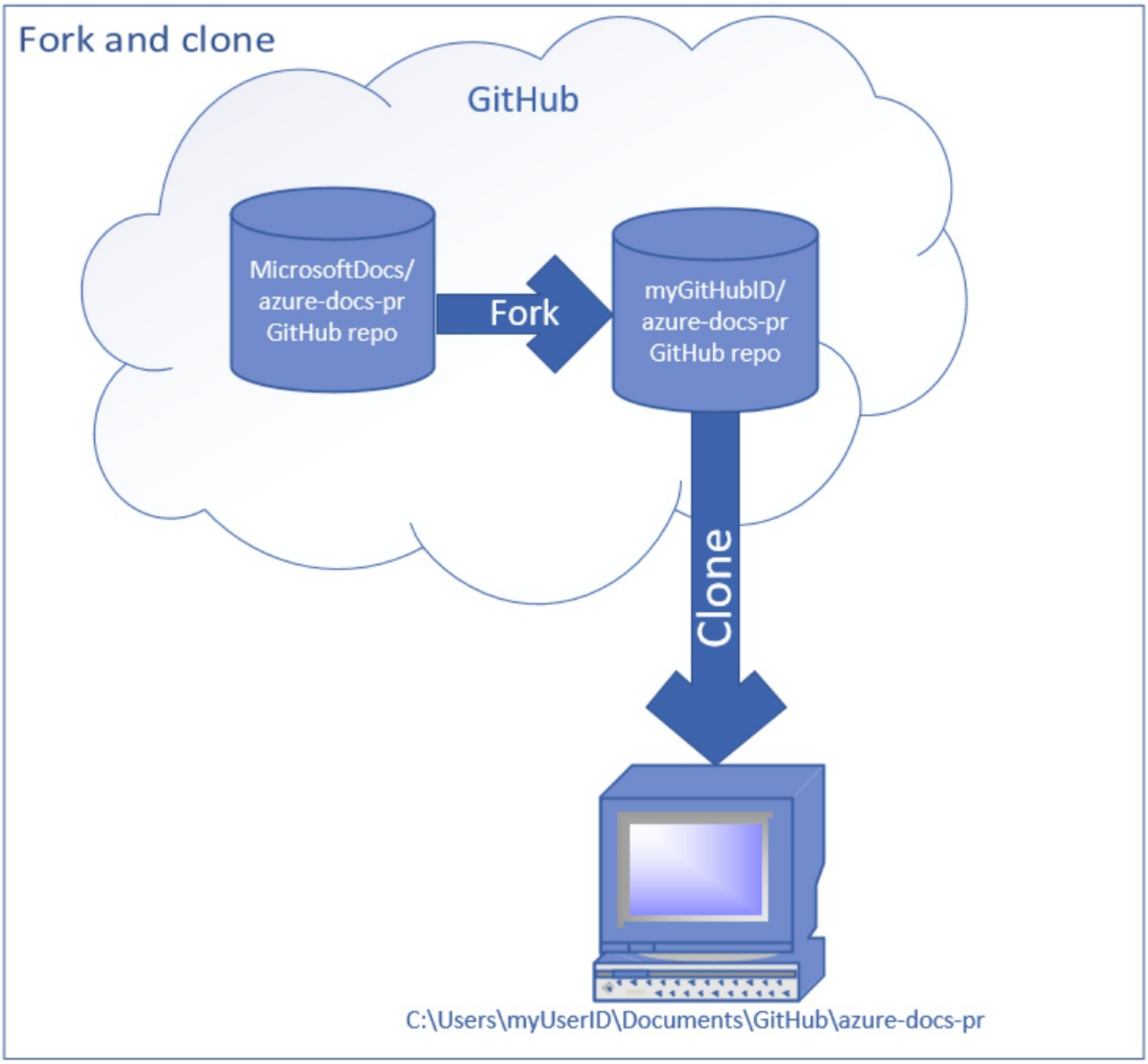
For more complex changes, it's more efficient to work with a specialized Markdown editor that has Git integration, such as Visual Studio Code. It will also be necessary at times to enter Git commands at the command prompt. For more about the *major workflow*, see Make changes to an article published on docs.microsoft.com. The following diagram illustrates the major workflow.



Clone

Although you have *write* permission to your repository fork, you can't use a full-featured text editor like Visual Studio Code directly in GitHub online. So, for major changes, you instead *clone* (download a copy of) the repo to your local computer and edit files there. Clones are copies of the fork in your local environment (for example your c: drive.). Since it's local, you can edit files using Visual Studio Code or some other Markdown editor. When ready, you *push* your changes to your fork, and then you create a pull request to merge those changes to the upstream repo.

The following diagram illustrates the relationships between a repository, fork, and clone.



To clone a fork, you'll need the clone URL from the GitHub UI and run the clone command in VS Code Terminal or Git bash. Step-by-step instructions are provided in the Prepare your environment article.

Remote

Remotes are aliases that point to repos in GitHub. You will have on named *origin* that points to your fork, and one named *upstream* that points to the production repo. Step-by-step instructions are provided in the Prepare your environment article.

Commit

When you create a commit, Git records the changes that you made locally in your working branch. Before you run the commit command, you must clearly tell Git, which file(s) to commit. So, you'll first use the Git add command with the file path followed by Git commit. You can use the VS Code Terminal or Git bash to commit the article(s) you changed. Step-by-step instructions are provided in Make changes to an article.

Push

After committing your changes to your local branch, you push the changes to the remote that points to your fork. Step-by-step instructions are provided in Make changes to an article.

Pull request

After you push the content changes to your fork, go to **GitHub.com** to create a pull request. A pull request basically asks the production repo to pull in the changes from the working branch in your fork. A pull request compares and displays differences between your fork and the production repo so you and others can easily review the changes.

A pull request also notifies the content author and reviewers that the content is ready to be reviewed. All reviews, comments, and feedback can be provided directly within GitHub.com. After the feedback and reviews are incorporated, the changes are *merged* to the main repository. Step-by-step instructions to create a pull request are provided in Creating a pull request.

Merge conflicts

With many people working on the same files, it's inevitable that at times different people will make different changes to the same part of a file. When that happens, a pull request can't be merged until the conflict is resolved. There are two ways to resolve the conflict:

- Create a new commit that resolves the conflict.
- Rebase your existing commits.

Searching the contributor guide

The internal-only contributor guide that you're reading right now isn't indexed by Google or Bing. You have two options for finding something specific in it: the TOC filter and site search.

Table of contents (TOC) filter

Directly above the TOC on the left side of the page is a text box with a "Filter by title" watermark. This is not a full-text search. Enter a word here and the TOC is limited to nodes that contain that word in the title, plus nodes where that word was deliberately entered into TOC metadata. Enter "merge", for example, and you see the *Resolve merge conflicts* article but not articles with other titles that happen to mention merging.

Site search

For a full-text search, start on any contributor guide page, and select the **Search** link with magnifying glass icon at the top right of the page. Enter a search term such as "merge" here, and you get a list of more than 50 pages that use that word.

If you can't find what you're looking for, you can expand the search scope. When you select **Search**, notice the **Contributor X** at the left of the text box. That means results will be limited to the contributor guide. To include other related doc sets, select the **X** to delete **Contributor X**. Search results from other doc sets such as OPS and localization will then be included. For "merge," the number of results now exceeds 350.

Migrating GitHub project documentation to Docs

5/12/2021 • 2 minutes to read

Per our secondary site policy, technical documentation for consumers of a Microsoft product or service needs to be on docs.microsoft.com (Docs). It shouldn't be on secondary sites. Developer docs in GitHub repos that host a product or service's tools or programming libraries belongs on Docs.

This guidance only applies to GitHub projects supporting products and services whose documentation is already onboarded to Docs. If you need to onboard an entirely new product or service to Docs, use this template to open a user story to get started with the onboarding process. Visit the Repo admin guide for more info.

The guidance in this article doesn't apply to Azure sample GitHub repos. Create and manage that content per our Create and publish sample code guidance.

What can live on GitHub? What must move to Docs?

Content written for contributors to the GitHub project should remain on GitHub. Content written for consumers of the project should live on Docs. Examples include install and usage documentation, library reference, and any getting started documentation for new users to the project.

Bring your GitHub content into an existing content on Docs

If the product or service that your GitHub project supports is already on Docs, contact the content developers and PMs servicing that content. Find out the best way to bring your content over from GitHub. They'll need to approve the new content on Docs and the changes to the table of contents that your material requires via a pull request. Once the PR is approved, your changes will be merged into the existing content repo.

If your GitHub project is a library or an API that has reference docs, review Reference Onboarding.

Is your GitHub project a standalone utility, sample application, or integration with an existing third party or open-source tool? If so, work with the content team for your product or service to figure out how to bring usage of the tool on to Docs.

How to rebase and squash

6/15/2021 • 7 minutes to read

Rebase means replace a commit with an equivalent one based on a different, usually newer, commit. Rebasing a working branch on the current master branch lets you deal with merge conflicts in advance. Doing so prevents the conflicts from showing up in a pull request.

Squash means replace multiple commits with a single commit. Squashing simplifies the commit history in the repository. Squashing can also be useful if you accidentally committed a change containing personal data to your local repository. You can make a second commit removing it, then squash the two. The personal data doesn't show up in history.

This article shows how to rebase and squash in one action. Two sections at the end show how to change the procedure so you can rebase without squashing or squash without rebasing.

Prerequisites

The example git commands in this article assume that you:

- Are working in a repository clone that has `origin` pointing to your fork and `upstream` pointing to the production repository.
- Have checked out a working branch based on master and named *myworkingbranch*.
- Made the working branch track its corresponding remote branch in `origin` by using the `-u` option with `git push`.
- Have added multiple commits to this working branch.
- Have pushed the branch to your fork in GitHub and created a pull request.

Set up an editor

You'll use the *interactive* option to squash. Git will create a text file with a list of commits that are in *myworkingbranch* but not master. Git opens this file in a text editor so you can specify which commits you want to keep, drop, or squash. When you save your changes to this text file and close the editor, Git proceeds to rebase and squash as directed.

Before doing squash and rebase for the first time, make sure Git will spawn an editor that you want to work with. Notepad is an option, but we recommend Visual Studio Code because it has a helpful UI for handling conflicts. The following returns a list of the global settings in your local Git config:

```
git config --global -l
```

Check the value of `core.editor`: if it's Visual Studio Code as shown in the following example, leave it as-is.

```
C:\Source\dotnet\docs [master ≡]> git config --global -l
user.name=Nancy Davolio
user.email=nancy.davolio@microsoft.com
core.editor=code --wait
```

If `core.editor` is missing or is something different, set it to Visual Studio Code.

```
git config --global core.editor "code --wait"
```

Update your master branch

It's best to rebase on the latest version of the repository's master branch, so you need to bring that version into your local clone's master branch:

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

Begin rebase and squash

```
git rebase master -i
```

This command means rebase the current branch (*myworkingbranch*, for example) on the master branch. The `-i` flag means do it interactively. If all you want to do is rebase, you can omit the `-i`. In that case, Git will rebase all commits on master without squashing them.

The command creates a text file and opens it in Visual Studio Code. The file looks like the following example:

```
pick 0e66f24b initial
pick d7ff40a5 typo
pick e9f708e2 fix link

# Rebase a3aad717..e9f708e2 onto a3aad717 (3 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label
# m, merge [-C <commit> | -c <commit>] <label> [# <oneline>]
# .       create a merge commit using the original merge commit's
# .       message (or the oneline, if no original merge commit was
# .       specified). Use -c <commit> to reword the commit message.
#
# These lines can be re-ordered; they are executed from top to bottom.
#
# If you remove a line here THAT COMMIT WILL BE LOST.
#
# However, if you remove everything, the rebase will be aborted.
#
#
# Note that empty commits are commented out
```

To squash all of the commits, change the second through the last one from "pick" to "s", as in this example:

```
pick 0e66f24b initial
s d7ff40a5 typo
s e9f708e2 fix link
```

Save your changes and close the file.

Rebasing and squashing starts, and Git reports its progress as it applies each commit. If you're rebasing after merging master into a release branch, rebase finds no changes when it reaches the merge commit. In that case, you have to do a `rebase --skip` to skip the merge commit and continue with later changes.

When Git finishes processing all of the commits, it generates another text file and opens it. In this file, you specify the comments that will go with the new commit. Here's what this file looks like:

```
# This is a combination of 3 commits.
# This is the 1st commit message:

initial

# This is the commit message #2:

typo

# This is the commit message #3:

fix link

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Wed Dec 5 11:52:28 2018 -0800
#
# interactive rebase in progress; onto a3aad717
# Last commands done (3 commands done):
#    squash d7ff40a5 typo
#    squash e9f708e2 fix link
# No commands remaining.
# You are currently rebasing branch 'myworkingbranch' on 'a3aad717'.
#
# Changes to be committed:
# new file:   help-content/contribute/rebase-and-squash.md
#
```

Make any changes you want to the commit comments, and then save and close the file.

Git applies the comments to the new commit, and the rebase-with-squash process is done.

Force push

Now you're ready to push the new rebased and squashed commit to your fork, but `git push` doesn't work. The remote repository has commits based on where the master branch was when you started. The local clone has a new one based on where the master branch is now. The `git push` command can't just add commits to the end of your fork's commits so it fails. To get past that you add the `--force` option.

```
git push --force
```

Before doing a force push, make sure you don't need to access the original commits again. They'll be permanently gone when the push is finished.

Conflicts

During the rebase action, a conflict similar to a merge conflict may happen if the current master branch includes a change in a file that one of your commits changes.

When a conflict occurs, Git opens the file in Visual Studio Code. Visual Studio Code gives you several options for handling the conflict. Here's an example of the UI:

```
Accept Current Change | Accept Incoming Change | Accept Both Changes | Compare Changes
<<<<<<< HEAD (Current Change)
It exists solely to demonstrate concepts in Joshua's blog posts:
=======
It exists solely to demonstrate concepts in git blog posts:
>>>>>>> 348a372... commit 1 (Incoming Change)
```

- **Accept Current Change** - *Current Change* is the change in master that conflicts with the commit that you're rebasing.
- **Accept Incoming Change** - *Incoming Change* is the one in the commit that you're rebasing.
- **Accept Both Changes**
- **Compare Changes** - Opens a read-only compare window.

For each conflict in the file:

- Select the best **Accept...** choice.
- If you need to preserve elements from both changes, edit the file.

After you've resolved the last conflict in the file:

- Save and close the file.
- Stage your changes and continue the rebasing process:

  ```
  git add .
  git rebase --continue
  ```

- Git may generate a COMMIT_EDITMSG file and open it in Visual Studio Code. Enter a commit message appropriate for the way you resolved the conflict, and save and close the file.

  Git applies your comment to the commit and continues rebasing and squashing until it hits the next conflict or is finished.

Rebase without squashing

You may not want to squash commits. Sometimes there are commits from other contributors alongside your commits in the same branch. If those contributors don't want to lose the record of their contribution, don't squash the commits.

To rebase without squashing, follow all of the same procedures as rebase-with-squash except omit the `-i` (interactive) flag on the rebase command. Git won't present you with a list of commits to edit. Rebasing without squashing will replay multiple commits. You may need to resolve the same merge conflict area multiple times, once for each conflict.

Squash without rebasing

One way you can squash without rebasing is to rebase-with-squash on the parent of the first commit in the branch. In effect, you're not rebasing because the rebase is on the same commit the branch is based on.

To get the SHA1 of the first commit in a branch named *squashnorebase*, run the following command:

```
git log master..squashnorebase --oneline
```

The last line in the output will contain the SHA1 of the first commit. For example:

```
b836b8f7 (HEAD -> squashnorebase) third commit
90dee821 second commit
ec8c84d5 first commit
```

Then follow the same procedures as rebase-with-squash with one difference. Be careful to rebase on that first commit's parent by using its SHA1 with a tilde (`~`) appended:

```
git rebase -i ec8c84d5~
```

Next steps

For information about how to resolve merge conflicts without rebasing, see Resolve merge conflicts in Git and GitHub.

Best practices for writing/publishing content to Docs.Microsoft.Com

5/10/2021 • 3 minutes to read

This guide provides tips and best practices for Docs.Microsoft.Com contributors to publish quality content, on time.

Hit your publish date

These tips help you publish your content on time while avoiding last-minute frenzy. Avoiding last-minute frenzy is especially important when your content is required to align with a product/service release on a specific date.

1. **Author your article in Markdown** (*.md) from day one—do not use Microsoft Word to write your article.

   Why: You avoid the time required to convert from Word to Markdown, as well as the additional vector for introducing errors when you perform the conversion.

2. Use the major or long-running changes workflow, and **create a pull request (PR) for your article on day one.** Regularly push your changes to GitHub so that you can review it on staging as you go.

   Why: You can see the rendering of your article on docs.microsoft.com and receive error notifications from the automated builds. Staging your articles helps you avoid last-minute fixes required by the PR reviewers, and get things like broken links fixed well before sign-off.

3. Use GitHub's built-in pull request review feature to perform your article review with your team members and the content team.

   Why #1: It's a slick, built-in feature of GitHub, and allows you to track all requested changes (on even a line-by-line basis) from multiple reviewers.

   Why #2: Your article is already in Markdown, on GitHub, and staged on docs.microsoft.com, so you—and your reviewers—can review the article as it will appear when it's published live.

   Why #3: You aren't authoring in Word anymore, and you need a replacement for Word's review feature.

4. C+AI/Content & Learning contributors should **start the PR review process early**, and `#sign-off` on your article at least **two days** before your target publish date.

   Why: This gives you plenty of time to integrate any changes required by the PR Reviewers team (this is when they kick back your article and issue a `#hold-off` comment). Rarely are articles approved on the first try, even for those on the content team. Give yourself time to fix and resubmit the article for review via `#sign-off` well before your target publishing date. A resubmitted article does not get placed at the top of the review queue, so you have to wait in line with everyone else.

Review content using GitHub pull requests

Use these recommendations for reviewing content in GitHub pull requests.

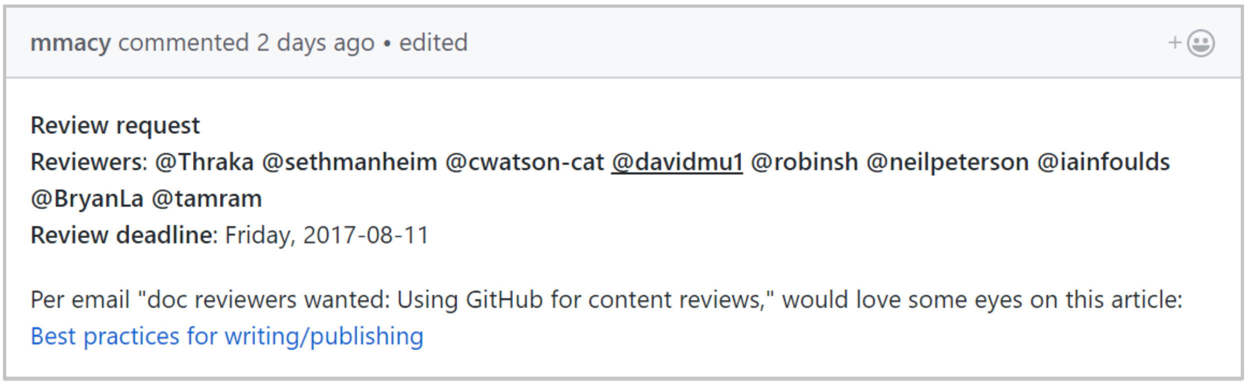
How do I invite others to review my content?

Once you've written, or updated, your article and created the GitHub pull request you can invite others to review your work and provide feedback.

To invite reviewers via your pull request, provide the following information in the PR comment section:

- **Review request**
- **Reviewers:** @githubalias1 @githubalias2 (Use @mention as the best practice to provide a record of invitees.)
- **Review deadline:** 2017-08-27
- Any other information that will help the reviewers understand your request.

For example:


mmacy commented 2 days ago • edited

Review request
Reviewers: @Thraka @sethmanheim @cwatson-cat @davidmu1 @robinsh @neilpeterson @iainfoulds @BryanLa @tamram
Review deadline: Friday, 2017-08-11

Per email "doc reviewers wanted: Using GitHub for content reviews," would love some eyes on this article: Best practices for writing/publishing

Create a PR review request template with GitHub saved replies

When you regularly solicit content reviews using GitHub pull requests, you can save time by adding the review request template to your saved replies. Add the following code block, select it from the drop down, and add the specific details.

```
### Review requested
---
- **Reviewers:** @githubalias1 @githubalias2
- **Review deadline:** YYYY-MM-DD
- **Notes**: some notes here
   more notes on a second line
---
```

How do I review content and provide feedback?

If you're asked to participate in a content review via GitHub, you review the article and provide your feedback in the pull request.

1. Start a review and add comments at the line-level
2. Submit the review
   - Leave review comments in the **Review summary** (this notifies the author)
   - Select appropriate review type (Comment, Approve, Request changes)
3. (Optional) Add *thumbs-up* or *thumbs-down* (reactions) to other reviewers' comments to indicate that you agree or disagree.

> **NOTE**
>
> If you have no changes, indicate that in the PR comment. That way, the author will know that you reviewed the content.

How do I review and incorporate suggestions from a review?

Once reviewers have provided their feedback (or the review deadline has passed) you review and integrate the appropriate changes and reply to comments.

1. Reply to review comments with:

   - Fixed
   - Won't fix (add explanation)
   - Or, ask for clarification
2. Edit and push the changes to GitHub
3. @mention the reviewer(s) to notify them that changes are ready for review
4. Iterate steps 1-3 as necessary
5. `#sign-off` on PR

> **TIP**
>
> Keep your Outlook rules in mind. For example, you might not see a notification if you use rules to route email to **Deleted Items**.

Additional resources

- Pull request best practices

GitHub for content reviews

5/10/2021 • 5 minutes to read

GitHub features pull request reviews that allow you and your team to comment on the content in a pull request (PR), approve the changes, or request additional changes before the PR is merged. Similar to the review feature in Microsoft Word, you can comment on specific sections of an article, and collaborators can comment on your comments.

Why use GitHub reviews

Now that you author content in Markdown instead of Word, you need a way to collaboratively review and comment on articles with your team. By using GitHub for reviews and keeping your content development cycle completely in Markdown, you can:

- Significantly lessen publish time
- Avoid Word-to-Markdown conversion errors
- Preserve complete review history along with published articles
- Support a consistent review model in Content & Learning

You can publish faster, eliminate errors introduced by format conversion, and your articles remain the "single source of truth."

GitHub review features

GitHub provides several review features for the content in a pull request. With GitHub reviews, you can:

- Approve the changes in a PR, or request additional changes
- Leave single (atomic) comments
- Leave comments as part of a larger review
- Reply to comments
- React to others' comments (e.g. thumbs-up)

In the following sections, we describe the different ways reviewers can leave comments on your content.

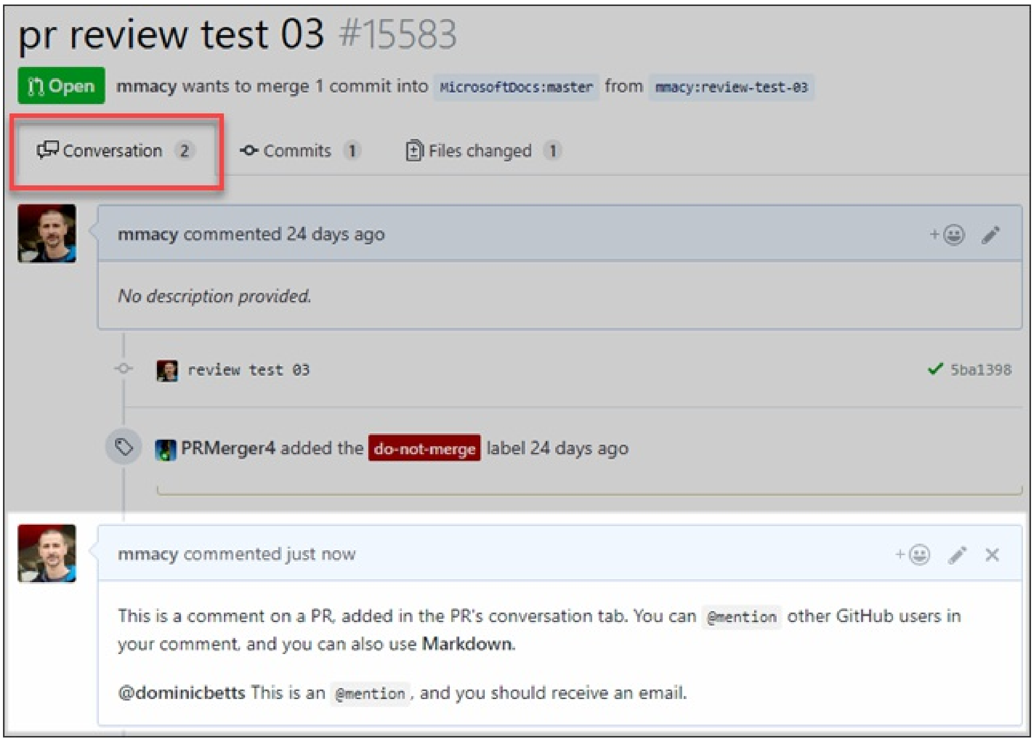
GitHub comments

Whether feedback is provided as part of a review or a simple one-off, reviewers use the comment feature of GitHub to do so. You can add both pull request comments and line comments. Line comments can further be added as part of a larger review.

You can *@mention* other contributors in your comments, as well as use Markdown. Mentioned GitHub users will receive an email containing your comment (use GitHub aliases when you *@mention*), and your comment's Markdown is rendered once you submit the comment.

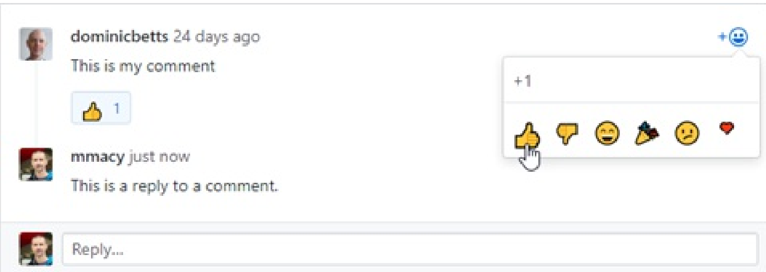
Pull request comments

To comment on a PR as a whole, add a comment to the PR's *Conversation* tab to add a **pull request comment**.



A comment in the Conversation *tab of a PR*

You can reply and react to any comment, even your own. Here's a comment with a reply and one "thumbs-up" reaction. Reaction selection is also shown.

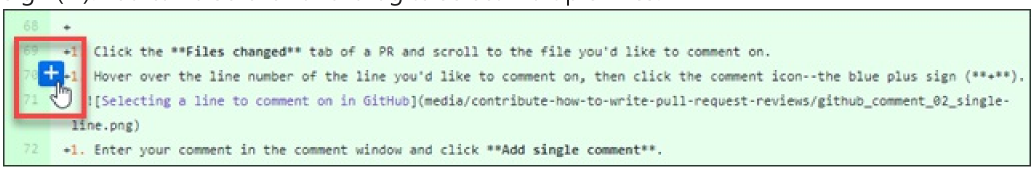


Reacting to a comment

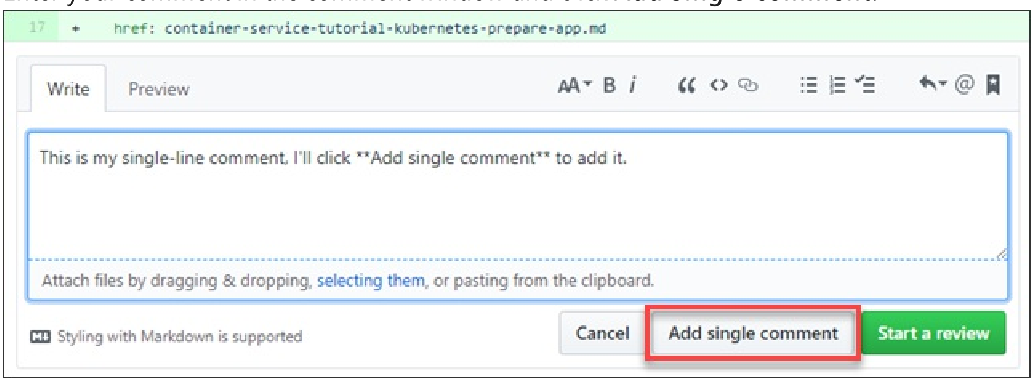
Line comments

You can add a **line comments** to call out on a specific section of a file. To add a line comment:

1. Click the **Files changed** tab of a PR and scroll to the file you'd like to comment on.
2. Hover over the line number of the line you'd like to comment on, then click the comment icon--the blue plus sign (**+**). You can also click-and-drag to select multiple lines.



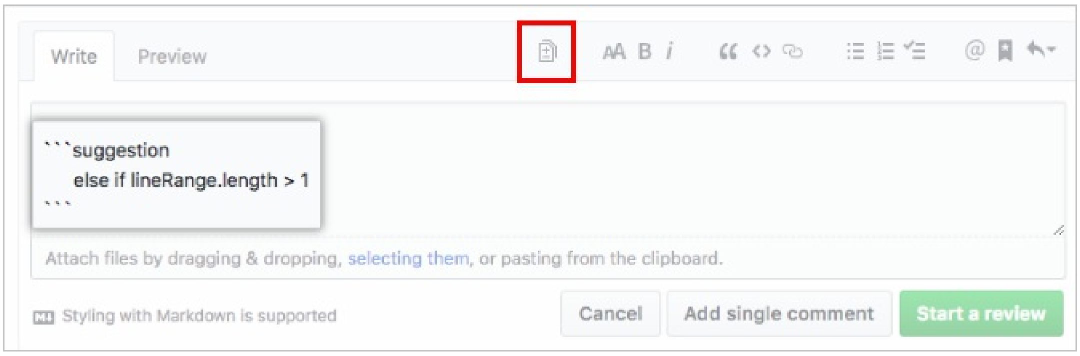
3. Enter your comment in the comment window and click **Add single comment**.



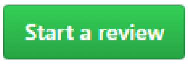
Suggest changes

In addition to commenting on a line, you can suggest specific changes. After you've selected the lines you'd like to comment on:

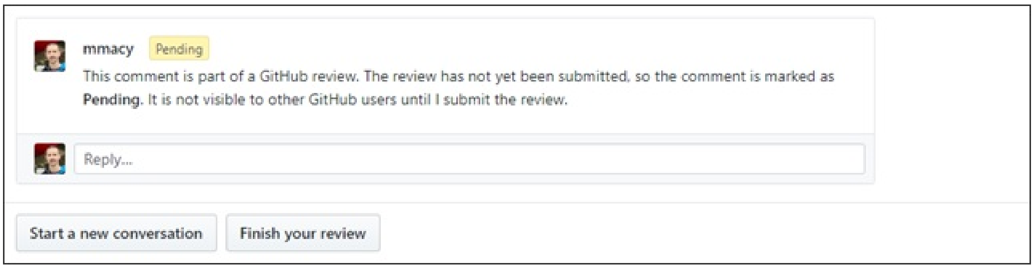
1. Click the documents-with-plus-sign icon to the left of the font picker, then edit the text within the suggestion block.



GitHub reviews



Reviews allow you to add several comments and submit them as a group. You can add as many comments as you like, each of which remains *Pending* until you submit the review. Pending comments are invisible to other GitHub users, and only become visible when you submit the review.



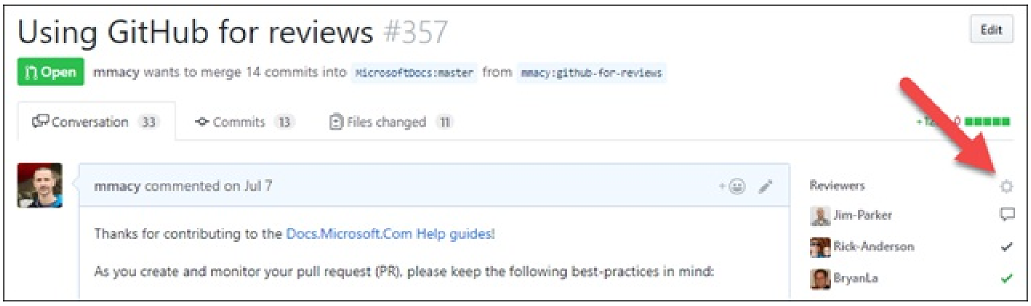
Invite reviewers

There are several ways you can invite others to review the content in your PR. Two of them, @mentions and email, are always available. A third, GitHub's *Request a review* feature, is available only if you have write permission to the repo.

- **@mention** (recommended): Add a PR comment and @mention your reviewers with their GitHub aliases. You can find Microsoft-linked GitHub aliases in the GitHub contributor alias search. An @mention in a comment acts as a permanent record of reviewers directly in the PR, and your mentioned reviewers automatically receive an email notice from GitHub when you submit the comment. TIP: Use Markdown in your PR comment to provide friendly links to the staged article(s) in your review request (`[article title](https://review.docs.microsoft.com/.../article.md)`).

- **Email**: You might also wish to send an email directly to your reviewers. Provide a link to the PR, as well as links to the staged articles you'd like them to review.

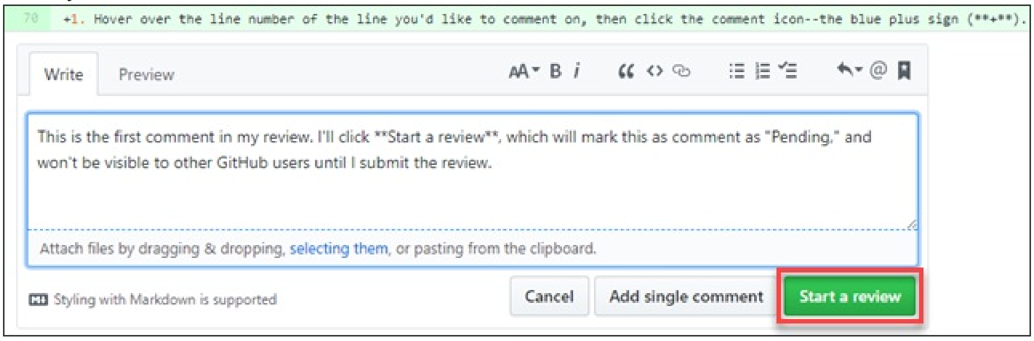
- **Request a review** GitHub feature: This feature is available only if you have *write permissions* to the repo. If you have write permissions, a small gear icon appears under the **Reviewers** heading on the right-hand side of the PR. Click the gear and enter a GitHub alias to invite that person as a reviewer; they'll receive an email containing the review request.



Start a review

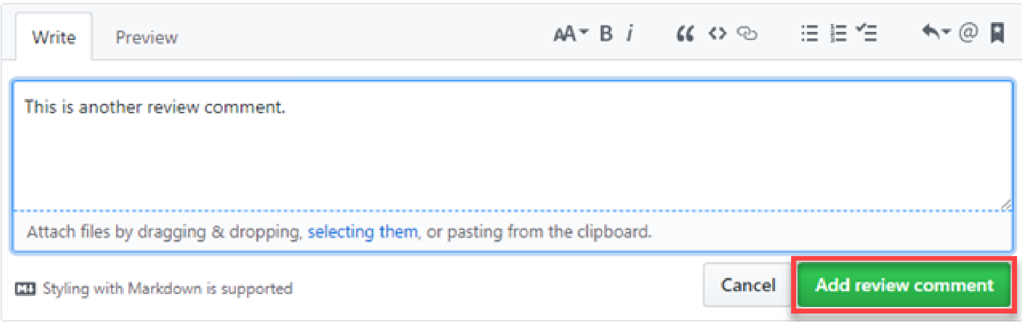
As a reviewer, start your content review by performing these steps:

1. Click the **Files changed** tab of a PR and scroll to the first file you'd like to comment on.
2. Hover over the line number of the line you'd like to comment on, then click the comment icon--the blue plus sign (**+**).
3. Enter your comment in the comment window and click **Start a review**.



Add review comments

To add additional review comments, repeat the steps to add a line comment, then click **Add review comment**.

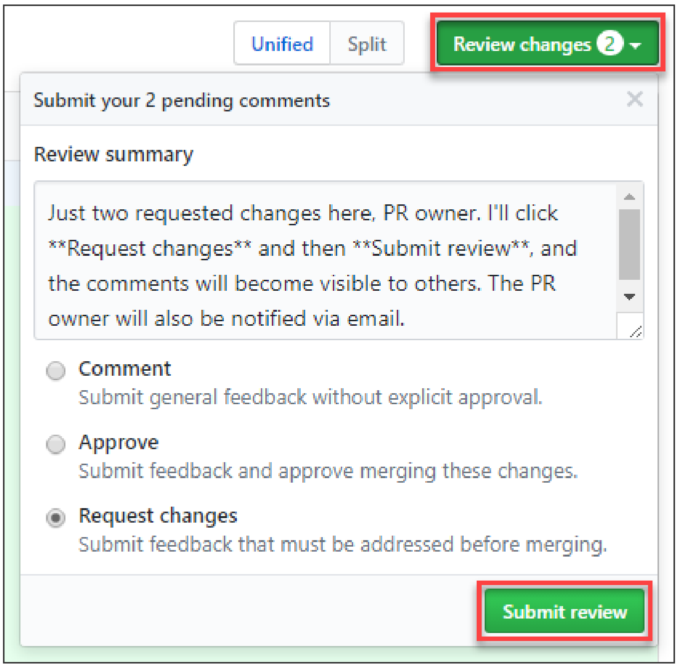


Add as many comments as you like to your review. Each comment is private and can't be seen by other GitHub users until you submit the review.

Submit a review

Once you've added all your comments to the review, submit your review.

1. Click the **Review changes** button in the upper-right of the *Files changed* tab to open the review submittal dialog.
2. Enter a descriptive **Review summary**.
3. Select the radio button for your desired review type, then click the **Submit review** button.



The pull requestor will receive notification of the review, and all of its comments can then be addressed as described above.

> **NOTE**
>
> Unless your GitHub repo enforces branch protections and required reviews, "Request changes" does not actually prevent a merge. Most of our repos do **not** enforce branch protections and required reviews. As such, "Request changes" merely notifies the PR owner that you're requesting changes they should integrate before signing off on their PR.

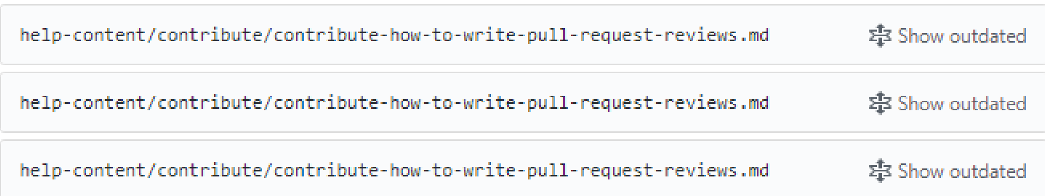
Respond to a review

When another GitHub user submits a review on your PR, you'll receive an email containing all the comments in the review. You are then free to navigate to the PR in GitHub and respond to those comments however you see fit. You can:

- Integrate changes suggested in the comments
- Reply and/or react to comments

If you agree with a comment, consider adding a "thumbs up" emoji rather than writing text, which triggers an unnecessary notification to the reviewer.

When you make changes to a line a reviewer commented on, and subsequently push your changes, the comment is shown as collapsed in the *Conversation* tab of your PR.



More info

Here are a few links to GitHub's documentation on commenting and reviews.

- Reviewing changes in pull requests
- Commenting on a pull request

- Reviewing proposed changes in a pull request

Git and GitHub glossary

6/1/2021 • 6 minutes to read

This article is a reference for common terminology and concepts that writers and contributors who work with GitHub will come across.

Acrolinx

A tool that checks spelling, grammar, style, tone, clarity, and key terminology use. It runs automatically for some repositories when you create a pull request. Acrolinx has a Visual Studio Code extension that enables you to run its checks on an article before you create a pull request.

Branch

A parallel version of a repository (or a fork or clone) that lives within the repository itself. A branch contains the actual files and folders that make up a project's content set. Branches are insulated from each other so you can edit a file in one branch without affecting that file in another branch. Branches are used to separate streams of work, and contributions are always made to a specific branch.

All repos contain a default branch (sometimes called *master* or *main*) and one or more branches destined to be merged back into the default branch. The default branch serves as the current version and single source of truth for the project. It's the parent from which all other branches in the repo are created.

Clone

A copy of a repository, including branches, that lives on your local hard drive as a folder. Often referred to as your "local repo" or "local branch." When you're working, the files that you're actually editing are in your cloned repository.

Commit

Confirmed changes made to your local working branch. Committing to GitHub is similar to saving a file to your hard drive. However, each time you commit changes, GitHub assigns a unique ID to the commit that tracks what changes were made, when, and by whom.

Default branch

The default branch that's created with every repository (and clone and fork). The default branch may be named *master* or *main*.

> **NOTE**
>
> You should **never** work in the default branch. Always create a *working branch* and then *merge* your edits into the default branch.

The default branch serves as the current version of content for the project. The content in the default branch aligns loosely with the organization of the articles in the corresponding Docs pages. Subfolders are used to separate articles, media content, and "include" files, which allow writers to reuse content.

Within the root folder, you'll find general articles about the service or product. There's also usually a series of subfolders that correspond to features, services, or common scenarios. Each writing group decides how to organize its folders.

Fetch

A Git command that copies changed files from a repository to your current branch *but doesn't merge them*. You'll usually see this command in the GitHub Desktop toolbar as **Fetch origin**.

Fork

A copy of a repository, including branches, that lives in your personal cloud space on GitHub. If you don't have write permissions to a repository, you can "fork" that repository and have full write permissions to the fork. Then, you can create a pull request to have an admin sync your changes back to the repository.

Git

An open-source, centralized version-control system that facilitates project collaboration through distributed version control of files that live in repositories. Git makes it possible to integrate streams of work done by multiple contributors over time for a given repository.

Unlike other centralized version-control systems (like SharePoint or Visual SourceSafe), Git has a unique contribution workflow and terminology to support its distributed model. For example, there are no files, which are usually associated with check-in/check-out operations. Instead, Git is concerned about changes at a finer level of granularity, comparing files byte by byte. Git also uses a tiered structure to store and manage content for a project. A repository is the highest unit of storage and contains one or more branches, which is where the actual files and folders for a project live.

Contributors interact with Git to update and manipulate repos locally using tools like GitHub Desktop and on GitHub, which integrates Git to reconcile contributions flowing back into the main repo.

GitHub

A web-based hosting service for Git repositories, such as those used to store docs.microsoft.com content. To edit and submit files, you must have a GitHub account.

All workflows begin and end at the GitHub level, where the main repo for any given Docs project is stored. Contributors create copies for their own use. These copies are distributed across multiple computers and eventually reconciled back into the project's default GitHub repo.

GitHub Desktop

A cross-platform GUI application for GitHub.

Label

A flag that's attached to a pull request in GitHub that indicates the status of the request.

Markdown

A user-friendly markup language that supports basic text formatting (like bold text, lists, headings). Markdown files use the file extension *.md*.

Merge

A command that combines the contents of two versions of a file.

Merge conflicts

Conflicts that can occur when two or more people have modified a file and try to merge it with the default branch. Some conflicts can be resolved automatically by GitHub, while others need to be manually merged. Keeping your branches synced regularly can help minimize merge conflicts.

Origin

When you clone a repository, the repository that you cloned ***from*** is called the origin. The origin is always relative to the branch you're currently in. For example, if you are in the default branch of your clone, the default branch on the primary repository is the origin. If you're in your working branch of your clone, the working branch of the primary repository is the origin.

Pull

A command that copies changed files **from** a repository **to** your current branch and merges them.

Pull request

The process of submitting a set of changes (also known as commits) to be merged from one branch into another branch (usually the default branch). The commits are stored in a contributor's working branch, allowing GitHub to first model the effect of merging them into the default branch. A pull request serves as a mechanism to provide feedback from the build and validation process to the contributor. The pull-request reviewer can resolve potential issues or questions before the changes are merged into the default branch. After the review is complete, anyone with *Write* permissions can merge the pull request.

Push

A command that copies changed files **from** your current branch **to** the target repository and merges them.

Repository

A storage area (think of it as a folder) on GitHub where all the files for a project are stored. Colloquially referred to as "repos." For example, the *OfficeDocs-ProjectServer-pr* repo stores the content and media files for Project Server and Project Online.

Repos can be *public* or *private*. Public repos are accessible to everyone using GitHub, while private repos are accessible only to Microsoft full-time employees and vendors. Most projects that publish to docs.microsoft.com use a separate private repo for internal contributors and a replica public repo for external contributors. Usually, private repos use the same name as the corresponding public repo but have "-pr" affixed to the end (as in *OfficeDocs-ProjectServer-pr*). As an internal contributor, you should be working with the private repo for your content area.

Stash

Command to temporarily shelve (or stash) uncommitted changes if you need to switch what you are working on. Stashed changes can be reapplied later.

Upstream

A name for the primary repository that contains the final changes for project files. Like origin, it's relative to your current branch. For example, say you have an environment with a primary repository, a fork, and a clone. Relative to your clone repository, the primary is the upstream and the fork is the origin. If you cloned directly from the primary repository, the primary is both the upstream and the origin.

Visual Studio Code

The editor we generally use to create and modify Markdown files. Colloquially known as "VS Code."

Working branch

A branch a user creates to isolate changes from the default branch.

How to recover a lost GitHub account

6/1/2021 • 2 minutes to read

This article discusses options for recovering your GitHub account if you can't sign in or are locked out.

Use your recovery codes

One requirement for contributing to private/internal Docs repos at Microsoft is to enable two-factor authentication (2FA) on your GitHub account. As part of this process, you should have downloaded and saved your 2FA recovery codes. GitHub creates 16 recovery codes for you. You can use these codes to authenticate to GitHub if you lose access to your 2FA credentials (for example, if you lose your phone where you configured 2FA).

Once you use a recovery code to regain access to your account, it cannot be reused. If you've used all 16 recovery codes, you can generate another list of codes. Generating a new set of recovery codes will invalidate any codes you previously generated.

Use another recovery option

GitHub provides several ways to recover your account if you lose access to 2FA. These include authenticating with:

- A fallback number
- A security key
- A verified device, SSH token, or personal access token
- An account-recovery token

Many of these options must be configured *before* you lose access to your account.

Create a new account

If none of the above methods succeed, follow these steps to create a new account.

Caution

If you disassociate your Microsoft corpnet and GitHub accounts, you may lose permissions and group memberships. If you have private repos created as forks, those will be deleted. And there may be other unexpected side affects related to your GitHub data. *Simply relinking the accounts does not undo these changes.* You will have to onboard from scratch. Use this procedure only as a last resort.

1. Go to https://repos.opensource.microsoft.com/link.
2. Select **Remove your link**, and then confirm. This action unlinks your GitHub account from your Microsoft corporate account.
3. Follow the instructions at Sign up for GitHub to create a new account.
4. Return to https://repos.opensource.microsoft.com/link and link your GitHub account with your Microsoft account. (If your accounts are already linked, the page will show "You're linked!")

Triaging new GitHub issues

6/1/2021 • 13 minutes to read

In this article you will:

- Learn about the triage process you need to use when you and your team work with GitHub issues.
- Understand the workflow and issue classification system we are using
- Learn how to work with issues in our open-source projects

Whether you're working in a large docset or a small one, you need to have a plan for triaging incoming issues.

Planning for triage means:

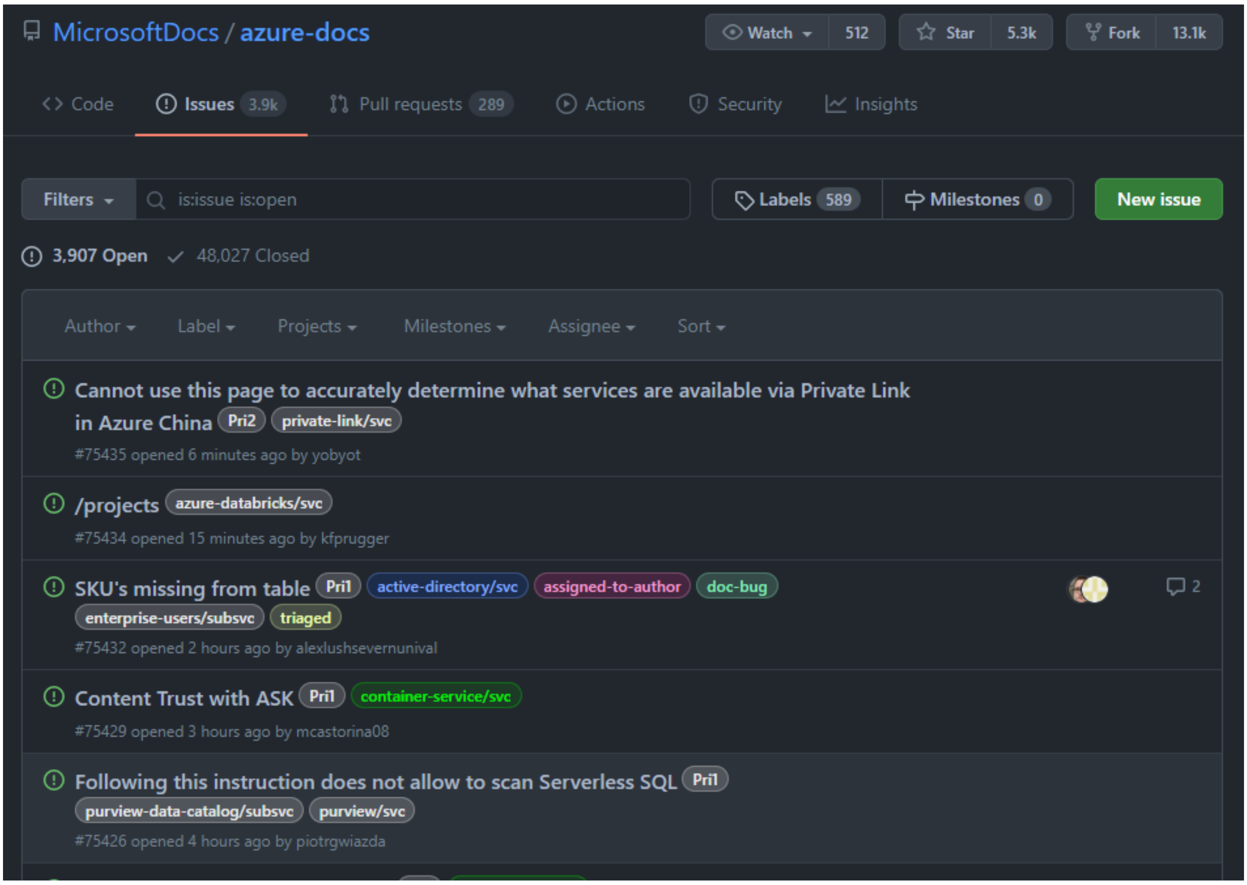
- You have people who are assigned to triage and respond to new issues regularly, daily if possible.
- Team members responsible for triage understand the issue classifications and process described later in this article; using our standard labels enables reporting and it allows you to identify feedback that needs to be handled by the right content partners or feedback endpoints.
- You have identified support channels you can direct product questions, support issues, localization feedback, site design feedback, and other feedback that isn't documentation related.

Types of feedback

If enabled, each topic on `docs.microsoft.com` gives readers the option of submitting feedback on either the topic itself or the product. This choice improves the flow of information directly to the product team while reducing the amount of non-content feedback "noise" we receive and have to triage.

- Selecting **This product** takes readers to a destination (for example, UserVoice, a GitHub repo, an email address) that allows them to provide feedback on the product itself. This feedback is independent of the content and has no relationship back to the original topic. The intent is for the product team to own this feedback process from end to end. This article doesn't discuss this type of feedback.
- Selecting **This page** opens the commenting user interface for the reader. They'll sign in to GitHub and submit feedback on the page.
- Selecting **View all page feedback** allows readers to see all feedback submitted on the topic. Anyone can read the feedback, but readers must sign in to leave feedback.

When a reader submits feedback on a topic, GitHub creates an issue in the public repo that contains the topic's docset. The issue links to the submitter of the feedback and the topic it was submitted on.



Alternatively, readers may create issues directly in your repo using GitHub. Regardless of how readers submit issues, triage them the same. Issues created in GitHub will lack a topic "Details" section. If the issue contains insufficient information, assign it to the creator and assign the **needs-more-info** label from the list of labels below.

Triage process

One of the advantages of building the docs.microsoft.com feedback system on top of GitHub issues is that all the patterns, practices, features, and tools available through GitHub are at your disposal to help you maintain your content, address your issues, and ensure your customer's satisfaction.

One of the most powerful tools in the open-source project maintainer's toolbox is issue and pull-request (PR) triage. Consistent triage, categorization, and directing non-doc issues to proper channels can be a great way for a content team to start the day and to ensure all incoming feedback receives a response. Whether it's a single person or an entire team committing their time to triage, you need a clearly defined process to ensure your team addresses all customer feedback and follows up in a timely fashion.

The triage process involves:

1. Assigning issues to yourself or others.
2. Labeling and classifying issues.
3. Replying to issues.
4. Staying engaged.
5. Closing issues.

This article details each step of the process below.

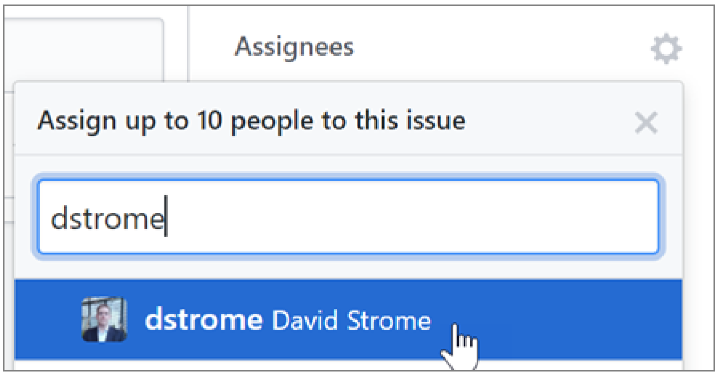
Assign issues

Depending on the repo, issues may be assigned to you by the repo admin, or you may need to assign yourself or someone else on your team. Either way, you'll receive a notification for the issue. Assignment is important because each issue needs an owner to drive it to resolution. Being assigned to an issue will send more notifications to the assignee when someone comments on the issue on docs.microsoft.com or on GitHub.

To assign an issue to yourself or someone else:

1. Navigate to the issue. Select the gear icon next to `Assignees` and type in your GitHub username (or the username of the person you want to assign this issue to) in the `Filter people` textbox.

2. Select your GitHub account when it appears in the list. Your GitHub username will appear in the `Assignees` list.



You're now assigned to the issue. If you assigned others, they'll be notified that you've added them to the issue. You can assign up to 10 people to an issue.

Label and classify issues

Labeling is key to triaging GitHub issues filed by customers against our content. Regular triage using our standard label set will improve your understanding of your GitHub documentation issue backlog. Regular triage also allows you to quickly direct customers to the correct location for their feedback.

Standard classification labels for GitHub issues

When you triage issues, you need to label each issue for its type - you need to classify them. Every team should align to the standard-issue classification labels when triaging issues created against their repo. The standard labels not only make it easier for your team to manage its issues but also supports standard concepts in broader discussions and our reporting.

Issues not related to documentation

- `code-of-conduct` - Assign to issues that are spam, trolling, or that otherwise violate Microsoft's code of conduct.

- `support-request` - A support-style question where the customer needs help with solving a problem. GitHub label color. Explain that a dedicated channel exists for their support request and direct them to it. Once you've provided the customer this information, you should close the issue.

- `docs-experience` - The issue contains feedback about the docs.microsoft.com experience or design. The customer should be directed to this location and the issue closed.

- `product-feedback` - Indicates that the feedback was related to the product itself. Explain that a dedicated channel exists for their feedback and direct them to it. Once you've provided the customer this information, you should close the issue.

- `product-question` - Typically, these questions are tightly focused on product capability or functionality, they aren't a general consulting request. Once you've provided the customer this information, you should close the issue.

- `needs-more-info` - The issue assignee, author, or product team has requested the customer provide more information about the issue they submitted before the issue can be classified into one of the following categories. These issues are assigned to the customer who filed the issue.
- `duplicate` - A duplicate of another piece of feedback.

Documentation-related issues

- `doc-bug` - If the feedback is about something that is out of date, unclear, confusing, or broken in the article, assign the bug label. This includes feedback about missing information that is blocking the customer's success.
- `doc-enhancement` - If the feedback is about suggested additions/improvements to the article, but there is no evidence the customer was blocked, assign the improvement label.
- `doc-idea` - The issue is a request to collaborate on or contribute new documentation.

Product or Service area labels

If your repository contains articles about different services or technology areas for a single product, you may want to use the GitHub automation app to have PR Merger apply labels for these values when a customer creates a new issue.

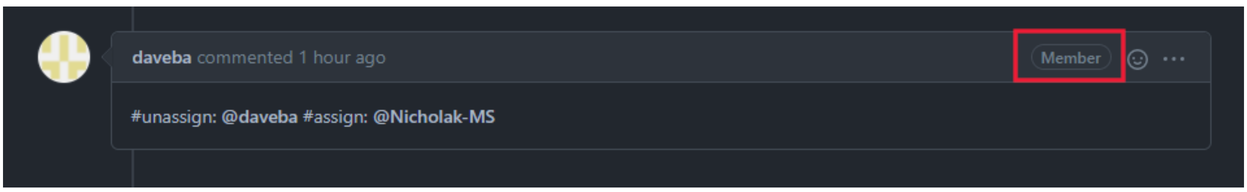
> **NOTE**
>
> The GitHub Issue and PR report uses both topic metadata and applied labels to enable sorting by product and service labels so using this automation is not a requirement.

Priority labels

The GitHub automation app can also apply Pri1 – Pri3 labels to issues based on the number of page views each month. If you want to sort your issues by priority in the GitHub Issues and PR report, you'll need to enable this automation.

Reply to issues

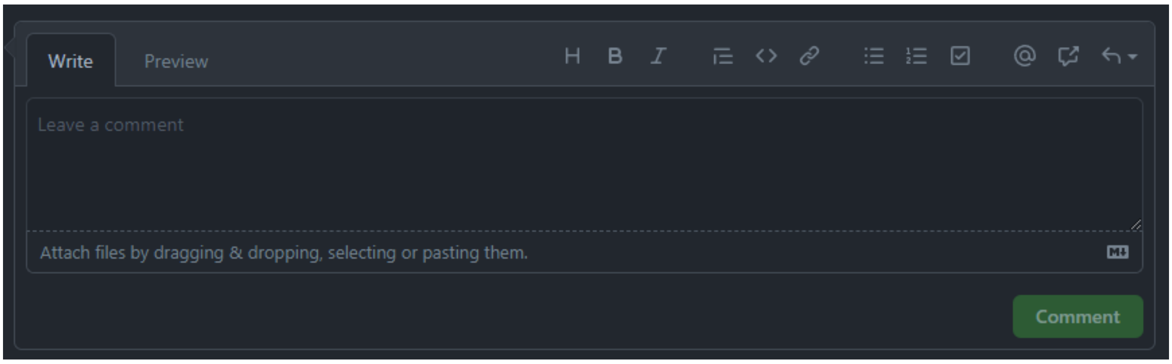
Anyone who has a GitHub account can reply to comments on docs.microsoft.com and on GitHub. Microsoft employees and vendors who are members of the MicrosoftDocs or Microsoft organizations have a "member" badge next to their name on their replies:



To reply to an issue:

1. Click on the link in the notification for your issue.
2. Log into your GitHub account if you're asked to do so.
3. Write a response in the editor field. In this field, you can:
    - Use Markdown by entering Markdown syntax or using the formatting bar at the top of the editor.
    - Attach files by dragging them into the editor field.
    - @-mention other GitHub users. People mentioned in this way will receive a notification.

4. Optionally, select **Preview** to see how your reply will look.
5. Click **Comment** to post your reply.



When you submit your reply, it will show up immediately as a reply to the original comment on the topic on docs.microsoft.com.

Remember that when you reply to an issue, you're speaking to Microsoft customers. Use the guidelines below to guide your response.

- **Keep the message concise** – Being concise exudes confidence and shows the customer that you know what you're talking about. Only explain what answers the customer's question, but ensure you have not left anything out. Concise doesn't mean short; it means ensuring you communicate just what's needed to make your point.
- **Be honest** – Honest answers build trust with our customers. The answer may not be the one the customer wants, but if we build trust with them, they'll have an easier time accepting an answer from us because they'll know it to be honest and true.
- **Be inquisitive** – If a customer is suggesting an idea to us on how something could be better, take some time to really understand what they're asking for. If we don't completely understand, ask them questions to clarify what they mean. **Example**: "You mentioned that instead of doing X, you did Y. What was the reasoning for doing Y? Can you expand on your current situation?"
- **Thank them** – Begin the response by thanking the customer for taking the take to make our content better. **Examples**: "Thanks for reaching out," or "Thanks for the question."
- **Acknowledge them with empathy** – If the customer is upset or struggling with using the content, it's important that we acknowledge their question. **Example**: "I'm sorry to hear that you're having a difficult time with our content."
- **If they're upset, align with the customer** - If a customer is upset, it's important to let them know that you hear their frustration and understand how hard it is for them. Careful consideration should be taken into account when aligning with the customer. Align with them only when it is obvious they are upset; otherwise, it may come across as pandering. **Example**: "I know that trying to learn a new technology is difficult, and confusing content can make the task even more challenging."
- **Know when to get help** – Some issues may require a response that you don't have. An example might be: "When is product X going to GA?" A response to this question has many implications. Careful consideration and discussion with the engineering team or other partners may be needed. Additionally, the question or issue may require another individual to respond to the customer. If so, contact to the person outside of the thread to ask for support.

Stay engaged

Usually, people who submit issues on GitHub are invested in the site and engaged in its community. They expect the same of us. Continue to monitor the issues you own after you've replied. Below are tips for staying engaged.

- Check your open issues daily for new replies, and respond as needed.
- If you need to research or follow up with others to resolve the issue, make your intentions clear when you first reply to an issue. Don't let more than a few days pass between updates. If it's taking longer than expected to get a response to an issue, update the thread letting the submitter know. Frequent communication builds trust with the customer.
- Know when to quit. Some customers may continue the conversation after you've closed the issue or if they're upset with the outcome. Carefully read their response to see if it requires a response. If the customer is venting or repeating their question, a response may not be warranted. If the situation gets too uncomfortable, contact your manager to review the response. If needed, we can always include CELA to help us determine how or if we should respond.

Close, edit, or lock issues

Closing an issue is equivalent to resolving it. When you close an issue, it will be moved from the **Open** list of comments on a topic to the **Closed** list. Closed issues on GitHub are filtered from the active list but can be accessed using the **Closed** filter.

Close issues when:

- The submitter agrees the issue is resolved.
- You've updated the content to reflect the feedback in the issue.
- The feedback isn't actionable, and the submitter hasn't responded to requests for clarification.

To close an issue, post a reply politely indicating why you're closing the issue. Next, depending on the repo, either select **Close and comment**, or use the hashtag command `#please-close` command in the issue comments.

If a customer follows up after the issue has closed and disagrees with your decision to close the issue, continue to engage with the customer to drive towards a mutually satisfactory resolution. When in doubt, reopen the issue and continue politely towards resolution.

Editing customer comments is discouraged and should be a rare occurrence. If for some reason you do edit a customer's comment, add a note that you edited it and describe what was changed.

Lock issues only if they violate the Code of Conduct. The community does not look favorably upon locking a conversation, so you must involve leadership in the decision. For more information, see Code of Conduct.

Azure-CXP maintained repos

In addition to the standard label set, the common label is a triage status indicator for public PRs:

LABEL	DEFINITION	ACTION REQUIRED
assessing_(C&L)	A PR reviewer has taken action and it's assigned to the author or another SME	No action required by the assignee. For the team helping to do assignments, it saves us from having to open lots of PRs, since we can easily see they are already in progress

Request an issue be assigned to another user or to another category

As repo readers, if you know an issue is assigned to the wrong person, you can reassign issues (`#assign:<GitHubID>` , `#reassign:<GitHubId>`), set priority (`#pri1` , `#pri2` , `#pri3`), or update labels. For more information, see Hashtag commands for managing issues with Read permissions.

If you need assistance with issue assignment, email the AzureCXPDocFeedback alias.

If you feel an article has been incorrectly categorized as a documentation-related issue, email the AzureCXPDocFeedback alias. Explain why you feel the article has been miscategorized, and indicate what the category should be. The CXP team will assign the `category-reassigned` label to these issues and assign the correct category label.

Contact Azure CXP team for issues in the azure-docs repo

To contact the Azure CXP team, send mail to AzureCXPDocFeedback alias.

Code of conduct violations (code-of-conduct)

The Microsoft Open Source Code of Conduct lists the following behaviors as disrespectful and unacceptable:

- Violent threats or language.
- Discriminatory or derogatory jokes and language.
- Posting sexually explicit or violent material.
- Posting, or threatening to post, people's personally identifying information ("doxing").
- Insults, especially those using discriminatory terms or slurs.
- Behavior that could be perceived as sexual attention.
- Advocating for or encouraging any of the above behaviors.

Trolling and spamming are also behaviors that are in violation of the Microsoft Open Source Code of Conduct. For more information, see the Code of Conduct FAQ, or contact opencode@microsoft.com with any questions or comments.

For guidance on how to handle cases where someone uses the feedback system in a way that violates the Code of Conduct, see Respond to inappropriate content (GitHub issues and comments).

Other GitHub-related features you can use with issues

Associating PRs with related issues

In your PR, you can use keywords in your commit message so that GitHub automatically closes the related issue. For the full list of keywords and instructions, see Linking a PR to an issue using a keyword. The keywords work across repositories, so you can use them in a private repo PR to close a public repo issue. The customer who filed the issue will receive a GitHub notification when the issue is closed in this way.

Notifications

When customers create an issue, they'll automatically receive email notifications on that issue. You can @-mention GitHub users in the issue description and comments, which will send those individuals email notifications as well. For any issue, you can visit the Issues page on GitHub and set the level of notifications you want or manually subscribe to that issue. If you want to be notified for a broader set of activity, you can subscribe to all issues on a repository, or you can subscribe to all activity for a GitHub organization using RSS.

Resolving "stale" GitHub issues

3/5/2021 • 4 minutes to read

This article defines the 'stale' issue concept and provides guidance for writers and teams to meet the challenge of managing the backlog of GitHub issues.

When is an issue considered 'stale'?

An issue is considered *stale* if it was created over 180+ days ago and has no new comments for 15+ days.

However, beyond simply using **created date** as your filter, you should take a moment to consider situations where an issue is stale, but the customer may still be 'checking in' on it. Taking recent activity, priority, or upvote/comment activity into consideration could help you avoid closing active issues on customers.

Scenarios

Since the goal of this process is to help our writers direct their available time to newer issues, we need to look for opportunities to not invest in 'stale issues'.

In all scenarios there should be a focus on acknowledging the experience the customer had, providing context, and concluding the exchange.

This process **should not be automated** and provided boilerplate examples should be adjusted as needed to fit the content of the issue.

The following scenarios categorize the different kinds of stale issues and suggested response messages to use when closing issues.

Scenario 1: Stale issue that is not triaged or assigned, but is still open

In situations where there has been no clear engagement on the issue, you have the option to close out.

Boilerplate example:

```
Thanks for your dedication to our documentation. Unfortunately, at this time we have been unable to review
your issue in a timely manner and we sincerely apologize for the delayed response. We are closing this issue
for now, but if you feel that it's still a concern, please respond and let us know. If you determine another
possible update to our documentation, please don't hesitate to reach out again. #please-close
```

Alternatively, you could also comment on the stale issue to ask if the issue is still current, and if a response is not received within a time range of your choosing, you can close the issue.

Scenario 2: Stale issue that is a doc-bug, doc-enhancement, or doc-idea that was fixed, but the issue was left open

This scenario is confirmed as fixed, either through issue comments or while triaging an untriaged stale issue.

> **NOTE**
>
> If an issue has already been triaged and comments do not identify resolution, it should be handled with Scenario 3

Boilerplate example:

```
Thanks for providing feedback that helps improve our documentation. This issue has been resolved and we are
closing the issue. #please-close
```

Scenario 3: Stale issue that is a doc-bug, doc-enhancement, or doc-idea and assigned to an owner, but not fixed

Similar to scenario 2, but in this instance the issue is not fixed.

Here you should review the potential doc action and decide if the work is already present in an existing work item for content, and if so, add the feedback to the item. Otherwise, create a work item.

The goal here should be to indicate why we are closing the issue, acknowledge, in a measured way, the low-quality feedback experience, and indicate that the issue raised is now being tracked internally.

Boilerplate example:

```
Thanks for your dedication to our documentation. Unfortunately, at this time we have been unable to review
your issue in a timely manner and we sincerely apologize for the delayed response. The requested updates
have not been made since the creation of this issue, and the timeline for resolution may vary based on
resourcing, so we've created an internal work item to incorporate your suggestions. We are closing this
issue for now, but feel free to comment here as necessary. #please-close
```

Scenario 4: Stale issue that is not doc related, but the writer or product team engaged

As an organization, we are not resourced to address or direct feedback that isn't doc-related. Typically issues of this type would be redirected quickly after they are created.

However, the last thing we want to do is close the door on an open engagement. If an assigned *owner* has clearly engaged the issue, the first step is to comment to the *owner* for a follow-up.

If the assigned *owner* is no longer in role, or at Microsoft, see Scenario 5.

Scenario 5: Stale issue that is not doc related and there is no clear engagement

The goal here should be to acknowledge the untimely response, explain why this issue type isn't within scope, and provide the correct channel for future feedback of that nature.

> **NOTE**
>
> Proper product and support feedback channels differ from team to team, so make sure the resource you provide is relevant to your space.

Boilerplate example:

```
Thanks for your dedication to our documentation. Unfortunately, at this time we have been unable to review
your issue in a timely manner and we sincerely apologize for the delayed response. Though the scope of our
feedback channel here on GitHub covers specific documentation fixes, we can help redirect you to the right
support channel to get an answer to your question: [reference to alternative product/service or community
endpoint]. Because this issue does not relate to documentation, we are closing this issue. #please-close
```

Pull request comment automation

4/9/2021 • 2 minutes to read

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

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

See the **Pull request processing** sections of the minor/infrequent and major/long-running contribution workflows, for more context on how comment automation is used in each.

PR (pull request) review quality criteria

5/13/2021 • 22 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.

Blocking content quality items

The updates in the pull request must meet the following criteria to be merged. Pull request reviewers provide feedback in pull request comments for these items and type `#hold-off` in the pull request to return it to you (the author) with feedback.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Prerequisites	The "ready-to-merge" label is assigned to the PR (applied from automation after the `#sign-off` comment), and the validation status is "passed."	Ensures the author intends to hand off the PR for review and that the PR has passed the build tests.
All	Prerequisites	Close any PR against the live branch. The user should be redirected to the master branch or to a release branch.	Only publishing PRs are allowed against the live branch.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Learn	Prerequisites	For most contributors, the initial publish of a new module must merge to the master branch directly from a release branch, and content development work for a new module must merge to a release branch. A PR that targets the master branch with a new module from a personal fork is allowed only from contributors that have been vetted by the Learn team.	Prevention of live site incidents. Too many modules have been submitted against the master branch and accidentally #signed-off as ready by the author (because they didn't understand the implications of pointing to master), but the modules were not ready. This causes publishing escalations to deprecate the content from the site, implement redirects, and rebuild the release branch.
All	Prerequisites	The PR cannot be blocked by a merge conflict. If a merge conflict exists, refer the user to Resolve simple merge conflicts on GitHub for instructions on how to use the GitHub UI to resolve merge conflicts. PR reviewers don't resolve merge conflicts for contributors.	Impossible to merge, sets expectation for who is responsible for resolving the conflict.
All	Prerequisites	The `#sign-off` comment must appear after the validation results and staging links. If the `#sign-off` comment appears before the staging links, the author didn't review the staged content.	Process requirement to help encourage authors to review staged content.
All	Prerequisites	All articles in the PR must have an Acrolinx score of 80 or higher (where Acrolinx enabled in the PR queue). Limited exceptions are available.	Helps ensure baseline editorial quality while allowing wiggle room for items that are correct but not in the Acrolinx dictionary.
All	Repo integrity	The PR contains no obvious content regressions, such as unintentionally reverted dates, branding changes, etc.	Helps ensure content integrity.
All	Repo integrity	No article-related files, images, or folders are being added to the root directory of the repo.	Helps support appropriate repo management and folder structure so the root does not become cluttered.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Repo integrity	The repository administrator and FTE program manager have to review PRs that make a non-routine change in a configuration file or in the root folder, such as `docfx.json`, `.openpublishing.publish.config.json`, and any `.git*` file. For example, the admin needs to review PRs that enable a new content type. Routine changes can be merged by the review team. Routine changes include changes to the render context feature for TOCs and breadcrumbs, updates to product feedback links, and adding links to samples repositories.	Helps ensure major configuration changes are reviewed by a repository owner/admin.
All	Repo integrity	The PR does not include an embedded repo or any unusual, extraneous files. All file updates should be restricted to the articles and includes folders in the repo. Items to watch for: `.DS_Store`, `desktop.ini`, `.gitignore`, the entire repo embedded in the root folder.	Prevents repo from becoming random file storage, prevents cloning mistakes by users from being replicated to other users.
Docs	Repo integrity	The PR contains fewer than 100 changed files, unless the PR is intentionally updating a release branch from master. See the section in this article about large pull requests.	Ensures iterative workflow and reviewability of changes.
Learn	Repo integrity	For new modules, no more than one module per PR. Updates to existing content can go across modules in a single PR.	Ensures iterative workflow and reviewability of changes.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Repo integrity	If articles are deleted in the PR, the deletions must be by the listed author. Where PRMerger is running, authors who want to delete an article must list themselves as the author in a commit that precedes the commit to delete the articles. This change allows you to avoid a validation warning.	Ensures that deletions are intentional, in the past people have signed off on deleting files that were not part of their intended file changes.
All	Repo integrity	When an author deletes an article, the master redirect file must contain a redirect for the deleted content. If the repo uses a master redirect file, redirects use only the master redirect method. File-based redirects are not allowed in repos that use master file redirection.	Prevents usage of outdated redirect method.
All	Repo integrity	Only markdown (.MD), YAML (.YML), and image files are allowed in content repos.	Prevents repo from becoming random file storage. Ensures content is meant for publication as official docs.microsoft.com content.
All	Naming	New files and folders introduced into the repo follow the File name and path guidelines.	Supports consistency, repo management, SEO guidelines, and prevents Git problems related to casing.
Docs	Metadata	All includes must contain a metadata section. See details on include metadata requirements.	Ensures that includes have basic metadata so we know who owns them and what service they are used with.
Learn	Metadata	Markdown (.MD) and YAML (.YML) files must meet the detailed Learn metadata requirements.	Basic compliance for module titles, navigation, and to support SEO metrics.
All	Metadata	The `title` attribute should be sentence cased in YML and MD files.	Basic compliance with editorial guidelines for text that may be displayed in search results.
Learn	Metadata	YAML (.YML) files must not include `ROBOTS: NOINDEX`.	The platform automatically applies this attribute. Manual settings revert the global update.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Acrolinx scorecard	No valid spelling errors remain in the Acrolinx report. The author must fix the spelling errors.	Supports basic editorial quality for content credibility.
All	Acrolinx scorecard	No remaining do-not-use terminology. The author must correct the terminology.	Ensures key terminology is used per Microsoft guidelines published in the Cloud style guide.
Docs	Content	The article is a technical document covering technical subject matter and therefore is in the correct content channel. See the what goes where guidance.	Ensures that content is published to the correct content channel.
Learn	Content	Module content should be for the product area scoped for the repository. See the detailed mapping of content to repo. learn-pr: Cloud & AI Content learn-bizapps-pr: Power Platform learn-dynamics-pr: Dynamics 365 learn-m365-pr: M365 LearnShared: Learn landing, support pages If content appears to be in the wrong repo, contact learn-repo-managers.	Ensures content is localized in the correct language sets and is easy to find for maintenance and support.
All	Content	All information in an article is meant for the general public. docs.microsoft.com is exclusively for technical documentation that is available to the general public. Do not publish private preview content, content subject to NDA, or content that is otherwise confidential to the site.	Helps prevent disclosure of embargoed content or content protected by NDA.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Docs	Content	TOC files: when a new article is added, a new TOC entry is added; when a file is deleted, the TOC entry is either removed, or it is modified to point to the replacement article. In either case, the updated TOC file is in the same PR as the new/deleted file. In the content model, samples articles are not listed in the TOC, they are listed in a curated page.	Prevents the publishing of hidden content; all content is meant to be discoverable on docs.microsoft.com.
Docs	Content	Content related to Cognitive Services must be reviewed and approved by a member of the Cognitive Services content team before it can be reviewed by the PR review team.	Ensures alignment of content written by infrequent contributors.
Docs	Content	If a hub page is modified, approval by the designated business approver is required. Minor fixes to hub pages such as spelling fixes and link correction or replacement do not need approval. Addition of any new content or removal of existing content requires approval. To identify a hub page, the **ms.topic** metadata is set to **hub-page**. Landing pages no longer require business approval.	Ensures design alignment for hub pages.
All	Content	Bylines are not permitted. If an article calls out the name of the author or any contributor in the text, that attribution needs to be removed. Articles published from the tech content repo are considered to be authored by "Microsoft." Contributors who have committed updates to the article are recognized automatically on the contributor bar of the published article.	Supports automatic author/committer recognition functionality on the site. In the past, more obvious author recognition led customers to view our docs as blogs and they asked where our docs went. This led to the current minimizing of authorship on the page.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Learn	Content	In markdown (MD) files, only H2 and lower headings are allowed; do not allow authors to add H1 headings.	H1 headings are automatically applied by the system, using the `title` value defined in the associated YAML.
Docs	Content	The article contains an introductory paragraph and a procedural or conceptual body of content. The article needs to contain sufficient, complete content to stand on its own as an article. It should not be a small fragment of information. (An exception is a "Limits" article if it's in the context of a large article that lists all of the limits of a service.)	Ensures content is minimally complete prior to publish. Very short articles are known to generate low CSAT. Very short articles often indicate another solution is needed for the content problem (such as a redirect).
All	Content	Elements that should be numbered lists are numbered. Elements that should be unordered lists are bulleted.	Supports basic editorial quality so procedural steps are set up correctly.
All	Content	Unusual or novel graphics, information architecture or structures, or nonstandard designs need to be vetted with the PR review program manager. Teams that are experimenting with new things need to have a plan in place for evaluating experiments. Contacts: Docs - justinc; Learn: barlan + asjohnso.	Helps ensure alignment with general content standards by providing PR reviewers a path for escalating content that is far outside the norm for further review by content leads.
Learn	Content	Each new module must contain at least one knowledge check or task validation.	Provides consistency for learners.
All	Legal/compliance	Articles cannot use the terms "General Data Protection Regulation" or "GDPR" outside the context of the CELA-approved includes listed in the GDPR guidance posted in the contributor guide.	Ensures content aligns with CELA-required guidelines for content about GDPR.
Docs	Site/design functionality	Switchers are used only for switching across multiple versions of the same article.	Ensures alignment with intended purpose of docs UI element.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Site/design functionality	CodePen functionality is a new feature not authorized for broad usage at this time in C + AI technical content. Only approved pilot content should use the CodePen iframes (Location-Based Services is the approved pilot). Any pull request that contains CodePen iframes must be approved by Martin Ekuan.	Prevents overuse of a feature that is intended for limited usage at this time.
Docs	Site/design functionality	In article sets that use switchers, the H1 in each article contains information that differentiates each article from the other articles in the set.	Ensures that published article titles are unique for SEO purposes and so customers can differentiate between flavors of the article.
Docs	Site/design functionality	Each new include file is used in at least one Markdown article. New include files can't be added to the repo before they are used in an article. An exception exists in repos where a new include file that has "Applies to" content (used to identify the platform support that appears at the top of most articles in the docset) can be added without an update to an article.	Prevents orphaned includes.
Docs	Site/design functionality	A manually authored on-page TOC is not permitted in an article. The article must rely on H2 headings for its on-page TOC.	Ensures the automated TOC based on H2 headings is the only on-page navigation; prevents duplicate stacked TOCs in mobile and narrow page views.
Docs	Site/design functionality	If H2 headings are present, the article contains at least two H2 headings. Using one H2 heading creates a single-item article TOC. H2 headings must be used before H3 headings to ensure that a TOC is created.	Basic editorial guidance, ensures on-page navigation offers at least two clickable options.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Markdown	Source content contains minimal HTML at the block level. Minor inline HTML is permitted - such as superscript, subscript, special characters, and other minor things that you can't do with Markdown. HTML elements that are permitted by the docs platform (HTML allowlist are not allowed in most situations. HTML tables are allowed *only* if the table contains bulleted or numbered lists. Often, an HTML table is an indication that the content needs to be simplified so the source can be coded in Markdown.	Ensures alignment with basic content authoring guidelines for docs.microsoft.com. Markdown is our authoring standard.
All	Markdown	Custom Markdown elements are used where appropriate. For example, notes are coded through the [!NOTE] extension, not as plain text.	Ensures consistent customer experience and rendering where customer markdown extensions provide design functionality on docs.microsoft.com
Docs	SEO	In all repositories, ensure that a branded product name is present in either the **title** or **titleSuffix** attribute. Note that the **titleSuffix** value may also be set globally in the docfx file for the repository. In Azure content, "Azure" must be present in one of the attributes (Intune, StorSimple, and Microsoft Genomics articles in the Azure repo are exceptions). Because branding can be defined for a folder or a repo, PR reviewers verify branding in staged content. Exception: the cpp-docs-pr repo does not use product branding for the C++ content.	Supports key SEO guidance.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Docs	SEO	The H1 title contains sufficient information to describe the content of the article, to differentiate it from other articles in our content set and to map to likely customer keywords. For example, "Overview" as the H1 title is generic and provides no useful information to a customer or to search.	Supports SEO and discoverability of our content in search.
All	Terminology	The use of the "ARM" acronym, or the use of V1 or V2 as a reference to the classic and Resource Manager deployment models in Azure, is a blocking terminology item. Exception: these terms are allowed in nonvisible text, such as the **displayName** field in TOC files, to support SEO.	Supports key Azure terminology guidance.
Learn	Terminology	Do not use the word **quiz** anywhere in the content. The only acceptable use is the YAML **quiz:** property.	Learn has standardized on using the term `knowledge check` for public facing content since it is friendlier and does imply compliance.
All	Images	Images have clear resolution, are free of misspelled words, and contain no private information. All images must render and be legible in both light and dark views on the docs site.	Supports basic quality and usability of content.
All	Images	SVG is the preferred format for conceptual art – it scales regardless of the size of the browser window. However, SVG files that are scripted (where `<svg>` elements are present with enclosed `<script>` tags) are not permitted. PNG is also fine for conceptual art and the preferred format for screenshots (to avoid the need to convert the screenshot to vector format).	Code around SVG files is stripped from the build automatically, this prevents wasted effort troubleshooting content that won't render.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Images	All media links must be to media within the repository; no hyperlinks to media hosted on my-sharepoint.com, personal URLs, other websites, or other repositories.	Ensures all media is managed by Microsoft, ensures media links are validated, future proofs for migration.
All	Staging	The content preview must be clean on staging in both light and dark view. It cannot contain any obvious formatting issues: - A numbered or bulleted list that appears as a paragraph - Code in a code block appearing partly in the code block and partly outside it - List steps numbered incorrectly due to faulty indentation - Leftover merge-conflict markers - Content must preview legibly in both light and dark views. Pay particular attention to conceptual artwork rendering correctly in dark view	Supports basic usability and credibility of published content.
Learn	Videos	Channel 9 is the preferred internal platform; RedTiger is supported but no longer required. External video links must go to Microsoft channels (that is, a Microsoft-sponsored channel on YouTube).	YouTube is not available to all learners globally, and using an internal platform allows us to handle providing different content to global learners (localization, content, etc.)

Non-blocking content quality items

For these items, PR reviewers provide feedback and instructions for the author to follow up with fixes in a later PR. This feedback does not block the decision to merge. Authors should follow up within three business days with fixes.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Docs	Content	Articles should have a "Next steps" section at the end with one to three relevant and compelling next steps. Brief text should be included that helps the customer understand why the next steps are relevant.	Best practice to drive engagement with additional content.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
All	Content	PR reviewers might provide feedback on a few minor spelling, grammar, and other writing issues as non-blocking feedback. If there are more than a few editorial issues, reviewers log an edit request for the article for a post-publication edit.	Supports basic editorial quality, content credibility.
All	Images	Images use the correct callout style and color, and screenshots use the correct border and placeholder style. See the guidance for screenshots.	Basic consistency.
All	Images	Images include alt text. See the guidance for screenshots and the guidance for conceptual art.	Basic accessibility.
Docs	Site/design functionality	The H2 headings, when rendered in the on-page TOC, should ideally wrap to no more than two lines. Longer headings make the article TOC harder to scan.	Basic design and usability, MVC content model support.
All	Style conventions	All titles and headings are sentence case, per MSFT style.	Basic consistency.
All	Process	If the pull request could easily be configured to benefit from PRMerger automation, pull request reviewers provide feedback to the author about how to use branches so the changes can be merged automatically. See the PR best practices article.	Educating authors to make use of the system.
All	Content	All external hyperlinks resolve to content that exists and that appears to be high-quality and credible.	Basic quality.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	RATIONALE
Learn	Content	Each new module must contain at least 3 units (YML files) and 3 corresponding MD files. One unit must be an introduction, one unit must be a summary, and the introduction unit must have learning objectives and prerequisites. (The inclusion of at least one knowledge check is a blocking item.)	Provides consistency for learners.

PR review and build status policies

The OPS build provides for three potential build results: error, warning, and suggestion.

STATUS	EFFECT ON PR REVIEW PROCESS
Error	Not possible to merge; the system won't allow it. The build error must be resolved.
Warning	The PR review policy is that no PR will be merged with outstanding warnings. If a warning occurs unexpectedly on a day with a major release, the repository admin can decide to merge a PR with a warning to ensure content for the release goes live. This should not be done during routine daily publishing. The warning must be fixed as soon as possible afterwards. PR reviewers will not merge PRs with warnings.
Suggestion	Suggestion status is used when new validation rules have been turned on in a repository to help educate authors. All suggestions, after an introductory period, will be turned into warnings. Users are encouraged to fix items called out in suggestions, but acting on them is optional. PR reviewers will not block PRs that have suggestion results.

PRs with more than 100 changed files

Large PRs have a demonstrated history of containing unintended changes that break, damage, or delete content.

Therefore, the best practice is to submit changes in MUCH smaller batches iteratively so that the changes can be easily and quickly reviewed. New articles should be submitted in batches of five articles or fewer - for more information, see Content release planning + process.

PRs that include 100 or more files have to be handled specially to protect the integrity of the repository. These guidelines apply to pull requests against the master branch and to all release branches.

- **Bulk file moves:** PRs that move many files will be reviewed per the process listed in the Moving or refactoring files in a repository article. The PR review team can accommodate broken link warnings when multiple PRs are used, as recommended; in these cases, links are verified only in the final PR.
- **"Dirty PRs":** When a release branch has a merge conflict with master, the PR review team verifies the "dirty PR" that is required to resolve the merge conflict and bring the release branch up-to-date with

master. PRs of this type will be reviewed per the process listed in the Resolve merge conflicts in Git and GitHub article.

- **Bulk updates**: Because the GitHub UI now shows diffs beyond 100 files under certain conditions, the PR review team may be able to review bulk changes in a large PR. To qualify, the changes in the PR must be scoped to a limited and clearly identifiable set of changes across all the files.
- Any PR that contains numerous changed or new files outside these conditions (bulk file moves, dirty PRs, and bulk updates) does not qualify as reviewable and must be broken up into easily reviewable chunks. This enables multiple reviewers to work on the reviews, it allows feedback to be provided in manageable chunks, and it facilitates an iterative workflow. If you have to break up a large PR, see Break up a single large pull request into smaller PRs.
- PRs that present special problems beyond the scope described here will be referred to the repository administrator, delaying review of the content changes. The administrator will review the scope of the PR and make a recommendation about how to proceed in a way that ensures the quality of the content and the integrity of the data in the repository. The repo admin is responsible for merging PRs like this.
- The file limit guidelines do not apply to PRs in Docs repos where an upstream release branch is being merged to an upstream master branch for publishing. The content of the release branch should have been reviewed as each change was merged to the release branch along the way, so a final PR review is not required.

Templates and stubs in release branches

Occasionally, writing teams may want to add templates for new articles in a release branch. This allows the writing team to stub out content so that file names, metadata, and TOC links are done correctly. The partners authoring the new articles can then focus on writing the needed content.

PRs that contain stubbed out content can be merged to a release branch if they meet these criteria:

- The PR contains only the stubbed out articles and TOC links to those articles.
- The PR is against a release branch.
- The stub articles use correct file naming.
- The stub articles contain complete metadata. The title, description, author, and other information should be complete.
- All content in the stub articles should be instructions about the content to write, not the actual content.
- The content team manager (business approver) must approve the PR to ensure follow-up so that the stubs are converted to full articles before the merge of the release branch.

An article is not considered a stub or template article if it contains partially written content. Articles that are partly written don't qualify for merge under this template/stub guidance.

The following PR is an example where the writer has correctly submitted stubbed out articles to a release branch: https://github.com/MicrosoftDocs/azure-docs-pr/pull/48223/files.

This guidance helps ensure that all new content goes through the PR review process and helps prevent incomplete content from being published.

PRs that require repository admin review, merge, or escalation

- Configuration file updates
- Any case where a user wants an exception to these PR criteria

Azure Architecture Center business approvals

All new articles in the Azure Architecture Center require business approval.

Process:

1. Contributor creates a new article, makes a pull request, and enters the #sign-off comment.
2. PR reviewer picks up the pull request and notices that it contains a brand new article.
3. PR reviewer checks the PNP GitHub team – if the new article is from one of the members of the GitHub team, they proceed with the review, and no business approval is needed. If the new article is from anyone else, the PR reviewer goes to the next step.
4. PR reviewer assigns the "pending-content-team/business-approval" label to the pull request and sends mail to the pnp@microsoft alias to request business approval to move forward with the review process.
5. Someone from the alias responds to the mail with approval to move forward and enters #sign-off again in the pull request.
6. Once someone from the alias signs off, the PR review team proceeds to review the new content per the standard process.

Automated PR criteria

The following pull request review criteria are now part of the automated validation checks, rather than human reviewed.

APPLIES	CATEGORY	QUALITY REVIEW ITEM	DEPRECATION REASON
Docs	Metadata	A metadata section is present at the top of the file. The metadata section starts with three hyphens, is followed by a list of metadata that includes at a minimum `title`, `description`, `author`, `ms.author`, `ms.date`, `ms.topic`, and `ms.service` *or* `ms.prod`, and ends with three hyphens.	Replaced by build validation in the Developer Relations/CGA Reporting ruleset. Removed from list 11/8/2019.
All	Localization	Links to pages on microsoft.com websites are coded as locale agnostic. Do not include `en-us`, `en-gb`, `en-in`, or any other locale in links to these sites. TechNet/MSDN forum links are an exception--locales cannot be removed from forum links on these sites.	Ensures correct localized content experience. Replaced by build validation in the Developer Relations/CGA Reporting ruleset. Removed from list 12/20/2019. Note that this automation is not yet enabled in Learn repos. However, when discovered in the human PR review process, URLs that include locales are flagged as blocking.
Docs	Content	The article contains only one H1 heading.	Multiple H1 headings break page rendering and are bad for SEO. Enabled 6/29/2020.

Overview of PRMerger

5/27/2021 • 8 minutes to read

PRMerger is workflow automation for our GitHub content repositories that supports review workflows in public and private repositories. In private repositories, it merges pull requests automatically. The merge occurs after the author signs off when the pull requests meet the criteria for automatic merge.

PRMerger problems/bugs

If PRMerger isn't working as expected, file a bug!

1. Go to https://aka.ms/sitehelp.
2. Select **Submit request or issue**.
3. Choose **Select Service** > **OPS**.
4. Choose **Select Service Category** > **PRmerger/repo automation**.

PRMerger private repo workflow

Here's how PRMerger works in *private* repos:

- It allows read-level contributors to assign the `ready-to-merge` label to a pull request. The `ready-to-merge` label lets the pull request reviewers for the repository know the PR is ready for review.
- It applies business rules (criteria) to allow certain pull requests to be merged automatically.
- Pull request reviewers focus their reviews on new content and significant updates.

After you configure a private GitHub repository to run PRMerger, the service monitors that repo for new pull requests. PRMerger offers a review workflow that relies on labels to indicate where in the process a pull request is. The following list describes the process:

1. Adds the `do-not-merge` label to all new pull requests in all branches except live.

2. Checks the `ms.service` / `ms.subservice` or `ms.product` / `ms.technology` metadata fields. If one or more is present with a value, it adds the value as a label. This helps reviewers route PRs appropriately. PRMerger appends a suffix of `/svc`, `sub/svc`, `/prod`, or `/tech` to the label value to assist with label filtering in some CGA reports.

3. Assigns the PR and requests a review from the authors of articles in the PR. There can be up to five people automatically included in that review. PRMerger notifies them of the change and gives them a chance to provide feedback, even if the PR gets automerged. The process skips this step if there are more than 50 articles in the PR. The process assumes that it's a bulk update and doesn't need author review.

   Any Microsoft employee or vendor with Read permissions can include multiple assignees by typing `#assign:<GitHubID>`. Or, reassign the PR and remove current assignees with `#reassign:<GitHubID>`.

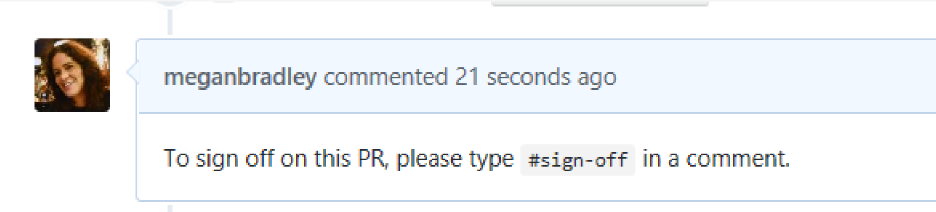
4. Next, PRMerger waits for a successful OPS build to complete for the pull request. If the build has errors or warnings, the OPS build writes the result to the pull request. You can't sign off until there's a clean build.

5. After a successful build, PRMerger applies its business rules. The business rules determine whether the pull request qualifies for automatic merge or whether it requires human review:

   - If all conditions are met, PRMerger adds the `qualifies-for-auto-merge` label.
   - If not, PRMerger adds the `needs-human-review` label.

6. At this point, you should review the staged content.

7. If the content looks good, type `#sign-off` as a comment in the pull request as plain text.

> **NOTE**
>
> If you type `#sign-off` within any formatting, such as inline code, bold, or an HTML tag, it won't be recognized as a sign-off. The sign off doesn't happen because PR reviewers often need to provide sign-off instructions to contributors in GitHub comments, and need a way to do so without actually signing off on the PR. For example, if you type a comment like the following example:
>
> ```
> To sign off on this PR, please type `#sign-off` in a comment.
> ```
>
> The comment displays like this example and won't be treated like a sign-off:
>
> 
>

- If the `qualifies-for-auto-merge` label is assigned to the pull request, PRMerger checks if the GitHub `merge-ready` flag is applied. If so, your pull request is immediately merged after you enter the `#sign-off` comment.

> **NOTE**
>
> In some cases there are delays on the GitHub side that prevent `merge-ready` from being applied. PRMerger will retry three times. If the PR is still not merge-ready, you'll see a message indicating a delay, and PRMerger will try again in two hours.

- After the pull request is merged, PRMerger assigns the `merged-by-prmerger` label to the pull request.

- If the `needs-human-review` label is assigned, your pull request will enter the queue for human review and feedback.

8. After the review of a `needs-human-review` pull request is complete, one of the following occurs:

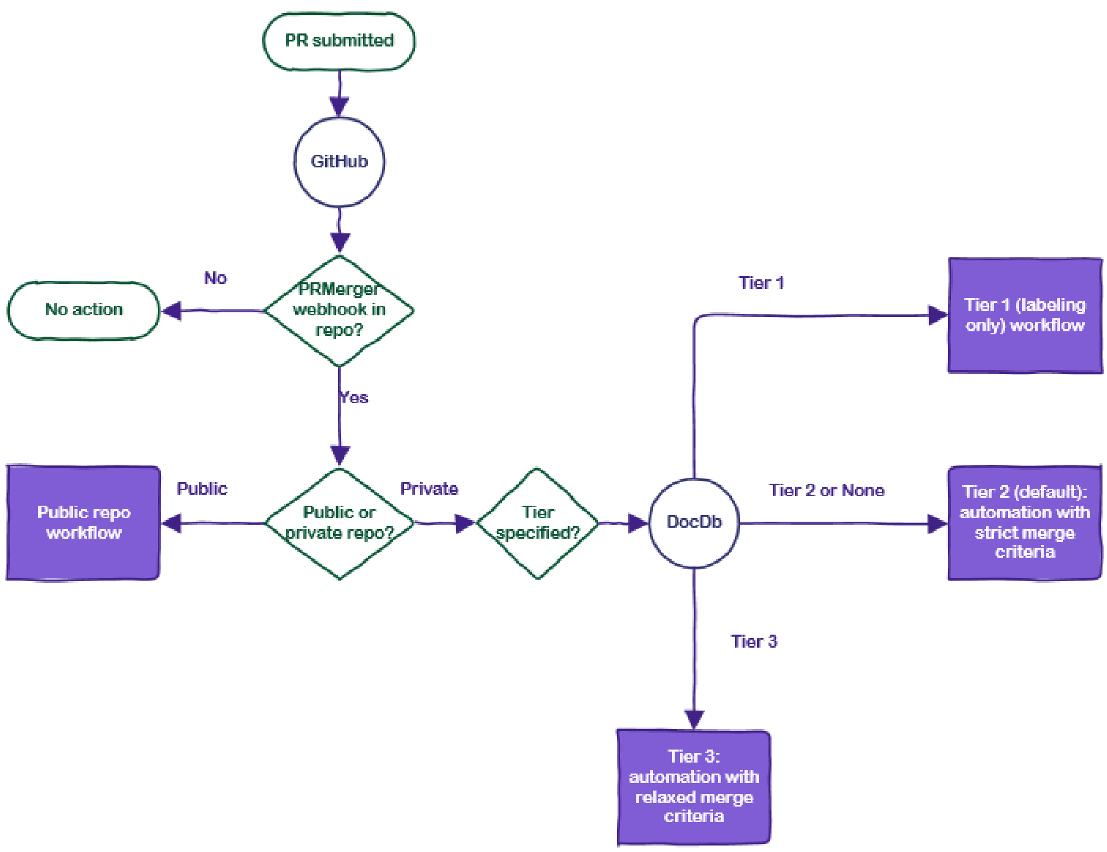
- If the pull request meets the pull request criteria, the pull request reviewer merges the pull request.
- If the pull request doesn't meet the criteria, the reviewer provides feedback in the pull request comments and types `#hold-off` to reassign the `do-not-merge` label. After you address the feedback, the article must go through the full workflow again.

A pull request can only be merged if **the following conditions** are met:

- The build is successful.
- All PRMerger validation filters pass.
- The author adds a `#sign-off` to the pull request, which assigns the `ready-to-merge` label.
- Pull requests that require human review must meet the pull request criteria.

Private workflow diagram

PRMerger high-level



PRMerger business rules

Pull requests need to meet the following business rules to qualify for automatic merge. If a pull request fails any of these rules, it's labeled for human review.

1. PRMerger first adds a label called `do-not-merge` to all new pull requests. Based on the business requirements, a pull request can only be merged if **the following conditions** are met:

   - Build is successful.
   - All PRMerger validation filters pass.
   - The author adds a `#sign-off` comment to the PR, which triggers the process to assign the `ready-to-merge` label to the PR.

2. The Pull Request triggers the OPS build. If the build is successful the PRMerger automation kicks in, otherwise the author is notified of the failed build.

3. After the successful build, the PRMerger business filters run.

   a. If all of the *filters are successful*, PRMerger adds a label `qualifies-for-auto-merge`.

   The author reviews and adds a `#sign-off` comment. Once PRMerger sees the sign-off comment, it adds the `ready-to-merge` label, confirms the filters were successful, and the build has passed. PRMerger will then continue to auto-merge this Pull Request and add a label `merged-by-prmerger`.

   b. If the *filters aren't successful*, PRMerger adds a label `needs-human-review`.

   When this label is assigned, the selected pull request reviewers for the repository have to review the pull request. If there are no content issues, they merge the Pull Request manually. If there are, they add a comment `#hold-off` and notify the author to fix. This reassigns the `do-not-merge` label so that PRMerger won't continue with merging the PR.

PRMerger filters

For a pull request to qualify for automatic merge, the pull request must meet these criteria:

- Affects no more than 10 files.
- Contains no more than 15 commits.
- Contains no more than 20% change.
- Contains no new files or deleted files.
- Doesn't change, add, or remove any images.
- Contains no index pages, hub pages, home pages, or landing pages.
- Contains no config files, breadcrumb files, or TOC files.
- PR title doesn't contain the words "edit pass".
- There are no unsupported file extensions being checked in.
- No articles were deleted by someone who isn't listed as the author of the article.
- No files submitted into the root folder of the repo.
- No file in the PR has an Acrolinx score of less than 80 (if Acrolinx is enabled on the repo).
- No references to the General Data Protection Regulation (GDPR).
- No file in the PR contains a CodePen reference.
- No changes to the casing of file or folder names.

PRMerger public workflow

This workflow is applied to all *public* repos when PRMerger is enabled.

1. When an author submits a PR, check the `ms.service` / `ms.subservice` or `ms.product` / `ms.technology` metadata fields. If one or more is present with a value, add the value as a label. The label helps reviewers route PRs appropriately. A suffix of `/svc`, `sub/svc`, `/prod`, or `/tech` is appended to the value in the label to assist with label filtering in some CGA reports.

2. Check the submitter's GitHub ID to determine if the submitter is the `author` of one or more files in the PR. If yes, close the PR with a message that the author should use the private repo.

3. If the submitter isn't the author of one of the articles in the PR, check whether `ALL CLA Requirements Met` (determined by CLA logic - pass if the submitter is a Microsoft employee OR has signed a CLA). If no, no action. If yes, continue with the next step.

4. Assigns the PR and requests a review from the authors of articles in the PR. There can be up to five people automatically included in that review. The process applies the `Change sent to author` label. The process also skips the PR assignment if there are more than 50 articles in the PR. It assumes that the PR is a bulk update and doesn't need author review.

   > **NOTE**
   >
   > Any Microsoft employee or vendor with Read permissions can add more assignees by typing `#assign:<GitHubID>`, or reassign the PR and remove current assignees with `#reassign:<GitHubID>`.

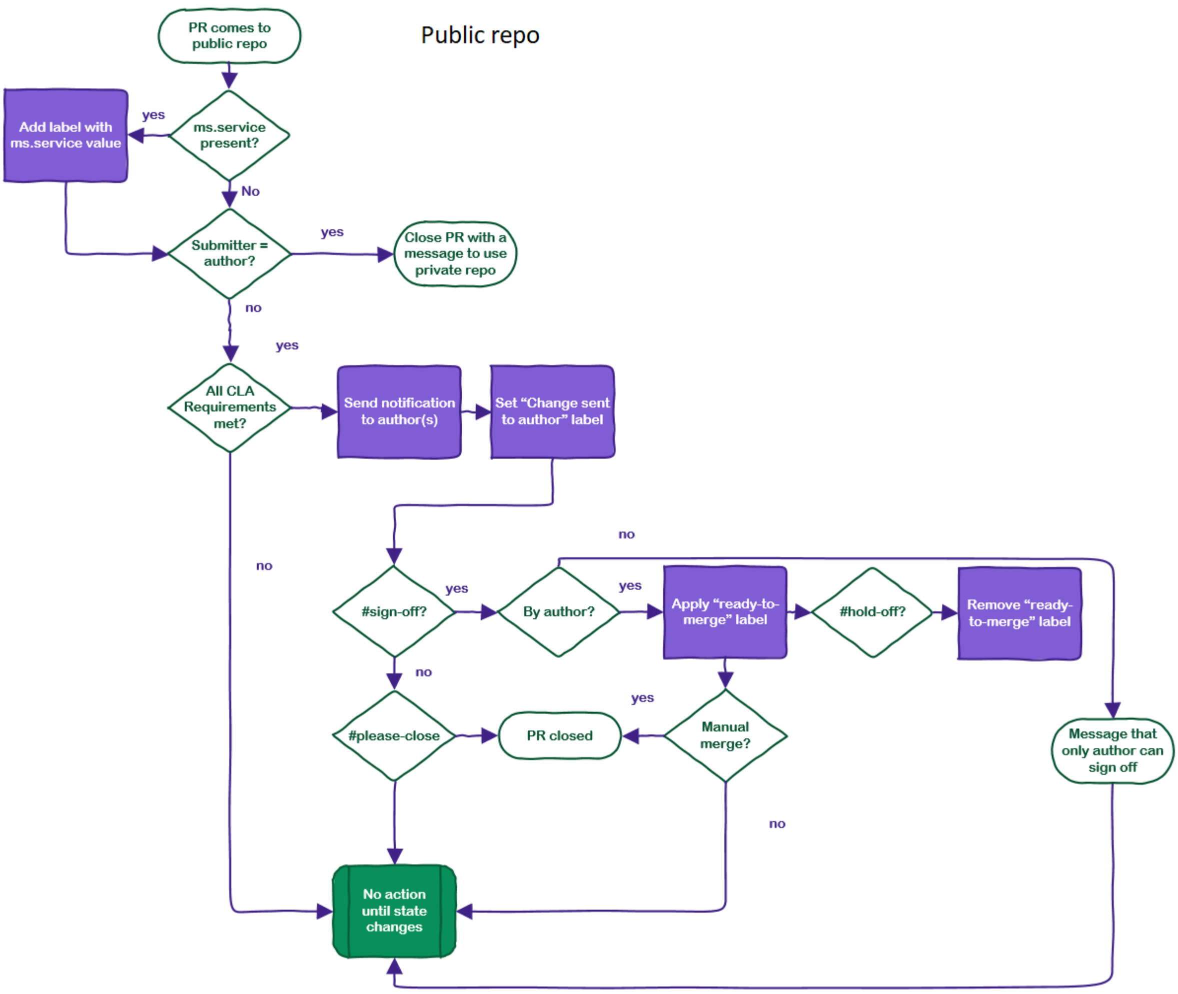
5. An author of the PR adds the `#sign-off` comment, confirms they're the author of at least one file in the PR, and adds `ready-to-merge` label. If someone other than an author adds `#sign-off`, a comment is added: `@<submitter>: Only the author, @<author>, can sign off on this pull request`.

6. The PR isn't automatically merged. First, review the PR and then merge it manually with a request from the relevant product or content team member.

7. Cancel the `#sign-off` comment by adding `#hold-off` in a GitHub comment for the PR.

8. Any user can type `#please-close` to close the PR if the changes can't be merged.

Public workflow diagram



How do I turn on PRMerger?

Discuss your workflow with your team before turning on PRMerger. Then, decide how the PRMerger automation will be integrated. Some teams use the Content & Learning PR Review vendor team, while some teams triage and review their own PRs.

For more information, see How to turn on GitHub automation.

Release notes

June 6, 2019

- In both public and private repositories, PRMerger automatically assigns the pull request to the listed author. You'll always receive a GitHub notification when someone creates a pull request against an article that you own - in both public and private repos. This change addresses longstanding feedback from authoring teams.
- The `needs-human-review` or `qualifies-for-auto-merge` labels can be applied in any branch except for `live`. Previously, only `main` and `release` branches were covered. Now, PRMerger has full functionality in `Learn` repos where there's a new convention for release branch naming.
- PRMerger adds product, service, subproduct, and subservice labels to pull requests in the private repository workflow. The functionality is now present in both public and private workflows. The labels let you sort and search PRs by service.
- PRMerger relied exclusively on in-article metadata for assignment and notifications. PRMerger now supports

metadata assigned through the `docfx.json` file.

Pull request review process when PRMerger isn't available

5/20/2021 • 4 minutes to read

Many repositories don't have PRMerger webhooks enabled. This article describes the PR review process for those repositories. The PR review process was designed with the following goals in mind:

- Publish high-quality content from our team, product team members, and community members.
- Provide timely, actionable feedback to authors in a consistent manner.
- Enable discussion between authors and reviewers.

The processes continue to evolve as teams innovate and as the platform matures.

Reviewers

One of the content team members reviews every PR. Content team members may request a review from the specific product-team members to verify technical accuracy. The content team uses GitHub's Code Ownership feature to automatically request reviews from content team members. As part of this process, a reviewer may tag other team members to review internal PRs. Team members see requested reviews from team members and community members in the same queue.

Community members can review PRs and provide feedback as well. However, at least one member of the core content team must approve any PR before it's merged.

Review process

Reviews follow one of three paths based on the scope of the PR:

- **Small PRs** - Small PRs are a single bug fix: typos, broken links, small code changes, or similar changes.
- **Major contributions** - Major contributions are significant edits to an existing article, new articles, or edits to multiple articles.
- **Draft** - Authors can request an in-progress review by opening a draft PR in GitHub.

The processing used by the Contributor License Agreement (CLA) bot is a good guideline for the distinction between "small" and "large" contributions. PRs that don't require the CLA to be signed are likely "small." PRs that do require the CLA are likely "large."

Small PRs

The changes in small PRs are reviewed for accuracy. They're checked using the build on the review site. Because they're small, these PRs don't trigger a review of the entire article.

Reviewers may notice other small changes that would improve an article. If the changes are also small, the reviewer should suggest the changes as review comments. However, if the suggestions are large, the reviewer should open an issue and address it later.

Larger changes

Larger PRs undergo a more thorough review. The following are all checked:

- Sample code must be included in the PR, in source and as a downloadable zip file.
- Sample code compiles and runs correctly.
- The article clearly describes the goals for the reader, and the goals are met.

- The article meets style and grammar guidelines (see CONTRIBUTING.MD in the root folder of the repo.)
- All links should resolve correctly.

Content team members review the PR and submit the review with comments. If the reviewer requests only minor changes, team members may "approve" the PR with the feedback. The author can then address the feedback and merge the PR. Most reviewers request changes and when the author makes the changes, the reviewer reviews the updates again.

If the edits require a technical review, the content team reviewer requests a review from the appropriate product team member.

Review draft pull requests

New authors may want feedback earlier in the process than after the updates are finished. If they want an early review, the author can open a draft PR in GitHub. Then, the author can request an early review in a comment.

These early reviews aren't as thorough as a full PR review. The content team will comment, but won't "approve" or "request changes" using GitHub's review feature. These early reviews focus on the structure of the article: the outline, the overall content, and the samples. These reviews don't include a thorough check for grammar and correct links.

Explain suggestions

GitHub lets you enter comments in triple-back-tick blocks of type `suggestion`. The suggestions are displayed as a diff and can be merged by clicking a button. On short lines, GitHub does a good job of highlighting the changes. On longer lines, such as a long paragraph in one line of text, GitHub doesn't highlight the changes. When entering a suggestion for a long line, notice whether your changes are clearly highlighted. If the changes aren't highlighted, include comments outside the suggestion block explaining what you changed. Without an explanation, it's often time-consuming for other reviewers or the PR author to figure out what the changes are.

Respond to reviews

The author updates the PR to respond to comments and marks each addressed comment with the "+1" reaction. The author marks the conversation as "resolved" when they've addressed the review comments. If the author disagrees with the comment or addresses the comment in a different PR, the author adds a comment to explain the change. In those cases, the reviewer will mark the conversation as "resolved" when the comment has been addressed.

The author @-mentions the original reviewer in a comment to request a new review.

Response time expectations

Content team members typically review new PRs in under two business days. If it takes longer to review, one of the team members adds a comment to the PR and sets expectations.

Once a review has been submitted, authors should try to respond to comments in a week or less. Volunteers may not meet these expectations because of other commitments. In those cases, team members ask if the community author will update the PR. If not, the team member updates and merges the PR for them.

Contribute Azure CLI and Azure PowerShell sample scripts

4/16/2021 • 6 minutes to read

> **IMPORTANT**
> This article only applies to the `Azure-Samples/azure-cli-samples` and `Azure/azure-docs-powershell-samples` repositories.

Scripts for Azure command line tools (Azure CLI and Azure PowerShell) are a separate product from Azure documentation, and have their own repositories. Even when you're writing a script to be shown in an `ms.topic: sample` article, users may come to scripts through organic search or the Azure Samples portal and never see the documentation site. Because of this, scripts are reviewed separately from documentation PRs and have different requirements.

This document outlines the general steps to get your scripts ready for submission and what the pull request process is like for samples, including a breakdown of the review process.

Follow style guides

The first step for submitting to Azure tool repos is to make sure that your scripts meet the style conventions:

- Azure CLI style conventions
- Azure PowerShell style conventions
- Best practices for Azure CLI and Azure PowerShell scripts

Following the style guides is a good way to make sure that your pull request is processed as fast as possible. Keep in mind that no style guide is exhaustive. Reviewers are allowed to block a PR as long as there's a clear reason for requested changes. Part of the review process is encouraging a dialogue between the reviewer and the submitter to evolve these style guides and the sample scripts, so that they reflect the best of what we have to offer users.

Test script changes

Testing is the most important part of any code submission process. There's no automated testing available for the Azure tool samples at this time, so testing is up to you and the pull request reviewer. Azure CLI and Azure PowerShell have different testing requirements.

Make sure that all scripts in your PR are tested as described.

> **IMPORTANT**
> Make sure you **don't** test in an elevated session ('admin' on Windows, or through `sudo` or as `root` on Linux/macOS.) Testing a script as an elevated user can hide problems like writing to directories not accessible to a normal user.

When you can skip testing

You can skip testing if your PR has **only non-code changes**. Non-code changes include comments, whitespace, and files that aren't executable scripts such as `README` documentation. **All other changes require testing**, no matter how small they are.

Testing Azure CLI samples

When testing Azure CLI samples, **make sure** you have the latest version of the Azure CLI. It's recommended to test scripts on Azure Cloud Shell since it's an environment guaranteed to have the latest version.

Test on **at least one of** the following environments:

- Linux (any distribution)
- Azure Cloud Shell `bash`
- macOS (10.13 or later)
- Windows Subsystem for Linux (v1 or v2)

Testing is done by running the script directly. **Don't copy-paste the script.** Copy-paste can cover up some subtle formatting errors that affect execution. Start a shell session and test a script with:

```
bash ./script-name-here.sh
```

Your script should complete without unexpected errors when:

- Author-supplied values for variables aren't modified
- Correct user-supplied values are used where required

Testing Azure PowerShell samples

When testing Azure PowerShell samples, **make sure** you have the latest version of Azure PowerShell. It's recommended that you test on Azure Cloud Shell for your PowerShell 6.x platform.

Test on **all of** the following environments:

- PowerShell 5.1 (Windows-only)
- PowerShell 6.x (any supporting OS)
- PowerShell 7.x (any supporting OS)

Make sure that you record which version of PowerShell 6 you test with, and on which platform. Azure Cloud Shell is considered an OS/platform, for this purpose. This information is required for the pull request submission.

Testing is done by running the script directly. **Don't copy-paste the script.** Copy-paste can cover up some subtle formatting errors that affect execution. Start a PowerShell session and test a script with:

```
.\script-name-here.ps1
```

Your script should complete without unexpected errors when:

- Author-supplied values for variables aren't modified
- Correct user-supplied values are used where required

Create your pull request

Create your pull request against the appropriate repository. There's a pull request template that has the following requirements:

- A short description of your changes
- A checklist signing off on testing, security, and important script conventions
- A list of the tool versions and platforms you tested on

> **IMPORTANT**
> If your PR is blocking acceptance of changes to another repository, *make sure* you link to that pull request in your description to expedite PR processing.

The templates are hosted on GitHub:

- Azure CLI checklist
- Azure PowerShell checklist

All of these sections need to be filled out before pull request processing begins.

Designated reviewers

Azure PowerShell samples

- Submit PR to Azure/azure-docs-powershell-samples
- Reviewer is Mike Robbins @mikefrobbins

Azure CLI samples

- Submit PR to Azure-Samples/azure-cli-samples
- Reviewer is Delora Bradish @dbradish-microsoft

Pull request review and processing

Because of a lack of resources, there's no SLA for when PRs will be reviewed. To speed up pull request processing, make sure that you test *before* submission. Also complete the checklist and make sure it's accurate when creating your pull request. Most pull requests have review done within the first day of checklist completion.

Pull requests are tagged with GitHub labels to help make it easier to understand where in the process the pull request currently is.

Reviewer responsibilities

Reviewers should handle PR processing in a timely fashion. When reviewers leave comments on a pull request, they must:

- Be clear about what needs to be changed, with a rationale.
- Make it clear if a review comment references a **required** change, a **suggested** change, or an **upcoming requirement**.
- Make it clear if a review comment asking for clarification of a command/comment requires an answer before merge.

Reviewers **don't** review scripts for technical accuracy ("does what it claims.") That is the responsibility of the script author and their team - reviewers can't be experts on all Azure services. If a reviewer sees an obvious technical error they're still expected to call it out.

Reviewers are allowed to block PRs for anything they consider to be critical to change or clarify. If there's a dispute between the submitter and the reviewer, the sample repo owner is the final arbiter of whether a change is required or not.

Review steps

Review follows these steps:

1. **Submitter**: Fill out the `Description`, `Checklist`, and `Testing information` fields when you create your pull request. If your PR *only* consists of whitespace, comments, or non-script changes, you may replace the `Checklist` and `Testing` sections with `N/A`.
2. **Repo owner**: Assigns the PR to a reviewer if the checklist is complete. If the checklist isn't complete, the submitter is told what is missing and the PR is labeled as `incomplete-checklist`.
   - **Submitter**: When the checklist is completed, @ the repo owner in a comment with `#sign-off`, and goes back to step 2.
3. **Reviewer**: The reviewer starts review when they have availability, and labels the PR with `in-review`. The reviewer:
   a. Checks for compliance with the style guides, leaving comments for anything that needs to be changed. If there are no required changes, or the requested changes don't block testing, moves on to:
   b. Performs review. Reviewers check for style guide compliance, obvious errors, and use their best judgment to request changes. If there are no required changes, or the requested changes don't block testing, moves on to:
   c. Tests PR scripts. All scripts are tested as described in Test script changes. All platforms the submitter recorded in `Testing` are retested. For PowerShell, reviewers also test an OS that was **not** checked by the submitter.
4. **Reviewer**: If the PR requires changes, the `in-review` label is removed and the `requires-rework` label is added. The submitter is @ed in a comment.
5. **Submitter**: Address any issues, and then @ the reviewer in a comment with `#sign-off`. Review goes back to step 3.
6. **Reviewer**: Once there are no issues, all labels are removed and the PR is merged.

Pull request submission best practices

4/9/2021 • 3 minutes to read

To publish changes to content, you submit a pull request from your fork. Pull request submission and processing are covered in both the minor/infrequent changes and major/long-running changes workflows.

Pull requests that add new articles, new images, or that significantly update existing content are human-reviewed before being merged. Read this article to learn how to work with pull request reviewers. Also, learn how you can create pull requests that are easier and faster to review. The pull request queue works better for everyone when you create easier pull requests.

Work with pull request reviewers

Pull requests commonly go through a technical quality review or a content quality review, or both, to ensure complete quality.

Technical quality review

Some teams require a technical review of pull request content before it can move to the content quality review and merge stage. If your team implements a technical review, you'll likely receive an invitation and link to the PR to provide your feedback. The feedback mechanisms used to review and update the PR vary, depending on the tool that's used. Work with your technical review team for specifics.

Content quality review

Here are the basics you need to know about working with PR reviewers during the content quality review process:

- **Understand the role of the pull request reviewers.** The PR reviewers:
  - Ensure the basic quality of the content.
  - Prevent regressions in the repository.
  - Provide feedback before merging.

  Content quality reviewers are in a content governance role. The primary intent isn't to merge submitted content as quickly as possible. Expect feedback that will require you to make updates, especially for new and heavily revised articles. The content quality reviewers are also responsible for the final step of merging the pull request, after it passes all content quality criteria.

- **Plan ahead with the pull request reviewer team:**
  - For high-priority pull requests.
  - For pull requests for timed/dated releases.
  - For pull requests that change or add lots of files.
- **It may take a few days to have a pull request merged**

  The publishing process happens at least two times a day. It can still take some time to address blocking feedback from the PR reviewer team before your pull request is ready to merge.

Make the pull request queue work better for everyone

There are two basic realities in the PR queue:

- Pull requests that are small in scope and that contain similar changes take less time to review.

- Pull requests that are large in scope or that contain mixed kinds of changes take more time to review.

You can help make the pull request queue work better by following these best practices:

- Separate minor updates to existing articles from new articles or major rewrites. Work on these changes in separate working branches.
- When you delete articles or images, avoid mixing the deletions with content additions or updates. Handle the deletions in a separate PR that gets merged after the PR for related additions and changes. For example, update the redirects file and check that the redirect is working before deleting an article.
- If you're a Microsoft employee:
  - Contact the PR review team to speed up pull requests only when necessary. You can request sped up PR handling for Red Zone, privacy, legal, and security issues; for truly broken customer experiences; and for executive escalations.
  - For releases or refactoring of content, plan ahead with the publishing team to create a release branch. Coordinate merge-to-master times with the team so your content is published at the right time. See Use release branches for details.
  - If you're trying to coordinate out-of-band dates made by others with changes you're making, which are interdependent, you must coordinate that work ahead of time with the PR review team. Otherwise, you risk having many broken links.
  - For any significant and/or timed release, you can request a "pre-review" from the PR review team to make sure you have time to address any blocking issues.

In a hurry? Submit PRs that can be accepted automatically

If your repository supports it, use the PRMerger automation rules to get more of your day-to-day PRs merged automatically.

PRMerger can accept your PR automatically, if:

- It affects 10 files or fewer.
- It has 15 commits or fewer.
- Less than 20 percent of text changes.
- Selectors/switchers aren't updated.
- No files are deleted or added.
- No images are new, changed, or deleted.

Need to make many little changes?

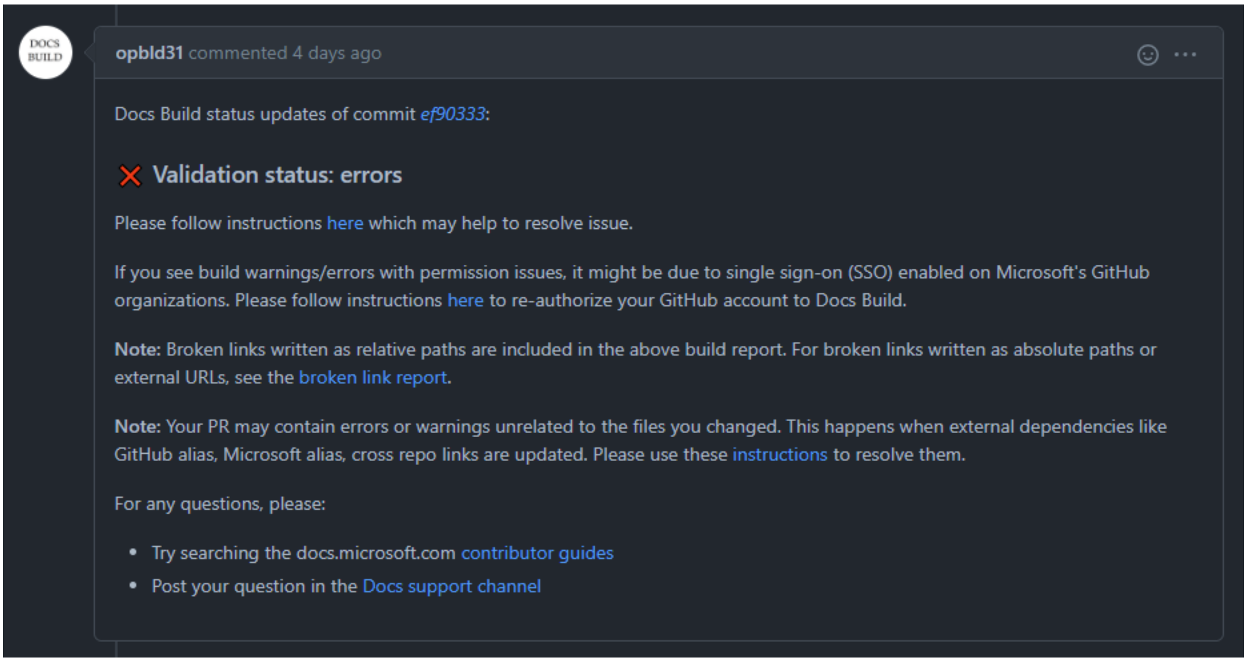
Take your cue from the preceding PRMerger automation rules, and follow these steps:

- Submit articles with light changes together in a PR with 10 or fewer files.
- Create a separate PR for articles in which images or selectors change. These changes require human review.
- Create a separate PR for new or deleted articles. These changes require human review.

Resolve a stuck pull request

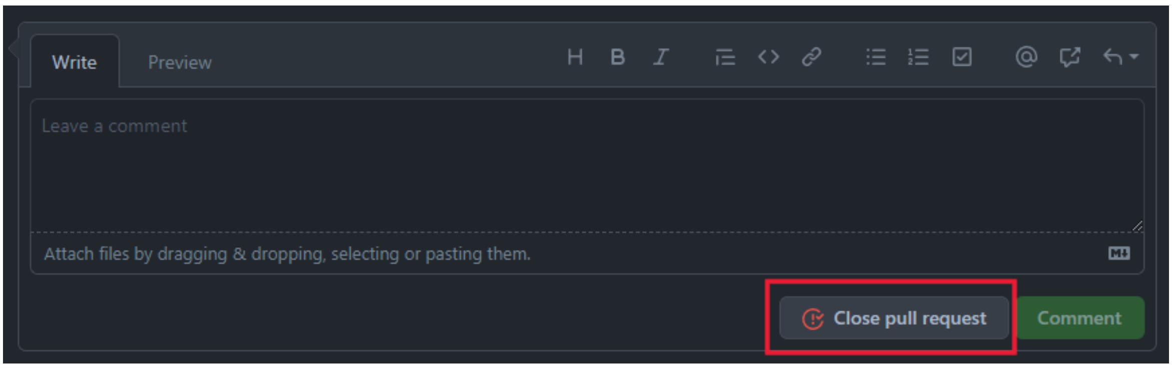
6/1/2021 • 2 minutes to read

Occasionally, your pull request (PR) may appear stuck or frozen and can't be merged. When this happens, the Docs Build bot sends a generic error message that looks like this:



When you get this generic error, try these steps to resolve the issue:

1. Refresh the GitHub page in your browser.
2. Check the Docs portal for queue or performance issues. For example, check if builds are triggering in the repository as expected. Use your GitHub credentials to sign in if necessary.
3. Close and reopen the PR to try to trigger a build. At the bottom of the GitHub page for your PR, select **Close pull request**. Wait for GitHub to close the PR, and then select the button again after its label toggles to **Reopen pull request**.



4. Check GitHub's status.
5. Check Azure's status.

If none of these steps resolves the build error, post to the Docs Support channel for help.

Resolve merge conflicts in Git and GitHub

6/11/2021 • 11 minutes to read

What is a merge conflict?

A merge conflict sounds like a bad thing, but it really is a good thing - merge conflicts are part of the collaboration process.

Merge conflicts can occur in multiple situations, but boil down to the same thing. Different users have submitted changes that Git can't reconcile automatically. It requires human judgment to determine which is the right change. Or, human judgment may determine that the author has to rewrite the two changes to integrate both updates into the article.

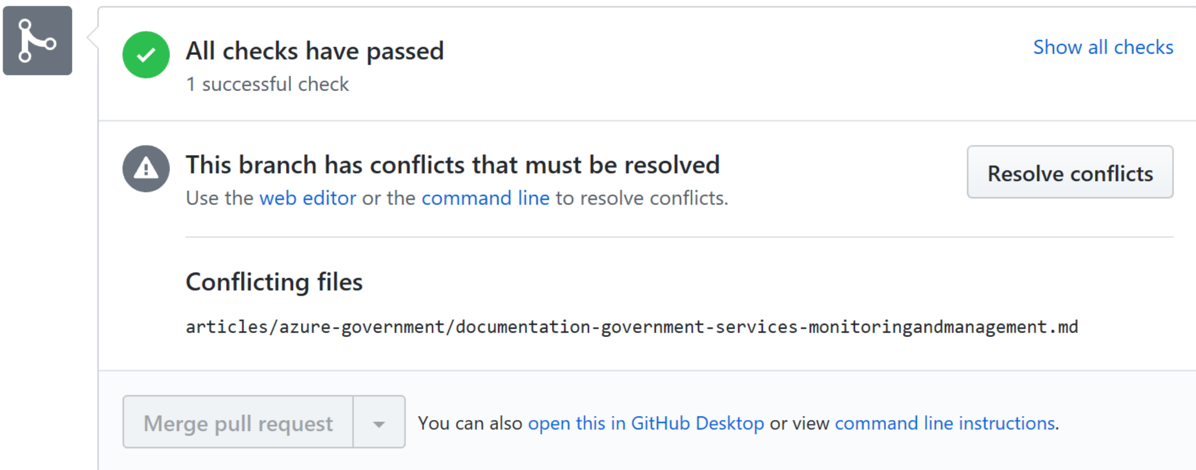
Git marks the merge conflicts in the source using conflict markers. Delete the markers and edit the text to be what you want. Sometimes, you pick one option in the conflict, in other cases, you have to delete both and rewrite the final text you want in the article. Here's an example of what the markers look like:

```
<<<<<<HEAD

==========

>>>>>>GUID-numbers-and-letters
```

This article addresses the common merge conflict scenarios and provides tips, tools, and warnings.

Caution using the GitHub UI to resolve merge conflicts

If you have write permissions in your repo, you'll have access to the GitHub merge conflict resolution tool. **Resolve conflicts** appears in pull requests when a merge conflict is present:



The only time it's safe to use this tool is if the pull request is against the `master` branch.

For a six-minute video about how to resolve a merge conflict in the GitHub UI, see: Resolve a merge conflict.

WARNING

Don't use the GitHub UI to resolve merge conflicts in pull requests filed against release branches.

When you commit the merge conflict fixes, GitHub automatically commits the contents of the release branch against master in a direct commit. This means you are unintentionally merging the release branch to master, causing your content to go live before you meant to.

Scenario 1 - Your pull request has a merge conflict

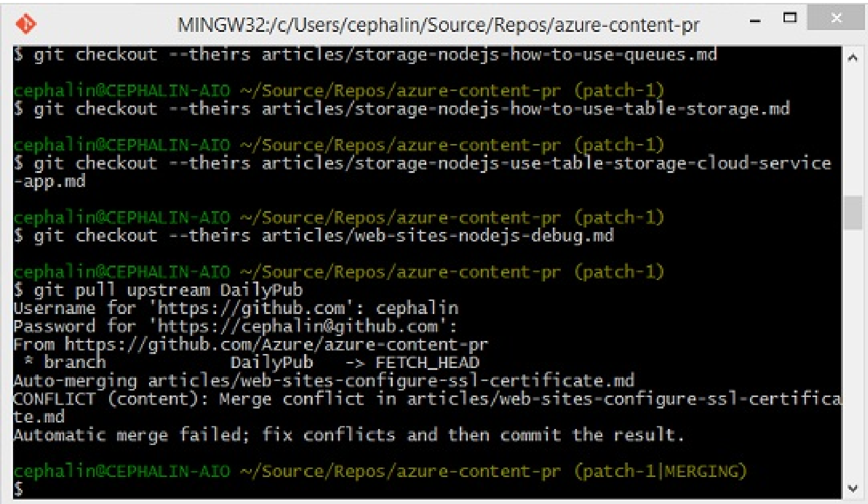
As a first step, make sure your local branch is in sync with the upstream master branch.

1. In Git Bash, make sure you're working in the local branch that is causing the conflict in the pull request. For example, if you're trying to merge content from your fork's 'snoopy' branch to the main repo's 'master' branch, make sure you're active in your local 'snoopy' branch.

2. Run our favorite command:

```
git pull upstream master
```

3. The prompt will probably change to include `| MERGING`, indicating you have to manually resolve the merge conflict. The command-line output should list the specific files that have a conflict:



If you can't tell which files have a conflict, run this command to list the conflicted files:

```
git ls-files -u  | cut -f 2 | sort -u
```

4. Open the conflicted file in a Markdown editor.

5. Find the merge conflict markers. Delete the text you don't want to keep. Then delete the merge conflict markers, including the beginning marker, the divider, and the end marker.

   If you forget to delete them, they'll publish in the article. The only text left from the conflict should be the correct text you want to show in the published article.

6. Save, add, commit, and push to your fork. Create a pull request to the appropriate branch in the upstream repository.

Sometimes, it will look like nothing happened - the prompt doesn't change. If it looks like nothing happened, just run the standard commands to add, commit, and push. Running the commands should unblock the pull request so it can be merged.

Scenario 2: You're trying to resolve a merge conflict in another person's pull request

There are many ways to accomplish the same task in Git. Here's two methods for pulling the contents of another person's pull request to your computer so you can resolve the merge conflicts. These methods are good to know because you can follow the same steps to pull another person's changes from a PR to your local computer.

Method 1 - Pull their branch

1. Create a new local branch from master, just like you do for normal, daily work, then check it out and let your fork know it exists:

```
git pull upstream master
git checkout -b <branch name>
git push origin <branch name>
```

2. Pull the other contributor's pending changes from their fork and working branch into your fork and working branch. When you enter the command, get the capitalization right - Git is case-sensitive:

```
git pull https://github.com/<contributors-github-account>/azure-docs-pr.git  <contributors-working-branch>
```

3. If there's a merge conflict, open the conflicted files, resolve the merge conflict, save, add, commit, and push.

Method 2 - Fetch the upstream references

1. Locate the PR number you want. GitHub list the PR number prominently in the pull request. It's also part of each pull request's URL: https://github.com/Azure/azure-content-pr/pull/**25614**.

2. Fetch that specific pull request and put that PR into a new local working branch by running these two commands:

```
git fetch upstream refs/pull/25614/head
git checkout -b 25614 FETCH_HEAD
```

3. Run `git pull upstream master` now.

4. List the files that have conflicts:

```
git ls-files -u  | cut -f 2 | sort -u
```

5. Resolve the conflicts, then add, commit, and push to YOUR remote origin (your fork). In this example, the command created a working branch named 25614, so the push command would be:

```
git push origin 25614
```

Scenario 3: You're resolving a merge conflict in your upstream release branch (authors)

In repositories where the Content & Learning Content Production Service (CPS) team provides publishing support, the team runs a script to update all the existing release branches with the latest content from the master branch. They run this script after each publishing run. Multiple times a week, they come across merge conflicts when they run the script. The merge conflicts prevent them from updating the affected release branch.

> **WARNING**
>
> **Do not use the GitHub UI to resolve merge conflicts in pull requests filed against release branches.**
>
> When you commit the merge conflict fixes, GitHub automatically commits the contents of the release branch against master in a direct commit. This means you are unintentionally merging the release branch to master, causing your content to go live before you meant to.

Historically, the CPS team resolved the merge conflicts to help the process along. However, the team lacks the domain and release-specific knowledge to accurately resolve the conflicts. To ensure correct conflict resolution, the writer or PM responsible for the content in the release branch needs to resolve these merge conflicts. The CPS team will contact you to let you know a merge conflict exists in your release branch. Once they do, here's how to resolve these sorts of merge conflicts:

1. On your computer, open Git Bash or Visual Studio Code. Then, change to the affected repository using the command line:

   ```
   cd <repo name>
   ```

2. Check out your local copy of the upstream release branch:

   ```
   git checkout <release branch name>
   ```

   If you don't have a copy of the release branch in your local clone, follow the steps in the Use release branches article to create it.

3. Update the local release branch with the latest upstream content:

   ```
   git pull upstream <release branch name>
   ```

4. Pull the master branch into your local release branch:

   ```
   git pull upstream master
   ```

   At this point, all the changes from master will come into your local release branch. The output of the command will list the files that contain conflicts.

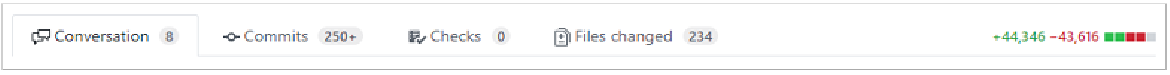
5. In Visual Studio Code, open each conflicted file, resolve the conflict, and save your changes.

   Sometimes, the files that show conflicts aren't files that are part of your content set. And, you didn't intend to change them in your release branch. If this problem happens, you need to check out the versions of those files that are in the master branch to restore them to the correct version. Run this command to restore the files to match the master branch:

   ```
   git checkout upstream/master articles/<service>/<filename>.md
   ```

6. At the command line, add and commit the changes. Then, push the changes to your fork.

7. Compare your branch with upstream master. This step helps to validate the changes you expect to see. When you look at your PR, GitHub is likely to display hundreds of commits and file changes. It makes it hard to learn what actually changed.



To see the diff of your branch and upstream master, use these commands:

```
git fetch upstream
git diff upstream/master --name-status
```

8. In the output of the `git diff` command, Git Bash displays the type of change for each line like this:

   - A: addition of a file
   - C: copy of a file into a new one
   - D: deletion of a file
   - M: modification of the contents or mode of a file
   - R: renaming of a file
   - T: change in the type of the file

   The following condensed example output shows two *modified* files, two *deleted* files, and one *addition* of a file:

   ```
   $ git diff upstream/master --name-status
   M .openpublishing.redirection.json
   M articles/active-directory-domain-services/TOC.yml
   D articles/active-directory-domain-services/active-directory-ds-admin-guide-configure-secure-ldap-enable-ldaps.md
   A articles/active-directory-domain-services/breadcrumb/TOC.yml
   A articles/active-directory-domain-services/compare-identity-solutions.md
   [...]
   ```

   Review the list of changes to upstream master. The only changes listed should be files that you, or others in your team, have worked on as part of your release branch. If there are other file changes, it's an indication you may have pulled across some other changes by accident.

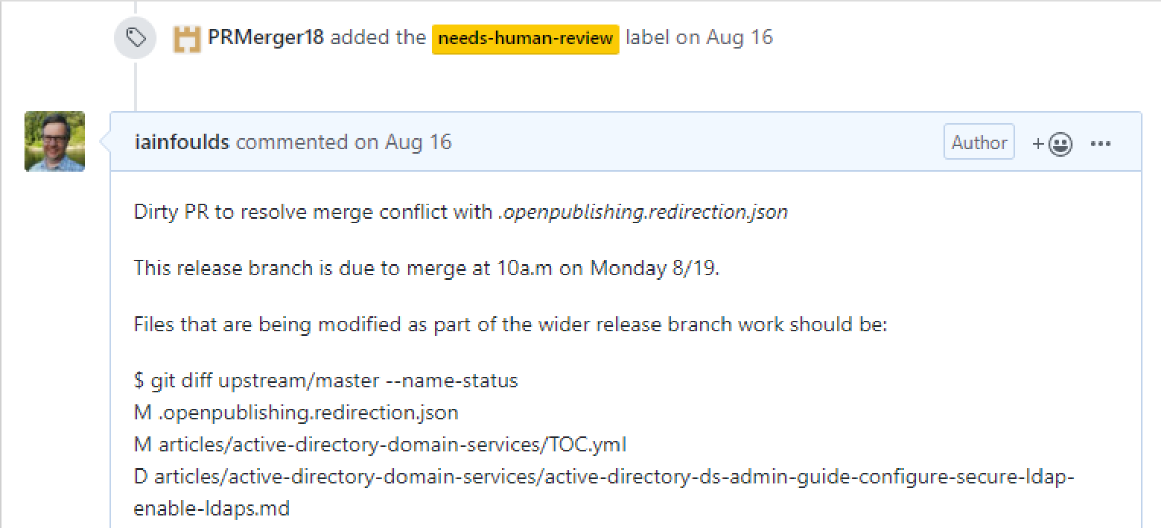
9. In GitHub, create a pull request *from* your forked copy of the release branch *to* the upstream copy of the release branch. Make sure you clearly title and describe the pull request so your repo admin knows what you intend the pull request to do. For example, *[ServiceName] Dirty PR for release branch merge conflict with upstream master*.

   This pull request contains commits and changes from other authors - they represent all the changes in master that you're bringing into the release branch via your fork. This type of pull request is known as a *dirty PR*. They aren't optimal, but in this case they're sometimes unavoidable.

10. After the PR passes build validation, check the staged articles in which you resolved conflicts. Make sure that you correctly handled the conflicts and the articles are rendering without any remaining merge conflict markers.

11. Sign-off on the pull request per the normal process.

    To help the repo admin or PR reviewer learn about the scope of your changes, include the output of your `git diff` commands and list of changes. If you've deleted or modified files outside of the core working area, add that note in the sign-off comments.



The repo admin or PR reviewer should carry out similar steps, shown in the next section, and the list of changes should align.

PR review team instructions for reviewing the PR created in Scenario 3

The steps in the previous section create a "dirty PR" that lists all the commits from master that haven't been merged to the release branch because of the merge conflict. PR reviewers need to follow these steps to ensure the pull request is safe to merge.

1. On your computer, open Git Bash and change to the repository the pull request is in.

2. Add the author's fork of the repository as a remote. This process relies on your GitHub personal access tokens:

```
git remote add <username> https://<your-user-name>:<your-access-token>@github.com/<username>/azure-docs-pr.git
git fetch <username>
git fetch upstream
git diff upstream/master...<username>/<branch-name> --name-status
```

   In the output, Git Bash codes the type of change for each line:

   - A: addition of a file
   - C: copy of a file into a new one
   - D: deletion of a file
   - M: modification of the contents or mode of a file
   - R: renaming of a file
   - T: change in the type of the file
   - U: file is unmerged (you must complete the merge before it can be committed)
   - X: "unknown" change type (most probably a bug, please report it)

As a PR reviewer, you're looking for anomalies to what seems to be the pattern in the pull request. For the dirty PRs that bring release branches up to date with master and fix merge conflicts between master and the release branch, look at these items:

- What is the release branch name? It usually indicates a service area, and the changed files typically will clearly map to the service area. If there's a large variety in the services affected, check the pull request for information and clues. If it's a release branch where branding updates are being made, you might see broad changes across services. For new feature releases, scope would normally be tight so you wouldn't expect to see broad changes across services. Ask the author if in doubt.

- In most cases, for a release branch, you're expecting mostly new and updated files. If you see deletions, ask the author to confirm that the deleted files for the release branch are intentional deletions.
- If there are no changes listed, something odd is going on. Most likely, the author made a mistake and resolved the merge conflict by making the release branch match the master branch. It's likely a mistake because it's eliminating any differences between the release branch and master. Check with the author to check if they intended to essentially wipe out differences in the release branch compared to master.

You're trying to resolve merge conflict with no change between branches

Sometimes contributors with write permissions to a repo will see another case of merge conflict with `no change`. In this special case, the target branch already includes everything in source branch and usually happens during merge of master into live branch. It occurs because of the below reasons:

- Master branch doesn't have a common history with live branch.
- One contributor created a PR from master to live branch. But another contributor already pushed the same change directly on live branch.

To resolve merge conflict on PR with `no changes` between master and live branch, just go ahead and merge the PR into live branch.

Use Visual Studio Code to resolve merge conflicts

Visual Studio Code includes a lightweight merge conflict resolution UI - once the conflict is present on your computer locally, just open the conflicted article. See the VSCode documentation for a screenshot and description of the merge conflict resolution UI.

Use Visual Studio to resolve merge conflicts

If you're running VS, you can resolve merge conflicts in a visual editor. See the Visual Studio documentation. You probably only want to resolve merge conflicts this way if you use VS for other work.

Break up a single large pull request into smaller PRs

5/12/2021 • 5 minutes to read

One cause of long turnaround times for pull request reviews and associated publishing delays is when a single pull request changes more than 100 files. These PRs require repository admin review before merging. Reviewing large PRs is also difficult to do in GitHub's web interface. PR reviewers often reject these PRs because all the requested changes cannot be reviewed.

This article provides proactive guidance on how to prevent these types of PRs. It also includes a tutorial on how to break down a PR with numerous changed files into smaller, reviewable chunks.

Avoid branches that update large numbers of files

Bulk text or link changes drive PRs with large numbers of changed files. Bulk find and replace actions can be dangerous. It's best if you carry out these changes and PR them per directory in your content repo so you can review every change before merging.

To avoid large PRs when teams are working together on a content release, use the release branch process and merge changes into the release branch. The team reviews the changes in the context of the overall shared effort as the work progresses instead of all at once when merging to the master publishing branch.

Fix a branch that updates a large number of files

To fix a pull request with many modified files, you'll need to be comfortable with using the Git command line and familiar with basic Git concepts. The goal is to take your branch with numerous changes and create a set of smaller branches that collectively have all of the changes of the original branch. You then open pull requests for each of those branches.

You can also use this process if you pushed one or more files to the wrong branch, or if you just want to clean up a PR by restricting it to files that have related changes.

> **NOTE**
>
> This process will not retain the original commit history of your changes once merged into the master publishing branch in your repo.

Prepare your local repo

Run the following commands to create a state in your repo where the changes in the PR are unstaged. These steps assume that `upstream` is the publishing repo and `origin` is your fork of the publishing repo.

```
git checkout master
git fetch upstream # this may be necessary (depending on your git config) to receive updates on
upstream/master
git pull upstream master

# Merge the feature branch into the master branch.
git merge <pull_request_branch>

# Reset the local repo to the current head of the publishing repo, leaving behind changed files as unstaged.
git reset upstream/master
```

If you run `git status` at this point, you'll be on the master branch in your local repo with a list of unstaged file changes. These file changes are the diffs between the pull request branch and `master`.

Group the unstaged files into multiple, smaller branches

1. Create a new branch off `master`:

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

    > **TIP**
    > If you're breaking up a large PR into more manageable chunks, use a similar branch name with an incrementing counter at the end to name the branches.

    You're now in the new branch, and the unstaged files have come along for the ride.

    ```
    ~/visualstudio-docs-pr (master)
    $ git checkout -b regroup-1
    Switched to a new branch 'regroup-1'

    ~/visualstudio-docs-pr (regroup-1)
    $ git status
    On branch regroup-1
    Untracked files:
      (use "git add <file>..." to include in what will be committed)

         file1.md
         file2.md
         file3.md
         file4.md
         file5.md
         file6.md

    nothing added to commit but untracked files present (use "git add" to track)
    ```

2. Add, or stage, a subset of the files using `git add` or your favorite graphical Git tool (like GitHub desktop or Visual Studio Code).

    ```
    ~/visualstudio-docs-pr (regroup-1)
    $ git add file1.md

    ~/visualstudio-docs-pr (regroup-1)
    $ git add file2.md

    ~/visualstudio-docs-pr (regroup-1)
    $ git status
    On branch regroup-1
    Changes to be committed:
      (use "git reset HEAD <file>..." to unstage)

         new file:   file1.md
         new file:   file2.md

    Untracked files:
      (use "git add <file>..." to include in what will be committed)

         file3.md
         file4.md
         file5.md
         file6.md
    ```

3. Commit the staged files:

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

4. Push the branch to `origin`:

```
git push origin <branch name>
```

5. Now, switch back to `master` branch:

```
git checkout master
```

The unstaged files will come along for the ride.

```
~/visualstudio-docs-pr (regroup-1)
$ git checkout master
Switched to branch 'master'
Your branch is ahead of 'origin/master' by 1326 commits.
  (use "git push" to publish your local commits)

~/visualstudio-docs-pr (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1326 commits.
  (use "git push" to publish your local commits)

Untracked files:
  (use "git add <file>..." to include in what will be committed)

     file3.md
     file4.md
     file5.md
     file6.md
```

6. Repeat steps 1-5 until you've staged and committed all of the unstaged files.

Create a pull request for each of the pushed branches

Create a pull request for each of the pushed branches on github.com. The sum of the changes in the new pull requests is equivalent to the file changes in the original (large) pull request, but since the changes are smaller it's more easily reviewed.

If you're submitting multiple pull requests for new articles, you can request an exception to the PR review criteria. The criteria requires every new article to have a TOC entry and submit the TOC in the last and final pull request. Email the C+L Pull Request Review Team team to request the exception. This exception applies only when someone's asked you to break up a large PR. Normally, you'd work iteratively and submit each new article with a new TOC entry. Each exempted PR must contain a link to the PR that contains the TOC updates for all the PRs.

Working with someone else's pull request

If you're working with a pull request opened from another user's fork, you first need to pull down their branch to your local repo. See Checking out pull requests locally for instructions on how to take another user's pull request and bring it down to a local branch. From there, you can follow the steps above in your local repo.

How to work with public pull requests

6/16/2021 • 9 minutes to read

Most content sets have two repositories - a public repo provided mainly for external community contributions, and a private repo for contributions from Microsoft contributors.

Employees working in the public repo

Content developers and core product team PMs should work in the private repositories so they can address quality feedback from the quality automation present in the private repos.

Microsoft employees from other teams can work in public repos at this time to ensure their contributions are reviewed by the article author. Or, work in the private repo, but request that the author review your changes using an @mention.

Workflow overview

1. The customer files a pull request in the public repo against one or more articles.
2. The contributor license agreement (CLA) automation runs and determines whether a CLA needs to be signed.
3. After the CLA requirements are met, the author listed in the metadata is notified of the pull request.
4. The author reviews the PR, thanks the contributor, and decides what the next step is:

   > **IMPORTANT**
   >
   > Reminder, changing steps in the articles may impact dependent repositories such as `Azure-Samples`. Please ensure to step through the edited article to ensure that everything is functioning as it will be once the contributions are merged.

   - The changes are correct and relevant as-is. The author lets the PR reviewer for the repo know to merge the pull request.
   - Changes are needed before the PR can be merged. The author works with the contributor to get the PR to the point where it can be merged and then lets the PR reviewer for the repo know to merge the pull request.
   - The author determines that the changes cannot be merged. The author lets the contributor know why, then asks the PR reviewer for the repo to close the pull request.
5. Where there are private/public repository pairs, the public repo and private repo are merged together as part of the publishing process.

Automation to support this workflow

Content & Learning has automation to support the workflow described in the previous section. After the automation is set up, the automation supports the process as follows:

1. When a PR comes in, the automation checks the GitHub account of the contributor to determine whether the listed author is the person submitting the change. The automation checks against the value entered in the **author** metadata attribute. If the listed author submits the changes, the PR is closed with a message that the author should use the private repo. If the contributor is not the listed author, the automation goes

to the next step.

2. The automation checks to see if the CLA requirements are met. If a CLA must be signed, the PR sits until the CLA is signed. If the changes are small and do not require a CLA, or if the CLA has already been signed, the automation goes to the next step.

3. The automation writes a comment to the pull request to acknowledge the contribution. This comment causes a GitHub notification to be sent to the person whose GitHub account is listed in the author metadata attribute. An additional notification and instructions email are also sent to the alias listed in the ms.author metadata attribute. Finally, the automation applies the `Change sent to author` label.

4. After review, these comments are used to drive the next action:

   - Type `#sign-off` to assign the ready-to-merge label (listed author only).
   - Type `#please-close` to automatically close the pull request (anyone).
   - Type `#sign-off` can be canceled with `#hold-off` (anyone).

   The comments run off a timed job, so there may be up to a 30-minute delay. If someone other than an author adds `#sign-off`, that person receives a message indicating only the listed author can sign off. If you are a subject matter expert qualified to evaluate the change, you can use the exception process to sign off on the pull request - send an email to the techdocprs alias and request it be merged. Let the reviewers know you are on the product team or writing team for the product or service.

5. The pull request reviewer for the repo queries for PRs that have the `ready-to-merge` label and merges them.

Pull public commits to the private repo for updates, staging, publication

This section of guidance applies in cases where your repo has both a public and private repo, and your primary workflow is in the private repo. This is currently the case for most Content & Learning teams.

Sometimes, you need to modify changes submitted by a community member, or you need to pull new articles and major rewrites submitted through a public repo to the private repo for fixing, staging, Acrolinx, and merge.

Build validation is not turned on in public repos at this time, and Acrolinx is not turned on. Neither tool works in public repos in a way suitable for customers. The majority of changes have typically been small in scope. To help ensure quality, when a new article or major rewrite is submitted through a public repo, you should pull the changes to your private repo and run it through the standard process of validation, staging, Acrolinx, and PR review.

You can use these basic steps to pull changes from a public to a private repo:

1. In your local private repo, create a new local branch from master, just like you do for normal, daily work, then check it out and let your fork know it exists:

   ```
   git pull upstream master
   git checkout -b <branch name>
   git push origin <branch name>
   ```

2. Pull the other contributor's pending changes from their fork and working branch into your fork and working branch. When you type the command, get the capitalization right - git is case-sensitive:

   ```
   git pull https://github.com/<contributors-github-account>/<public-repo-name>.git <contributors-working-branch>
   ```

3. Push to your fork again, and create a pull request to master. Once you have a successful build, review the staged content, make updates required to meet the Acrolinx min score, and sign off when you are ready. The PR review team will provide feedback if the PR review criteria require updates.

In case the other contributor plans to add more content before submitting a pull request to master, you can submit a PR to the same branch of contributor's fork instead of upstream. This way you can stack commits from different authors into a single branch before submitting a PR to master. When submitting a PR to the contributor's fork, it may not be available in the drop-down by default. However, you can submit the PR by updating the URL of the page. Here is the URL format to submit a PR from your fork to someone else's fork:

```
https://github.com/<contributors-github-account>/<public-repo-name>/compare/<contributors-branch-name>...
<Your-github-account>:<Your-branch-name>
```

There are plans under discussion around how to make the public and private workflows the same, provide the same quality support in both public and private, and make the workflow account for trusted and untrusted contributors. This will help ensure that contributions from external sources and internal sources get the right level of review and approval. However, we're not there yet.

Guidance for new articles contributed by external community

Process

For Azure and other product areas where pull request review is provided by the Aquent vendor teams, the vendors will contact the business approver for the product or service when a new article or significant rewrite is submitted in a public repository.

The FTE content team must work with the external contributor to determine how and whether to proceed. If the decision is made to publish the article, the FTE works with the external contributor to get the article as complete as possible in the public repo. Then, the FTE must pull the commits to the private repository and run the new article through build validation, staging, and Acrolinx. The new article must be merged via the private repo. After the commits are merged to live for publishing, the public pull request will be automatically closed.

Metadata and ownership of articles contributed by external community

There are two metadata attributes that describe who is accountable for an article. Many downstream processes and some automation are dependent on these values:

- **author**: Lists one value, the GitHub account of the author.

  If your repo is covered by current public repo PR automation, this person can sign off on pull requests against the article in the public repo, even if that person is not an employee. (This is a known issue.)

- **ms.author**: Email alias for the author; can accept multiple values.

To ensure that a Microsoft FTE signs off on all updates to the article, an FTE must be listed in the metadata as the author:

- **author**: The GitHub account of the FTE sponsor for the article.
- **ms.author**: The email alias of the FTE sponsor, and the full email address of the community author.

While not optimal, this will ensure an employee reviews any updates to the article that come in on the public side. The community contributor is recognized through their commit history to the article.

Customer etiquette

Be nice to customers who make contributions - use friendly language and remember they have taken time from of their day to contribute to our content! If you have to close a PR, let the contributor know why.

Finding open pull requests

To find open pull requests for your articles, use one of the following methods:

- Where the automation is running:
  - Watch your email for notifications and respond promptly.
  - In the Azure-docs-pr repo, you can filter the PR queue by service. For example, this PR query finds the open PRs for Azure Stack: `is:pr is:open label:azure-stack`. The service label is based on your service metadata.

SLA for responses to public PRs

- PRs that are 1-7 days old are green
- PRs that are 8-14 days old are yellow
- PRs that are 15 days old or more are red

Contributor license agreement (CLA)

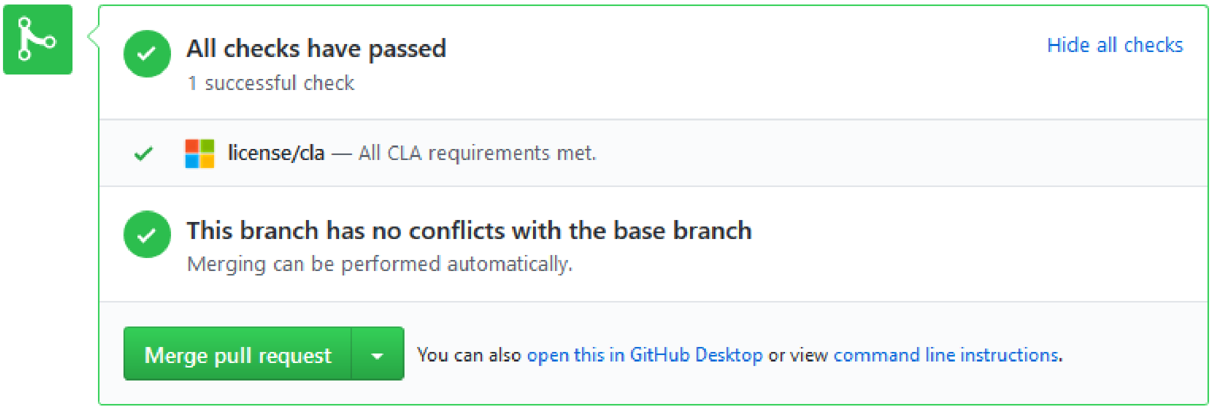
It's crucial that we work correctly with pull requests where a CLA must be signed. Here's the basics:

- Authors must not look at the content of a public PR if a CLA is required and not yet signed due to the legal issues around IP ownership.
- The CLA basically says that the contributor releases ownership of whatever IP the contributor is providing.
- A CLA must be signed when the contribution exceeds the CLA limit. It's about the size/scope of the change.

Here are some screenshots of how the new CLA service makes itself visible in pull requests.

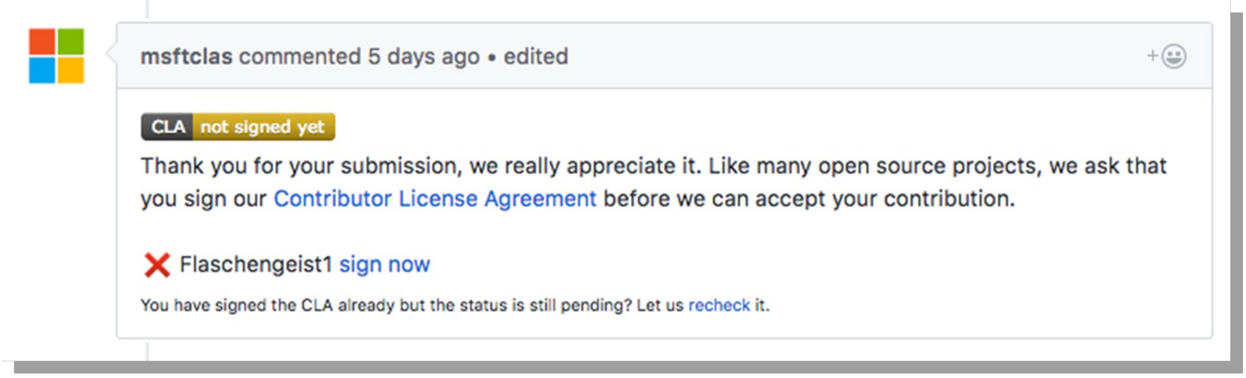
No CLA is required

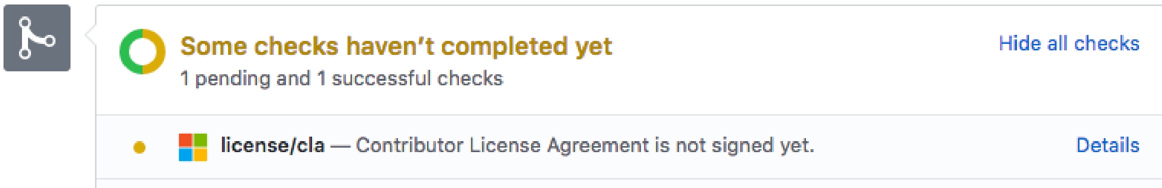
Only a status is provided. This applies when no CLA is required, and it applies when a CLA is required but the contributor already signed the CLA in an earlier PR. Microsoft employees whose identities in GitHub are linked to their MSFT identity also automatically pass the CLA requirement.



CLA is required and not signed yet

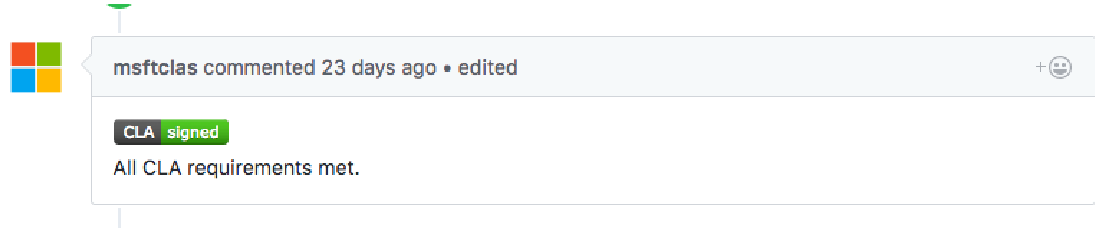
A comment and status are provided.

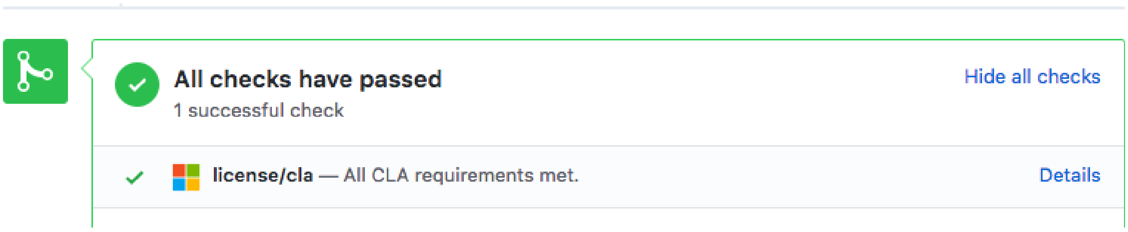




CLA is required and signed by submitter

A comment and a status are provided – this happens only in the PR in which they had to sign the CLA the first time.





Regular, out-of-band, and hotfix publishing

5/7/2021 • 8 minutes to read

This article is for repo admins, business approvers, and the Content & Learning Content Production team. It defines the three types of publishing, and it explains how to perform hotfix publishing. These tasks require write permissions or admin permissions in the repo you're working in. You must also have access to merge to the live branch if the live branch is protected in the repo.

Publishing definitions

Regular publishing: The established daily publishing cycle that happens every work day at approximately the same time for your repo. Here's how the regular publishing process works:

- For a public-only repo, merging the master branch to the live branch automatically triggers a publishing run to docs.microsoft.com.
- Where a public and private repo pairing exists, the private master branch is merged to the public master branch. Then, the public master branch is merged to the private master branch. The final step is to merge the private master branch to the private live branch, which automatically triggers the publishing run to docs.microsoft.com.

Out-of-band publishing: Like regular publishing, out-of-band (OOB) publishing is accomplished by merging the master branch to the live branch. However, an OOB publish is an extra planned publishing that is run outside the standard times. An OOB publish includes all the contents of the master branch. OOB publishing should be pre-planned, pre-announced, and infrequent. The actual publishing process includes all the elements of regular publishing, just at a time and day that is different from the normal schedule.

Early morning publishing runs for large events like Build and Ignite are considered OOB. Why? This is because they happen much earlier in the day than regular publishing, but they're planned and announced well before the date. OOB publishing events should be broadly communicated with content stakeholders. Other content owners may be counting on the regular publishing time. The repo admin must approve OOB publishing and must ensure that stakeholders have been warned/alerted adequately.

Hotfix publishing: Hotfix publishing is unplanned. It's for content that needs to publish outside a standard publish time with little or no prior notice. Unlike Regular and OOB publishing, only content that needs to be published is merged to the live branch. The entire master branch isn't merged. Hotfix publishing prevents content that is already in the master branch from going live before it should.

Hotfix publishing is typically for publish requests that come in after 3:00 PM, the last regular publish of the day for most repositories. You might also use Hotfix publishing during a content freeze for high priority changes that can't wait until the end of the publishing freeze. These publish runs don't involve a full publish cycle where public and private are cross merged.

Hotfix publishing also has some limitations. If you use the process to publish a new article, the **Edit** link on the article becomes unavailable until the next full publish cycle. But, the limited scope of hotfix publishing makes it a lightweight, easy, and fast solution for high priority unplanned publishing requests. Hotfix publishing requires business approver sign-off. Hotfix publishing is also subject to the availability of staff with the required permissions.

Hotfix publishing is reserved for legal, security, privacy, and executive escalations. **Hotfix publishing should be rarely used.** It isn't a replacement for good planning. Don't use the hotfix publishing process to publish large amounts of change. Hotfix publishing assumes you're targeting specific articles for publish.

Why does any of this matter?

If you work in a small repo with only a few contributors, the distinction between publishing types may not matter. But, if people from different teams work in your repo, and if your contributors are many, things can get complicated. The different teams and large number of contributors are constantly planning routine content releases around the planned publishing schedule. A team or contributor may have content in the master branch that is intentionally waiting for the next planned publish - the next morning, or for an event. An urgent unplanned publish to master will also push the other content live, unexpectedly. You can avoid this problem by being disciplined about planning OOB publishing. Use hotfix publishing only to move specific content live, leaving all other content waiting in master for the next planned publishing event.

General processes

General processes for planning your release appear in the Content release planning + process article in this contributor guide.

For Content & Learning, the Content Production team provides publishing support of different types and is typically limited to regular and OOB publishing. The process the Content Production team uses to run regular publishing is documented in the Content Production Service process guide.

How to run hotfix publishing

Prerequisites

To run hotfix publishing, you must meet one of the following requirements:

- You must be an admin of the repo.
- You must have write permission in the repo.

  And, if your live branch is protected, you must be explicitly named as a person who is permitted to merge the live branch.

  Or, you must be a member of a team that is explicitly permitted to merge to the live branch.

> **NOTE**
>
> Usually, if you are doing this work, you will know what level of access you have to the repo where you need to run hotfix publishing. You can tell if you are an admin in a repo if the `Settings` tab is visible to you. The most obvious way to tell if you are a write-level user is if the merge button is available to you in a pull request. Most repos have protected branches, so you may only be authorized to merge to certain branches.

Instructions

Follow these steps when the change you need to push live only includes a few commits:

1. Ensure the changes that you need to push live are merged to the master branch of the repo.
2. In your local clone of the repo, make sure you are in your local master branch:

   ```
   git checkout master
   ```

3. Fetch all upstream branches:

   ```
   git fetch upstream
   ```

4. Depending on your situation:

   - If you already have a copy of the live branch on your local computer, change to your live branch:

     ```
     git checkout live
     ```

   - If you need to create the live branch locally, enter:

     ```
     git checkout -B live upstream/live
     ```

     The latter part of this command causes your local live branch to track the upstream live branch.

5. Update your local live branch from upstream:

   ```
   git pull upstream live
   ```

   > **IMPORTANT**
   >
   > **DO NOT** allow your muscle memory to cause you to enter the pull command against master. Don't do it!
   >
   > If you do, run this command to reset your local live branch to match the upstream live branch:
   >
   > ```
   > git reset --hard upstream/live
   > ```

6. Now, find the pull request that has the changes you need to push live. Use `git cherry-pick` to move each relevant commit in order into the live branch, starting with the oldest:

   ```
   git cherry-pick <commit>
   ```

   Repeat for each commit.

7. After you cherry pick all the relevant commits, run:

   ```
   git status
   ```

   There should be nothing listed - go to step 9. However, if the changes are listed in green in the status message, you need to commit them.

8. Commit your changes:

   ```
   git commit -m "picked commits for hotfix publish"
   ```

9. Push the changes to your fork:

   ```
   git push origin live
   ```

10. Create a pull request from your fork's live branch to the live branch of the upstream repo, and merge it.

    Because the changes you moved to the branch were reviewed and validated in the master branch, it's OK to merge this PR yourself after you verify the changes.

Larger changes

If the changes you need to push live are more extensive than a few commits, you can use the `git checkout` command instead of `git cherry-pick` for hotfix publishing.

For example, say you want to move all the content from the iot-central folder. And, you also want to move a few related changed files to live. But you don't want to publish the rest of the changes from the master branch.

1. Follow steps 1-5 in the preceding section.

2. Then, use `git checkout` to copy the changes you want from your local master branch to your local live branch. Here an example of how to move all the content from one folder along with three other individual files:

   ```
   git checkout master articles/iot-central
   git checkout master bread/toc.yml
   git checkout master docfx.json
   git checkout master includes/iot-central-howto-raspberrypi-selector.md
   ```

3. Add and commit your changes, then push them to the live branch in your GitHub fork. Create a pull request from your fork's live branch to the live branch of the upstream repo.

If the other methods described in this article don't work, you can merge the entire contents of a pull request into your local live branch. You need to be certain about the contents you're bringing to the live branch. The `git cherry-pick` and `git checkout` commands are more precise and safer because they target specific, known files. Check the diff in your pull request with extra care. To pull an entire pull request to your local live branch, follow the steps for cherry picking in the preceding section, but use the following command in step 6:

```
git pull upstream pull/<PR#>/head:<users-branch-name>
```

Caution

Bringing in the changes by targeting a pull request risks merging the contents of the master branch or other release branches. It's *not* the preferred method.

Checking on build status

You can check the status of your build in the Docs Portal. Search for your docset (the docset for azure-docs-pr is azure-documents), and select the **Builds** tab. Under **Build History**, filter on the **live** branch in the **Select a branch** box. Builds may have to wait in the queue for up to 50 minutes during busy times. And, the build may then take from 15 to 30 minutes, depending on usage of the publishing platform. Outside work hours, queues and build times are often faster.

Pull request review and publishing schedules

5/10/2021 • 3 minutes to read

Pull request review schedule

Small, text only changes to existing articles are merged automatically in repositories where PRMerger runs. For more information about which pull requests qualify for automerge, see PRmerger filters.

The vendor PR review team is available six days a week.

Redmond Hours

DAY	PR QUEUE COVERAGE TIMES
Sunday	12:00 AM - 4:00 AM, 2:00 PM - 4:00 PM
Monday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM, 7:00 PM - 9:00 PM
Tuesday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM, 7:00 PM - 9:00 PM
Wednesday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM, 7:00 PM - 9:00 PM
Thursday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM, 7:00 PM - 9:00 PM
Friday	12:00 AM - 4:00 AM, 7:00 AM - 5:00 PM
Saturday	no coverage

NOTE

All times are Pacific time (Redmond). Sunday afternoon hours are approximate and may be shorter or longer depending on the number of pull requests in the queue. The Sunday early morning shift begins September 2019.

Israel Hours These are the same times as the Redmond hours listed above, displayed in Israel timezone for the ILDC team's convenience.

DAY	PR QUEUE COVERAGE TIMES
Sunday	10:00 - 14:00
Monday	00:00 - 02:00, 10:00 - 14:00, 17:00 - 00:00
Tuesday	00:00 - 03:00, 05:00 - 07:00, 10:00 - 14:00, 17:00 - 00:00
Wednesday	00:00 - 03:00, 05:00 - 07:00, 10:00 - 14:00, 17:00 - 00:00

DAY	PR QUEUE COVERAGE TIMES
Thursday	00:00 - 03:00, 05:00 - 07:00, 10:00 - 14:00, 17:00 - 00:00
Friday	00:00 - 03:00, 05:00 - 07:00, 10:00 - 14:00, 17:00 - 00:00
Saturday	00:00 - 03:00

Daylight savings time is observed in both Israel and the US, but the effective dates are slightly off - the US goes to DST in early March, Israel in late March. The US reverts to regular time the first Sunday in November while Israel reverts one weekend earlier.

Publishing schedule - Azure-docs-pr

Redmond Hours The publishing schedule for Azure technical documentation in the azure-docs-pr repository is:

DAY	TIME	TYPE
Sunday	4:00 AM and 4:00 PM	Private repo
Monday - Friday	4:00 AM, 10:00 AM and 3:00 PM	Both public and private repos

Israel Hours These are the same times as the Redmond hours listed above, displayed in Israel timezone for the ILDC team's convenience.

DAY	TIME	TYPE
Sunday	14:00	Private repo
Monday - Friday	14:00 and 20:00	Both public and private repos
Tuesday - Saturday	01:00	Both public and private repos

Publishing schedule - other repos

Many other repos have a morning publishing run between 9:30 and 10:30, and an afternoon run between 2:30 and 3:30. The complete list of repos and schedules is posted on the Content Production Service site.

Publishing team - hours available (PST)

The publishing team is working on requests and available for contact during the hours below. (File requests here. Contact here: pubdesktech@microsoft.com.)

DAY	WORKING HOURS
Sunday	6:00 PM - 2:00 AM (next day)
Monday	9:00 AM - 2:00 AM (next day)
Tuesday	9:00 AM - 2:00 AM (next day)

DAY	WORKING HOURS
Wednesday	9:00 AM - 2:00 AM (next day)
Thursday	9:00 AM - 2:00 AM (next day)
Friday	9:00 AM - 6:00 PM
Saturday	no coverage

Holiday schedule

Limited pull request review and publishing will occur on the following Microsoft US corporate holidays to support authors working outside the US:

DATE	HOLIDAY	PR REVIEW AND PUBLISHING
January 18, 2021	Martin Luther King Jr. Day	Yes - limited hours
February 15, 2021	President's Day	Yes - limited hours
May 31, 2021	Memorial Day	No
July 5, 2021	Independence Day	No
September 6, 2021	Labor Day	No
November 25, 2021	Thanksgiving Day	Yes - limited hours
November 26, 2021	Day after Thanksgiving	Yes - limited hours
December 23, 2021	Christmas Eve	Yes - limited hours
December 24, 2021	Christmas Day	No
December 26-31, 2021	Holiday week	Yes - limited hours
January 1, 2022	New Year's Day	No
January 3, 2022	Resume normal schedule	Yes - back to normal schedule

Troubleshooting Git errors

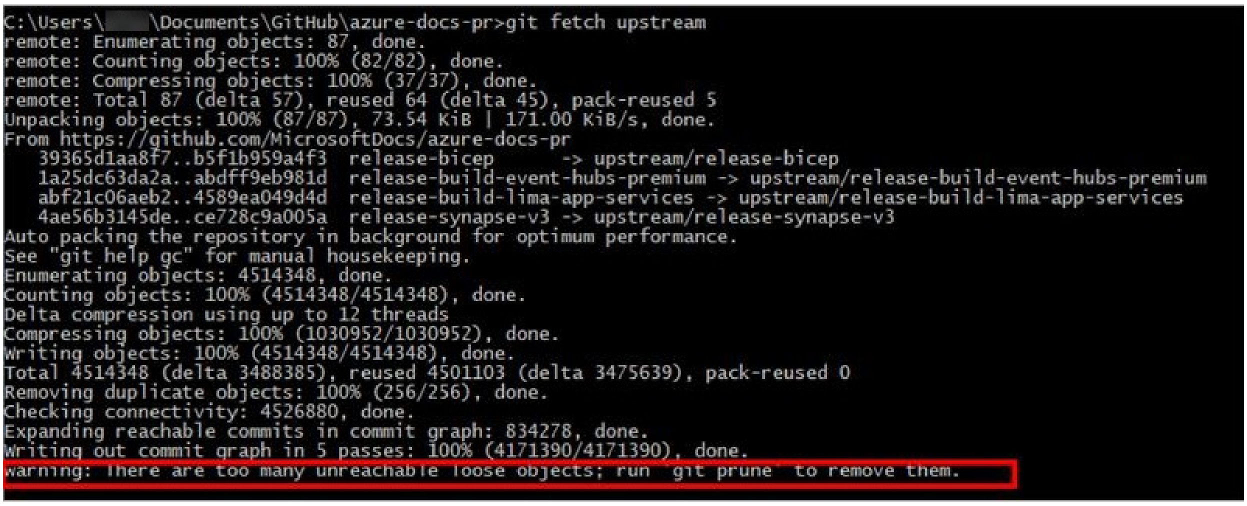
5/21/2021 • 2 minutes to read

This article provides instructions to resolve some Git specific errors you may get when updating content.

Git warning: unreachable loose objects

When you fetch upstream or push a commit in Git, the following error appears:

`warning: There are too many unreachable loose objects; run git prune to remove them`



This warning appears when there are too many orphaned commits, known as dangling commits. For example, any commit that cannot be accessed through a branch or tag is considered dangling.

To resolve this warning, run the *git prune* command:

```
git prune
```

Or a if you prefer, run the *git gc* were gc stands for garbage collection) command, which will not only clean up the dangling commits but will also do a compression on stored Git Objects, freeing up precious disk space.

```
git gc
```

After running the command, try to fetch or push again and your warning will be resolved.

Data, BI, and Reports overview

3/5/2021 • 2 minutes to read

A dashboard of available Content & Learning reports is available at https://aka.ms/clcia.

Data scenarios

In general, the reports and dashboards available provide insights at two different tiers of granularity: at the portfolio level and at the article level.

Portfolio-level metrics

Metrics on a portfolio of article (such as all of the articles published to docs.microsoft.com, the articles in the azure-docs-pr repo, or the articles for a specific service) are typically gathered for partners and stakeholders who need a high-level view of content health.

QUESTION	RELEVANT METRICS	REPORT
What is the overall traffic or CSAT for a content set or section over time?	PageViews, Visits, Unique visitors, CSATHelpfulRate	Documentation and reference report
Are customers engaging with the docs or are they reaching a page and leaving right away?	Bounce rate	Documentation and reference report
How is traffic coming to the doc set?	Referrer types	Documentation and reference report
How many GitHub issues do we have by repo? What types of issues are customers having?	Open and closed GH issues	GitHub issues report, GitHub

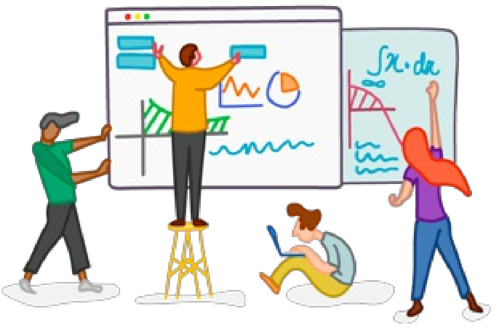
Article-level metrics

QUESTION	RELEVANT METRICS	REPORT
What is the overall traffic or CSAT for a specific page or section over time?	PageViews, Visits, Unique visitors, CSATHelpfulRate	Content Performance report
What are customers saying about my article?	Verbatims, GitHub issues	Verbatims report, GitHub issues report, GitHub
How is traffic coming to my section? A specific page?	Referrers, Referrals	Content Performance report, Microsoft Docs Metrics (MDM)
What are customers engaging with on the page? Or are they reaching a page and leaving right away?	Heat map, click data, Bounce rate	Content Performance report, Microsoft Docs Metrics (MDM)
Which keyword searches are driving traffic to my section? Which keywords are driving to a specific page?	Internal and external search terms	Microsoft Docs Metrics (MDM), Google search analytics

What is Content Data Advocacy Group (CDAG)?

3/5/2021 • 2 minutes to read

CDAG is a collective of champions across all of Content & Learning focused on content health & performance.



Strategy & working groups

Beyond the general data and process work we perform, we also have several working groups working on these initiatives:

1. Promote clearly understandable Content Health & Performance: We help define and update health and performance metrics
2. Drive consistent use of supported, standardized metrics/reports
3. Educate peers on the best practices to using these reports and metrics
4. Engage, evangelize, and communicate with our org through fun demo days, show-and-tells, newsletters, and more.

For more information on CDAG:

- See the CDAG FY21 strategy document
- See the docs and presentation archives
- Sign up for our newsletter by joining the DL: cdagnews@microsoft.com

Resource cheatsheet

This list of culled resources is a great place to start when learning or in need of support:

Reach CDAG

Email CDAG at cdag@microsoft.com.

Learn to use C&L reports

Learn more about our metrics and each supported BI report in this guide.

Find reports/tools

- Get in-browser page data using the [Microsoft Docs Metrics|aka.ms/get-mdm extension tool.
- Open our collection of BI reports at aka.ms/clcia.

Report bugs & issues

You can use the Analytics Teams channel to:

- Get help with the existing data and reports.
- Report bugs in the existing data and reports.

Request changes or new reports

Open a request in our DevOps board at aka.ms/dashboard-requests. When submitting your request:

- Add clear descriptions
- Add examples of what you want
- Include a business rationale for the request

Membership & roster

Are you committed to making a difference in the data maturity and literacy in C&L? Do you want to join? If you are interested, review the membership requirements in the CDAG Strategy doc with your manager. Then, send a note to cdag@microsoft.com.

In October 2020, the CDAG roster contained the following names:

ROLES	NAMES
Sponsors	Jennifer Cheung, C&L Martin Ekuan, Docs Mei Liang, Analytics Saurabh Dwivedi, OE & G
Leads	Jill Reinauer, Health + Perf Josée Martens, Engage + Comms Kelly Pittman, Education Maggie Sparkman, Consistent Reports Peter Lu, Consistent Reports
OE&G	Shweta Agarwal Suzanne White
Patterns & Practices	Dhanashri Kshirsagar Janet Thomas
Learn	Martha Denham
CCS	Mike Kinsman
Docs	Bobby Reed Matthew Baldwin Ryan Thompson Ryan Wike

Metric definitions

5/4/2021 • 16 minutes to read

Here are the definitions of the metrics most commonly used in our reports.

Traffic metrics

- PageViews
- Visits
- Visitors
- OrganicSearchReferrers

Composite metrics. These take several other metrics as inputs:

- PrimaryKPIScore (and PrimaryKPIScoreRank)
- Percentage of page views to higher performing content
- Freshness

Primary metrics. Used to directly calculate an article's KPI score:

- ClickThroughRate
- CopyTryScrollRate
- BounceRate
- ExitRate
- LearnModuleCompletionRate
- LearningPathCompletionRate

Secondary metrics. Not used to calculate the KPI score, but useful for gaining insights into the content:

- CopyRate
- TryRate
- ScrollRate
- MedianMaxScroll
- DwellRate
- StartTutorialRate
- UserEngagementRate
- AvgCopyTryEngagementRate
- DeployClickRate
- CSATHelpfulRate
- CSATResponseRate
- GitHubOpenIssueCount
- GitHubTotalIssueCount
- GitHubNewIssuesCount
- RatingVerbatim

Referrer metrics. Use to determine the origin of incoming traffic

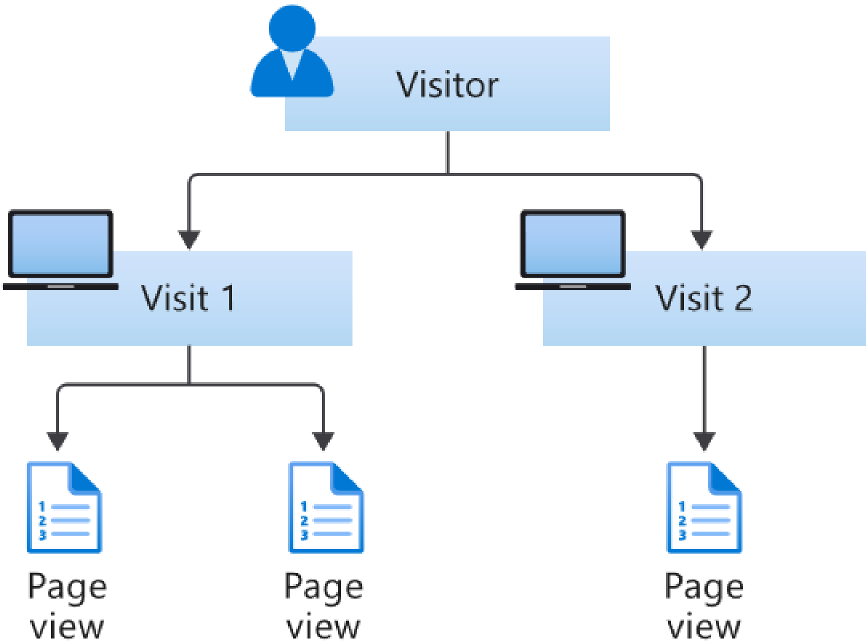
- OrganicSearchRate
- PaidSearchRate

- DirectTrafficRate
- MicrosoftTrafficRate
- ExternalTrafficRate
- SocialMediaTrafficRate

Traffic metrics

Why they're important: Indicates the discovery and awareness of the content. Growth in traffic should normally correspond with adoption and usage of the product or service the content covers.

How to improve them: Two ways: 1) Optimize for SEO to ensure your content is as discoverable through organic search as possible; and 2) Ensure strong engagement and satisfaction with the content to build trust with users and a stronger likelihood they will return to the site and the same area of content the next time they need help or information.



PageViews

Every time a page (article) is loaded.

Visits

A sequence of page views in a sitting. A session starts when the user first arrives on a site and ends after 30 minutes of inactivity.

Visitors (MAU)

An individual user to a site. Unique users are identified via MC1 web cookies, which are created for each user-browser-machine. Therefore, the same user could be counted as two unique users if they visited the same site from different browsers. If a user clears cookies in the browser, it will then be a new MC1 visitor.

Note that visitors are what make up the monthly active user (MAU) metric in organization-level reporting.

Don't sum visitors from multiple pages together or you'll unintentionally over count users who visited more than one of the pages. The data is deduped on the data collection side. Instead use https://aka.ms/docrefdashboard.

OrganicSearchReferrals

The page views from organic search referrals.

Composite metrics

PrimaryKPIScore (and PrimaryKPIScoreRank)

The PrimaryKPIScore (and affiliated metric, PrimaryKPIScoreRank) appear in the Content Performance report.

Definition: This is a score that lets us roll up the primary metrics into a single number we can rank when we compare articles against each other to determine which ones are lower performing. The topics will be ranked by the primaryKPIScore in a type group for us to identify the relative lower performing topics in the set of content by flagging if the rank is below 25%.

Formatting: In the content performance dashboard results, articles with `PrimaryScoreRank` highlighted in light yellow are performing in the bottom 25% when compared to articles of the same topic type. Articles highlighted in dark yellow are performing in the bottom 10% when compared to articles of the same topic type.

Why it's important: We use this score ranking to determine whether an article is lower performing versus other articles of the same topic type. After you sort to find the most-viewed articles in your content area, the score allows you to identify articles where your efforts are most likely to produce impact.

How we calculate the primary KPI value: Here's how the primary KPI rollup score is calculated. In the end, the calculations may seem arbitrary, but they are effective because the resulting primary KPI score lets us compare the relative performance of thousands of articles and identify the lowest performing ones.

Rationale: The desired behavior for most pages is for customers to come to a page, read to learn or understand new concepts, click through to explore more, or copy code snippets. We don't want them to bounce away! Landing pages have one purpose, and that is to help a customer find what they are looking for and click through to an actual technical article. For this reason, the 'good' half of the equation is focused on click-through rate. We don't want users to exit a landing page, and we definitely don't want them to bounce, so we subtract those two bad things from the click-through rate. We multiply the click-through rate by 2 because we are subtracting two negative metrics from the positive metric.

KPI formulas by type group

TYPE GROUP	TOPIC TYPE	LEGACY CALCULATION	NEW CALCULATION
Article	article	DwellRate + ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Conceptual	conceptual	DwellRate + ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Overview	overview	DwellRate + ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Quickstart	quickstart	ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Tutorial	tutorial	ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Sample	sample github-sample	ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate

TYPE GROUP	TOPIC TYPE	LEGACY CALCULATION	NEW CALCULATION
End user help	end-user-help	DwellRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
How to	how-to	DwellRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Troubleshooting	troubleshooting	DwellRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Hub/Landing	hub-page landing-page sample index	2 * ClickThroughRate - (ExitRate + BounceRate)	No change
Reference	reference language-reference managed-reference ioctl structure callback enum enumeration function include interface macro method struct error reference	ClickThroughRate - BounceRate	CopyTryScrollRate - BounceRate
Others	Topic types not on the metadata Allow list or is missing	DwellRate + ClickThroughRate - BounceRate	CopyTryScrollRate + ClickThroughRate - BounceRate
Learn module	interactive-tutorial	LearnModuleCompletionRate	No change
Learning path	interactive-tutorial	LearningPathCompletionRate	No change

NOTE

- If a topic type isn't covered in the content performance model, it's grouped with **others**.
- Approved ms.topic types are listed in the ms.topic allowlist.
- Descriptions for common article types are listed in the metadata attributes article of this Contributor Guide.
- The PnP Topic Type groups example-scenario, reference-architecture, design-guide were deprecated from reporting logic in Feb 2021.

Exception tracking

- **SampleIndex**: Articles where ms.topic=article and the title contains the word "Samples" are treated as SampleIndex.
- **Troubleshooting**: ms.topic starts with 'troubleshooting' or title contains 'troubleshoot' are treated as

TroubleShooting. These customizations were created at a particular point in time; ideally correct metadata would be assigned directly in the articles.

Percentage of page views to higher performing content

The total number of page views (PVs) to pages whose Primary KPI Score metric is above the bottom 25% versus its peers, divided by the total number of page views

- **Caveats**: The bar can be set anywhere. It's currently at the bottom 25%. As we increase the % of PVs to lower performing content, we'll ratchet the bar up.
- **Why it's important**: This is a rollup metric for topic-level performance that helps us steer our efforts across broad sets of content.
- **How to improve it**: Focus on improving high-traffic articles that under perform versus their peers.

Freshness

How many days have passed between the Last Review Date to the end of reporting month, categorized as: A: 0-30, B: 31-60, C: 61-90, D: 91-120, E: >120 days.

- **Why it's important**: This metric provides context for the primary metrics - for example, an article that has poor performance may just need to be updated to fix outdated screenshots, outdated procedures, broken links, and other things that could be blocking or discouraging users. For more information, see the content maintenance checklist/freshness article.

Primary metrics

ClickThroughRate

The ClickThroughRate for an article is equal to (# of page views with click activity) / (# of total page views). All link click events are counted regardless of the link target (links in the body of the article along with navigation, TOC, in this article).

- **Formatting**: This metric is a higher-is-better metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Caveats**: Clicks on the **Copy** or **Try** buttons for code blocks aren't calculated in this metric.
- **Why it's important**: We use this metric on several page types. If we put links and buttons in front of users, we want them to click those links and buttons.
- **How to improve it**: Use the MDM browser extension or the Click report to drill into clicks on the page. Check to see if some links lower on the page outperform links higher on the page. Moving them up, where appropriate, can reduce the exit rate and increase the click-through rate. Also, consider clearer and more actionable labels on links and buttons to sell the outcome of clicking through. Consider linking to other things of potentially more interest and removing links that aren't of interest.

CopyTryScrollRate

% unique visitors who copy, click try, or scroll on the page / total unique visitors to the page

- **Definition**: Engagement is counted whenever one of the actions (copy, try, scroll) happens no matter which one is first or how many times a user takes the same action
- **Formatting**: This is a higher-is-better metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Why it's important**: We use this metric to measure the amount of overall page engagement. Did customers find something on the page or did they copy code?
- **How to improve it**: If the BounceRate is high, this lowers the CopyTryScrollRate, so address BounceRate

/ SEO issues. For additional guidance, see Tips for low copy-try-scroll rate.

BounceRate

% of page views with a duration less of than 5 seconds and without clicks, scrolling, or copying of content.

- **Formatting**: This is a higher-is-worse metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Caveats**: The industry standard definition is a little different. Typically, it's defined as "% of page views that had only one page views in the session."
- **Why it's important**: We use this metric on all topic types. Leaving a page within 5 seconds without engaging most likely means a poor customer experience.
- **How to improve it**: Do a referrer analysis. Often a particular referrer is to blame for poor performance because they're pointing users to the wrong content to answer their questions. Work with referrers to point to better content or change the content to meet the expectations of that referred traffic. Lastly, consider clearer page titles and intro paragraphs to set expectations. If users can't tell what's in it for them right away, they'll bounce.

ExitRate

% of page views that are the last page view in a session. Currently ExitRate is only listed for hub and landing pages in the table on the Documentation page of the Content Performance Dashboard report. To find ExitRate for all your articles, go to the Overview page of the Content Performance Dashboard report. Select **Download the Excel Report** and choose either high-performing or low-performing articles. It's a large file, so it'll take a while to download.

- **Formatting**: This is a higher-is-worse metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Why it's important**: We use this metric on navigation pages like hubs, which are designed to get users to other pages. Exiting the site while viewing these pages is sometimes okay, but we don't want high-traffic navigation pages to have an above average exit rate.
- **How to improve it**: Users tend to exit when they can't find what they're looking for. Use the MDM browser extension to drill into clicks on the page to see if some links lower on the page outperform links higher on the page. Moving them up where appropriate can reduce the exit rate and increase the click-through rate.

LearnModuleCompletionRate

% of users who complete the module after starting it

- **Formatting**: This is a higher-is-better metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Why it's important**: We want to know if users are completing the modules.
- **How to improve it**: Consider shorter and more engaging content that creates *Aha* moments sooner.

LearningPathCompletionRate

% of users who complete the learning path after starting it

- **Formatting**: This is a higher-is-better metric. Light yellow identifies the bottom 25% and dark yellow identifies the bottom 10% among articles of the same type group.
- **Why it's important**: We want to know if users are completing the units.
- **How to improve it**: Consider shorter and more engaging content that creates *Aha* moments sooner.

Secondary metrics

CopyRate

Visitors who have a Copy event / Total visitors on the page

Definition: The percent of visitors that trigger a copy event for a page. Copy events include clicks on Copy button for code snippets, by clicking Copy in the right-click context menu, or key combination CTRL+C)

TryRate

Visitors that have a Try event / Total visitors on the page

Definition: The percent of visitors that trigger a try event for a page.

ScrollRate

% of visitors who have scroll event / Total visitors on the page

- **Why it's important**: Use this metric on any prose-heavy topics to see if users are engaging with the topic. A low percentage could indicate customers stayed longer than 5 seconds but didn't feel compelled scroll down the page and go below the fold.

MedianMaxScroll

Median max scroll depth, based on page views (one PV has one MaxScroll)

- **Why it's important**: Use this metric on any prose-heavy topics to see how far on the page users scrolled. Is there a point where most users leave the page?

DwellRate

Median dwell time/Estimated read time

- **Definition**: Dwell time is how long (in seconds) a user stayed on the page. Estimated read time is calculated during the Build process as 200 words/minute with a minimum of 2 minutes. It includes all text within HTML tags (monikers, code snippets, ALT+text, INCLUDES, tabbed conceptual, and zone pivots).
- **Why it's important**: We use this metric on any prose-heavy content types. Do users stay long enough to read it?
- **How to improve it**: Consider shorter content. Users often scan before they dive in. If it looks TLDR (too long, didn't read), they might leave.

Note: In June 2020, this was changed from **DwellTimeDelta** (MedianDwellTime – Estimated reading time)/Estimated reading time

StartTutorialRate

% of users who start a tutorial from a quickstart page

- **Caveats**: Intended quickstart design includes this tutorial button. Some pages may not include such a button/link, and get penalized by this metric. Docs signed off on continuing to include these in the reporting and not trying to exclude them.
- **Why it's important**: We use this metric on quickstarts. Do users start a tutorial after the quickstart?
- **How to improve it**: If the page doesn't include this button, consider adding one. Helping users learn something is a good thing. If it does include this button, consider a more descriptive call to action on the Tutorial button/link/intro. Use strong verbs and sell the outcome of completing the tutorial: what can you do after learning the next thing... or, try linking to a different tutorial.

UserEngagementRate

% of users who click at least one copy/try link (includes all copy events like right-click, Ctrl+C, and so on)

- **Caveats**: We eventually want to measure % of links on the page clicked, as a proxy for completion rate. This metric is interesting to consider in the meantime.
- **Why it's important**: We use this metric on any article types with steps in a procedure, and samples of code to copy/try. If we include these elements, we want users to click them. What % of users do?
- **How to improve it**: If the page doesn't include this type of UI control or doesn't encourage copying, consider relabeling the article type so we can measure it more appropriately. See roadmap request around "no-code" quickstarts, tutorials.

AvgCopyTryEngagementRate

% average total clicks on copy/try links per user (includes all copy events like right-click, Ctrl+C, and so on)

- **Caveats**: We eventually want to measure % of links on the page clicked, as a proxy for completion rate. This metric is interesting to consider in the meantime.
- **Why it's important**: We use this metric on any article types with steps in a procedure, and samples of code to copy/try. If we include these elements, we want users to click them. How often do users click these elements?
- **How to improve it**: If the page doesn't include this type of UI control or doesn't encourage copying, consider relabeling the article type so we can measure it more appropriately.

DeployClickRate

% clicks on deployable resource per page view

- **Caveats**: Some pages may not include such a button/link, and get penalized by this metric. AAC signed off on continuing to include these in the reporting and not trying to exclude them.
- **Why it's important**: We use this metric on reference architectures. Do users click the deployable resource?
- **How to improve it**: Consider placement on page or stronger calls to action. Consider the relevance of the scenario.

CSATHelpfulRate

% of responses saying the content was helpful divided by total responses

- **Formatting**: Light yellow is below mean. dark yellow is a standard deviation below mean.
- **Caveats**: Null values appear for pages that don't ask for feedback.
- **Why it's important**: This metric is our only quantitative attitudinal metric. It's always worth reading the verbatims, but with such low response rates, we don't include it in primary metrics.
- **How to improve it**: Read verbatims and act on them as much as possible.

CSATResponseRate

(CSATTotalResponses * 1.0 / Visitors) - % users that rated a topic divided by all users

- **Caveats**: Null values appear for pages that don't ask for feedback.

GitHubOpenIssueCount

Total number of issues open for the article as of the end of the reporting period.

GitHubTotalIssueCount

Total number of issues (open + closed) for the article as of the end of the reporting period.

GitHubNewIssuecount

Total number of new issues opened for the article on all locales as of the end of the reporting period.

RatingVerbatim

The number of verbatim created when a user rates the article in the reporting period.

Referrer metrics

OrganicSearchRate

% of page views from organic search (direct referrals only). Organic search comes from Google and Bing and other search engines.

- **Why it's important**: We want to make the most of the traffic we can get from Google and Bing. An increase in this % is a good indicator that the content is generally more discoverable through search. We want the raw volume from search to always grow, but if another channel happens to grow *more* that's OK.
- **How to improve it**: Try SEO best practices.

PaidSearchRate

% of page views from paid search campaigns. Paid Search comes from paid search campaigns on a search engine.

- **Why it's important**: We do not pay for search ads, so this will almost always be close to 0.

DirectTrafficRate

% of page views from direct traffic.

- **Why it's important**: Having links to your topic from within a product provides in-context Help for users.
- **What qualifies as direct traffic**: Direct traffic doesn't have a referring URL, making it difficult to identify the source. Direct traffic sources can be from an email campaign, in-product links (not Web-based), a bookmark, by typing in a URL directly, a server-side redirect, a fwlink or aka.ms link with an outdated target URL, or a bot.
- **How to improve it**: See this article for ways to improve.

MicrosoftTrafficRate

% of page views from any Microsoft domain, including docs.

- **Why it's important**: Having backlinks to your topic helps with SEO.
- **How to improve it**: Do a referrer analysis. Often a particular referrer is to blame for poor performance because they're pointing users to the wrong content to answer their questions. Work with referrers to point to better content or change the content to meet the expectations of that referred traffic. Lastly, consider clearer page titles and intro paragraphs to set expectations. If users can't tell what's in it for them right away, they'll bounce.

ExternalTrafficRate

% of page views from users following links from other domains outside Microsoft.

- **Why it's important**: Having backlinks to your topic helps with SEO.
- **How to improve it**: Because we don't control and can't easily influence these referrers, we may consider ways to meet the expectations they appear to be setting when they link to us: "Looking for X? It's here."

SocialMediaTrafficRate

% of page views from links posted on social media like Facebook and Twitter.

- **Why it's important**: Having backlinks to your topic helps with SEO.

Content Performance report

5/21/2021 • 8 minutes to read

Use this report to get traffic and engagement metrics such as click-through rate (CTR), scroll rate, and bounce rate for individual articles, and track the overall performance for a content set. The report helps us understand how customers engage with our documentation, and whether they are using it as we expect. For example, the purpose of a landing page is to provide navigational links that surface content that helps or interests our customers. If a landing page has a high bounce rate and exit rate, meaning customer's aren't clicking to another topic within the content set, the landing page is not meeting its designed purpose.

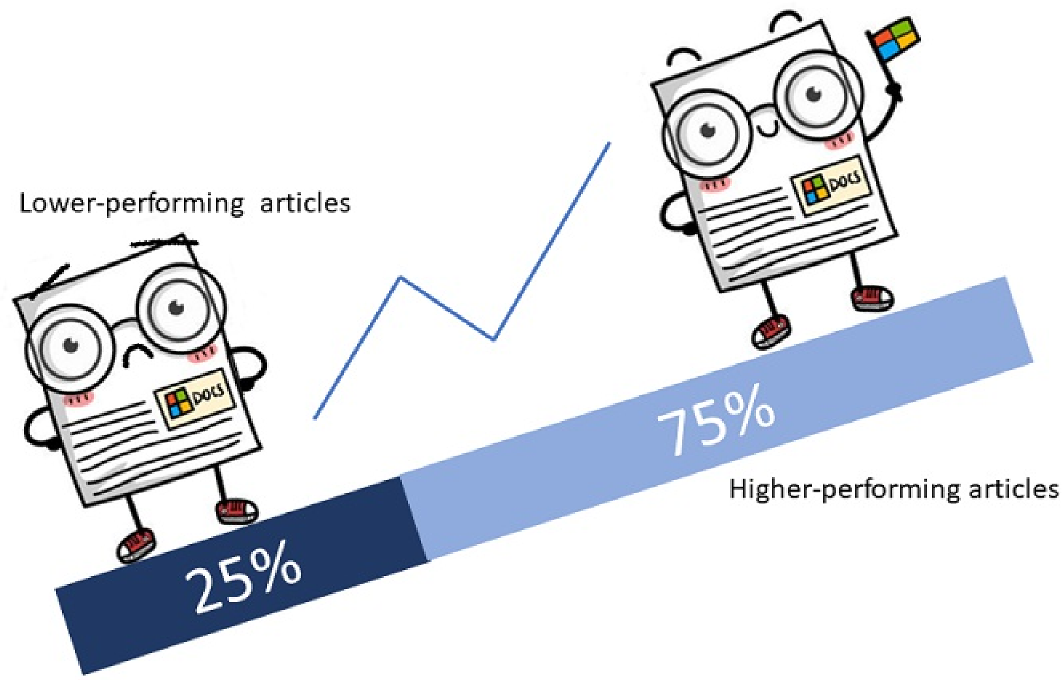
Select the report that maps to your organization. Be sure to bookmark the report by using the aka.ms link to ensure you get the latest version of the report.

- Content & Learning Docs & PnP: https://aka.ms/contentperformancedashboard
- Business Application Group: https://aka.ms/contentperformance-bag
- Microsoft 365: https://aka.ms/contentperformance-m365
- Microsoft Learn: https://aka.ms/mslearnreport

How is engagement measured?

Each topic is given a KPI Score based on the KPI formula assigned to its topic type (ms.topic metadata). The score is then stack ranked with articles of the same topic type to determine which ones are lower performing. For example, the formula for a Quickstart is CopyTryScrollRate + ClickThroughRate - BounceRate, which looks like this when data is filled in (.76 + .18 - .19) = .75. The score is then stack ranked with all the other Quickstarts in the Power BI report.

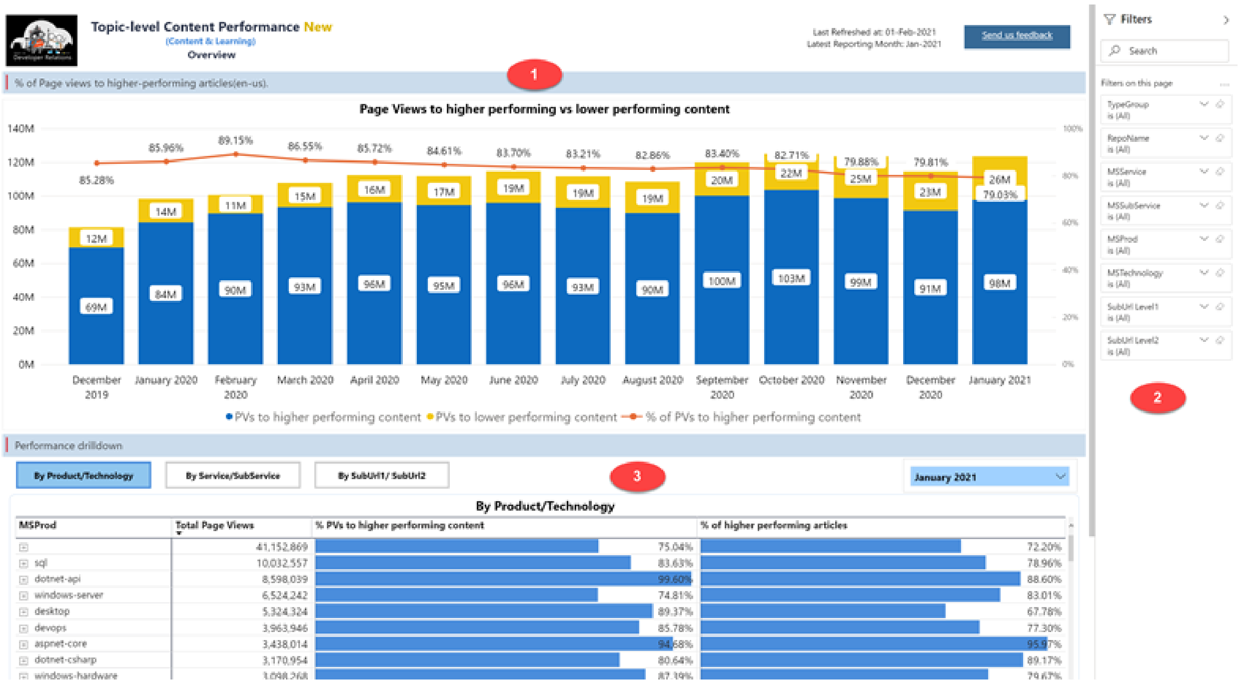
Topics whose primary KPI score ranking falls in the bottom 25% of all articles of that type are considered **lower-performing**. Articles whose performance is in the top 75% are considered **higher-performing**. The report highlights high-traffic lower-performing content so you can target your improvement efforts where they will make the most difference.



This recorded training session covers both the high-level strategy of our reporting and the detailed how-to of using the report and diagnosing performance problems.

Tracking performance for a content set

This page of the report is where you can monitor and track content performance progress across a Topic Type, Service, Product, URL pattern, or Repo. For example, in Fiscal Year 21, the C&L Docs team set an OKR to have 80% of overall page views go to higher performing pages.



Historical performance snapshot (Area 1) - Displays a historical snapshot of the monthly trends of page views of higher performing pages to lower performing pages. Hover over a month in the bar chart to expose a Tooltip that provides the total PVs, % of PVs, MoM PVs, and #articles for each of the low and high-performing groups.

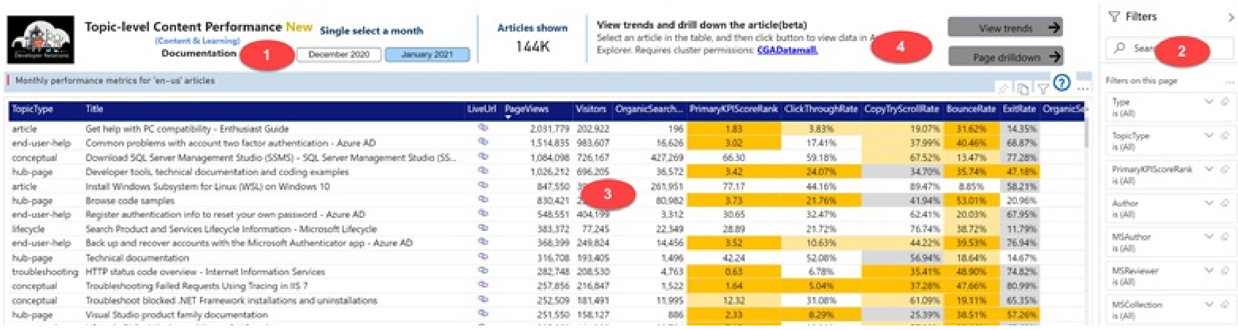
Filter by content set (Area 2) - Use the filters to view the monthly performance data by content type group, repository, service, product, or URL. The low and high-performance ranking is defined by the KPI ranking of the Type Group the topics are a part of and not by the topics within a particular content set.

Drill down of performance by month (Area 3) - Use the **Performance drilldown** section to view the content performance data by Service, Product, or two levels of the URL path for a given month. This view is useful for identifying product or feature areas that might need some investment of effort to improve the customer experience.

Resources and archive (footer) - Download detailed Excel worksheets for previous months (C&L version only), access the Kusto query library, and find key links to the Contributor Guide.

Tracking performance at the page level

Page-level data appears on the **Documentation** and **Reference** pages of the C&L report, and the **Business Applications Group** and **Microsoft 365** pages in their respective reports. Similar data is available in the Learn report on two pages devoted to learning paths.



Two-month filter (Area 1) - Use the buttons to filter by current or previous month. Because of the size of the Docs portfolio, the data is limited to two months.

Filter to your content set (Area 2) - Use the filters to view the monthly data by content type group, repository, service, product, URL, author, and more.

Page-level metrics and KPIScore Rank (Area 3) - Find an article's primary KPI score and primary key metrics and how they perform as compared with articles of the same type. The score rank or metric is highlighted in light yellow if it falls in the bottom 25% and dark yellow if it falls in the bottom 10%. When you sort by highest page views, the list of dark yellow and light yellow results is your list of articles to work on. Focusing on high page view articles ensures you are targeting for impact. The values that appear on a grey background aren't factored into that article's KPI score, but are available for more sleuthing. To identify which metrics a page's KPI score ranking is based on, see Composite Metrics.

View weekly/monthly trends and get Referrer and Click data (Area 4) - Use the **View trends** or **Page drilldown** buttons to view the weekly or monthly trends for a specific article and get drill-down reports in one single view (CTR, Referrers, Verbatim, and GitHub issues).You need Kusto cluster access for both of these dashboards. See how to request access in the next topic and get more details about using these dashboards.

Request access to the Kusto data clusters

To use related dashboards in the Content Performance report, you must request access to a Kusto data cluster. Which cluster(s) you choose depends on your organization. Requests require manual approval from the Customer Growth Analytics (CGA) team, which might take a few days.

Select the links in the following table to submit a MyAccess request. A prepopulated form should appear, but if not, search by the Cluster name, which is the link name.

KUSTO CLUSTER	ROLE	NOTES
CGA Kusto Public (cgadatamall.westus.kusto.windows.net)	Read-only	**Required** for all users. It's the default cluster
CGA KustoFollow DevRel (followercgadataout.westus.kusto.windows.net)	Read-only	**Optional** for DevRel team members. For use if having performance issues with Public cluster
1es CloudMine cluster (https://1es.kusto.windows.net/)	CloudMine-Data reader	**Required** for all users to view GitHub issue information in the Drill down dashboard

IMPORTANT

The CGA Public (cgadatamall) cluster is intended for exploratory queries. It is not scaled for computationally expensive queries or production workloads.The cgadataout cluster is higher capacity, but not infinite capacity. There are no guarantees on performance. Regardless of which cluster you use, we recommend limiting the output using constraints like date ranges and content filters to improve performance and avoid memory limits.

Data scope and update frequency

The Content Performance reports are updated around the 2nd of each month and support metrics for all documentation published to docs.microsoft.com, except for documentation under these paths:

- /answers
- /archive
- /contribute
- /dynamics
- /learn - Removed in May 2020 from current performance dashboard and covered in Learn report
- /legal
- /locale
- /previous-versions
- /search
- /teamblog

Data scope for Docs Partners

The content on docs.microsoft.com that's managed by the Microsoft 365 and Business Application Group were separated from the **Content & Learning** scope and reported in another Power BI report.

ORGANIZATION	URL PATTERNS

ORGANIZATION	URL PATTERNS
Microsoft 365 Contacts: John Martin, Daniel Simpson; Serdar Soysal	/compliance /deployedge (contact: Colleen Williams) /deployoffice /education/windows /internet-explorer /exchange /hololens /lync /microsoft-365 /microsoft-desktop-optimization-pack /microsoft-edge (contact: Reeza Ali) /microsoft-store **/microsoftteams** /office /office365 /OfficeUpdates /OfficeUpdates /onedrive /openspecs (moved to this report Jan 2021) /Project /SchoolDataSync /sharepoint /skypeforbusiness /SharePointMigration /surface /surface-hub /windows/application-management /windows/client-management /windows/configuration /windows/console /windows/deployment /windows/privacy /windows/release-information /windows/security /windows/whats-new /windows/windows-10 /windows-insider /workplace-analytics
BAG (Dynamics 365, Power Platform, and Data Integration)	/ai-builder/ /common-data-model/ /connectors/ /data-integration/ /dynamics365/ /forms-pro/ **/industry/** /powerapps/ /power-automate/ **/power-bi/** /power-platform/ /power-query/ /powerquery-m/ /power-virtual-agents/

> **NOTE**
>
> - docs.microsoft.com/windows contains content for both **IT Pro** (owned by the Microsoft365 team) and **Developers** (owned by Content & Learning team). Only **IT Pro** content is listed. The rest of the pages under /windows are part of the Content & Learning scope.
> - /microsoftteams (**bold**) is available in both the Microsoft 365 and C&L reports.
> - /power-bi and /industry content (**bold**) is in both the C&L and BAG reports.
> - Any repos that contain **office**, **kaizale-docs**, or **stream-docs-pr** are counted in the Microsoft365 scope.

FAQ

What does the count in the PowerBI reports refer to?

The number in the Filter Pane is a default feature in Power BI. For example, on the **Documentation** page, the data spans two months, so this number is the total of both months. If you need total articles in a month, the number is displayed at the top of the page and is also available in the Tooltip for each month on the **Overview** page.

If I make an update to the metadata of my topic, when will it show up in the report?

Any updates made before the last day of each month should show up in the monthly report refresh (the second of the month). The new metadata will be applied to the previous month and current month, but only the current month on the **Overview** page. Metadata updates can take 24-48 hours to get into the system.

Why do I see two records of my page with the same URL, but one that has low page views?

Two records of the same page typically means one record is a redirect. If they have two different TopicKeyIds, the redirect will eventually receive no page views as the cache is cleared and any existing links are updated. If you want to combine the data from the redirected page with the new page, then you need to set the `redirect_document_id` in the redirect.json file to `true`. Find more information for redirecting articles.

Why do I get different page view totals based on whether I filter by Repo, URL, or Service metadata?

The page view data might be different because Service and Product values can span multiple repos and URL patterns. If there's any missing metadata in a repo, then you could get higher page views for the repo or URL than for the Service or Product.

Next steps

By using the content performance dashboard to find high-traffic lower-performing content in your content sets, you can optimize the impact you have when you're working to improve published content. Here's some next steps:

- Troubleshoot lower-performing articles
- Understand how the primary KPI is calculated for each article type
- Review report release notes and known issues

Content Health report (preview) & score

6/16/2021 • 9 minutes to read

IMPORTANT

This article and report is in Preview. Please review the Content & Learning OKRs & key results related to health metrics as well.

In this article, you'll learn what content health is and how use Content Health score to prioritize your investments.

What is Content Health?

Content Health refers to *what we do, as content developers, to ensure quality content*. In response to health signals, we often perform content maintenance. Maintenance is the process of updating content so that it is accurate, free of defects, and up to date.

Content Health is different from content performance, which measures what our customers do, including engagement and feedback.

What is Content Health Score?

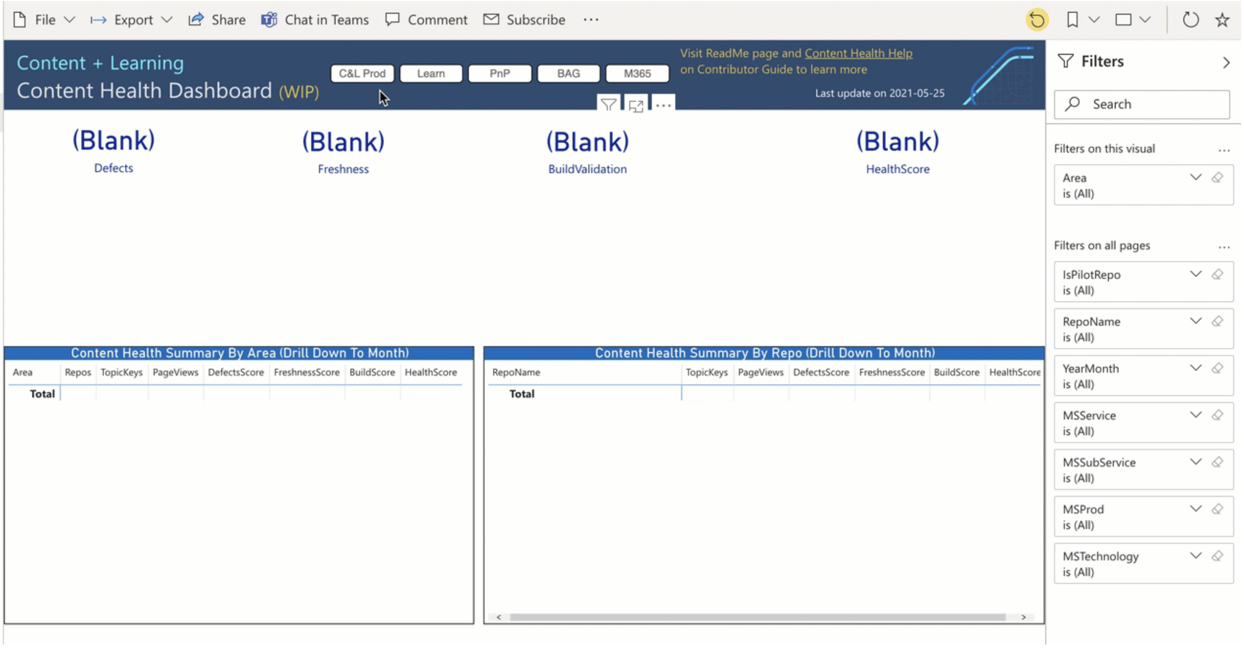
We monitor the health of a set of articles and modules using a **monthly Content Health score**. This score is available in the Content Health report at https://aka.ms/contenthealth.

The **audience** for this report are M1/M2 managers and leadership team, to get an overview of health of their content repos. After further analysis using other reports, writers and managers can figure out resourcing needs to improve metrics as required.

The score is a composite of three subscores of **equal weight**:

- Defect-free page views
- Fresh page views
- Build validations

Content Health report



Report scope and update frequency

The report covers C&L teams (Products, Learn, and P&P), BAG, and M365 (Commercial and Dev).

The report includes only US English (that is, the **en-us** locale) content, and purely non-reference repositories. A caveat is for repositories that have both reference and non-reference content, all files will be included for build validations.

The list of repositories included in this report can be found in this RepoTracker query (Accessible to C&L FTEs only).

The report will be updated at the beginning of every month.

What to do with the score?

> **IMPORTANT**
>
> The Content Health Report and the scores it contains are **not** a report card. They're intended to be used as a signal and to derive insights.

The Content Health score is primarily intended to help managers and content leads understand the health trends in their content spaces. They can use that information to prioritize team projects for content maintenance work, such as addressing build validation messages or doing a freshness pass.

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

How is the score calculated?

This section explains the components of the Content Health score. The overall score is an average of the available composite scores.

Defect Free score

This score calculates the percentage of page views (PVs) to articles/modules with zero defects opened in the public-facing repository. This score is calculated monthly.

The formula is:

```
1 - (monthly PVs to articles or modules with one or more defects created that month / total monthly PVs)
```

How to address defects?

From this score, figure out which articles receive defects labeled as "doc-bug" or which modules receive Learning content quality feedback. Then compare page views to determine impact. This may seem reactive as it requires a customer to make a complaint in the first place. To avoid a negative score, teams should consider proactively auditing their most popular docs and growing a culture of content hygiene. Note that the score does not measure time to close issues or number of issues closed. In other words, closing issues does not improve the defects score. The only thing that improves the score is if customers don't find and report defects in the first place.

Caveats

- If the GitHub issues feedback control is **enabled** for a repo, the defects score will be calculated. The repo must also use standard labels, including “doc-bug” to get an accurate, comparable score.
- If the GitHub issues feedback control is **disabled**, that content won't include a defects score in the composite.
- Learn has a different feedback mechanism, and since it doesn't have as much granularity, the Defect Free score for Learn is often slightly lower than Products or PnP scores. The upcoming unified feedback mechanism will bring similarity across all groupings.

Learn how to see the GitHub issues in your content.

Freshness score

The Freshness score calculates the percent of page views reaching content that has an `ms.date` metadata value within the last 365 days. This age range differs from the previous freshness rubrics.

The formula is:

```
PV count for the articles or modules with a last reviewed date <= 365d / total page monthly PVs
```

How to address freshness?

Find these articles by looking at the date on a live page, or by sorting the LastReviewed date column or the Freshness column in aka.ms/contentperformancedashboard, then compare page views to prioritize the most impactful stale articles. Then refresh these articles using this guidance.

Consider prioritizing your investments if you have lots of older pages. We recommend that you look at how many page views go to an article as compared to others in your content set, as well as adoption/acquisition content.

Caveats

- If `ms.date` is missing for an article/module, the page date is automatically considered to be > 365 days.
- Freshness data collection began in June, 2021, a month before the Content Health Score goes live for all repos. In July, a copy of the data for June will be replicated backward for April and May so that you can see how the reports will work.

Build Validation score

This score represents a *point in time*. The warnings and suggestions we count are those found in the most recent build in that month of the content in a pull request to the `live` branch (most common), or whatever the actual published branch name is (less common). These are builds on our private, internal repositories.

Warnings have twice the weight of **Suggestions** since they are considered a higher severity and they block PRMerger.

The formula is:

```
1 - ((2 * warning count) + suggestion count ) / total number of articles or units
```

How to address build validations?

Review the build report for your pull request to see details about the warnings and suggestions in the repo. For more info, see Docs Build validation overview.

Caveats

- **Changes to validation rulesets can greatly affect the build validation score.** (Remember, it isn't a report card.) Over time, new validation rules are introduced or suggestions are converted to warnings as part of our business. These build validation rule changes are made toward the beginning of the month and the Health score data is calculated at the end of the month. This gap leaves a few weeks each month in which to catch and correct validation messages.
- Builds are never forced by this scoring process to avoid unintended consequences.
- For any repo that contains both reference and conceptual content types, reference content will be included in the build validation score. The calculation does not differentiate by ms.topic type. (Reference only repos are not otherwise included in Content Health scoring).
- Build validation scores are set to stay at or above 0, to avoid negative values that may occur because of an article triggering more than 1 warning or suggestion.

FAQs

What if I want to see a subset of a repository?

You can filter by service or subservice for Docs and Patterns and Practice content. Learn isn't tagged with service/subvice, so those filters aren't available.

The build validation numbers aren't quite what I expected

We can't calculate the number of articles or units built at a service level when we build. Instead, we use the # of articles per service in our calculation, which includes reference content.

What do the spaces in the tables mean?

Sometimes due to unavailability of data, a particular score can not be calculated for a content set. In that case, a space will appear for that content set in the corresponding table. This differentiates from entries which have literal zero, which is when the score can be actually calculated and is found to be zero.

For build validation, why are we multiplying the warning count by 2?

Warnings have a higher severity than suggestions and they block PRMerger, so they are a higher priority to fix.

Who's the intended audience for this report? What can writers get from this report?

The audience for this report are M1/M2 managers and leadership team, to get an overview of health of their content repos. After further analysis using other reports, writers and managers can figure out resourcing needs to improve metrics as required.

Are include files considered in the topic keys?

No, only actual published articles or Learn units are considered as topic keys. Include files are a part of actual published *topics*. TBD: Build validations?

How is freshness score calculated if ms.date is missing?

If ms.date is missing for an article/unit, then the report considers that article as *Not updated in the last 365 days*. This will lower the freshness score for the selected content set.

How will the defect scores be handled when Unified Feedback rolls out?

The content health working group is working closely with the Unified Feedback PM on the Product Management & Design (PM&D) team to be sure that when changes to the feedback system roll out, the Content Health score will take those changes into account.

For defects, is the roll up of multi months by design? How can I see the current month?

You can select the current month in the YearMonth dropdown in the Filter pane, or for some tables in the report, by selecting the month in the YearMonth dropdown at the top right hand corner of that table.

Which scores are collected for public repos and which scores for private repos? What happens if the public-private pairing is complex?

Defect Free score is collected from the public repo, while Freshness and Build Validation scores are from private repos. (TBD: Add more clarification about the public/private pairing, as well as, what happens if there are no public or private designations.)

What happens if I push the same PR several times, and take care of warnings/suggestions that happened the first few times? Would the earlier warnings/suggestions still be counted?

Only the last build scores are considered per repo. If someone else has pushed another PR after your PR was closed, if that was the last PR of the month, the warnings/suggestions counted in that PR will be considered in the build validation score.

Are build validations from master main to live?

Yes, if *live* is the actual published branch, which is true for most repos.

Is this report considering other health-related scores, such as acrolinx, PR mergers, etc?

Current scope of the score is limited to freshness, reported defects, and build status. We are considering the Clarity score part of Acrolinx, or an SEO score, and updates to the score or the dashboard will be evaluated on a quarterly basis.

How do we account for evergreen content in the freshness score?

It depends on the individual team resources and business needs. Teams can either strive for a healthy freshness score by periodically reviewing evergreen content and updating the ms.date, OR they can simply accept a lower score and understand why it is low. Both are reasonable choices.

Any perspective on CSAs (Customer Solution Areas)?

The V1 of this report does not have filters for CSA, however, we are evaluating if the dashboard can have that capacity for Vnext.

Documentation and reference report

3/18/2021 • 5 minutes to read

The Documentation and Reference report provides metrics at the portfolio-level (as opposed to the Content Performance report, which provides metrics on individual articles). It can be found at https://aka.ms/docrefdashboard.

The report has five, high-level filters on the left-hand sidebar (**Area 1** in the image above):

- **By Repo**: Providing KPIs for individual repos, along with a view for all repositories(`AllRepo`) combined.
- **By SubUrl**: Providing KPIs for articles grouped by `SubUrlLevel1` and `SubUrlLevel2`, taken from an article's URL: `https://docs.microsoft.com/locale/` `suburllevel1` `/` `suburllevel2`. For the URL `https://docs.microsoft.com/en-us/azure/security/benchmarks`, for instance, `SubUrlLevel1` would be "azure" and `SubUrlLevel2` would be "security". This report also provides a data view for a `SubUrlLevel1` that includes all the `SubUrlLevel2` paths.
- **By Product**: Providing KPIs for the `Product` and `Technology` metadata elements. It also provides a data view for all `Technology` for a certain `Product`.
- **By Services**: Providing KPIs for the `Service`, and `SubService` metadata elements. It also provides a data view for all `SubService` for a certain `Service`.
- **By Doc Subset**: The Doc Subset Report (preview) shows the most-often-used KPIs at a content set level, either by URL, repo, prod/service, or custom. This is not an article level report. For more information, see Doc Subset Dashboard.

Area 2: Use the filters to view the monthly data by a dimension (`Repo`, `SubUrl`, `Service`, `SubService`, `Product` and `Technology`).

Area 3: Select the reporting month.

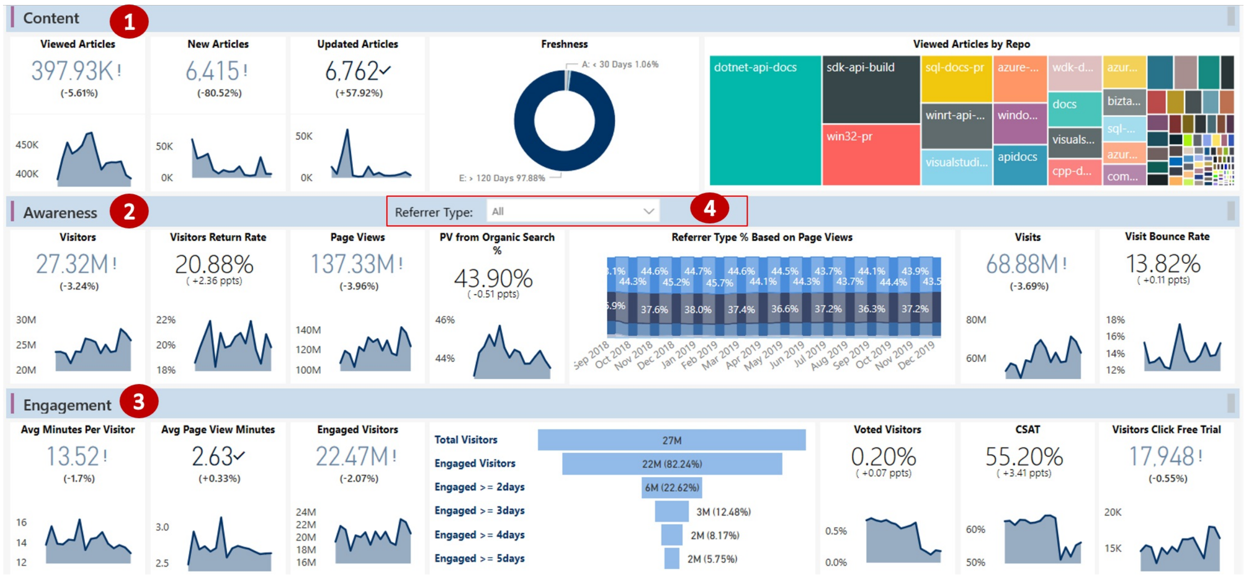
Area 4: you can pivot by `Repo Scope` (**Azure** means only the report the KPI for the repo marked `Product=Azure`, **Azure&NotAzure** means all the repo in repo tracker), and `Area` (**Reference** only reports the KPI for articles which has topic type of `reference`, `managed-reference` and `language-reference`, **Documentation** reports for all other articles, and **Documentation&Reference** is for all articles defined in the repo tracker).

Area 5: you can provide the feedback, clicking the button will launch outlook and create a DevOps task for CIA team triage.

Area 6: you can view the KPI for selected scope, including the **Content**, **Awareness** and **Engagement**.

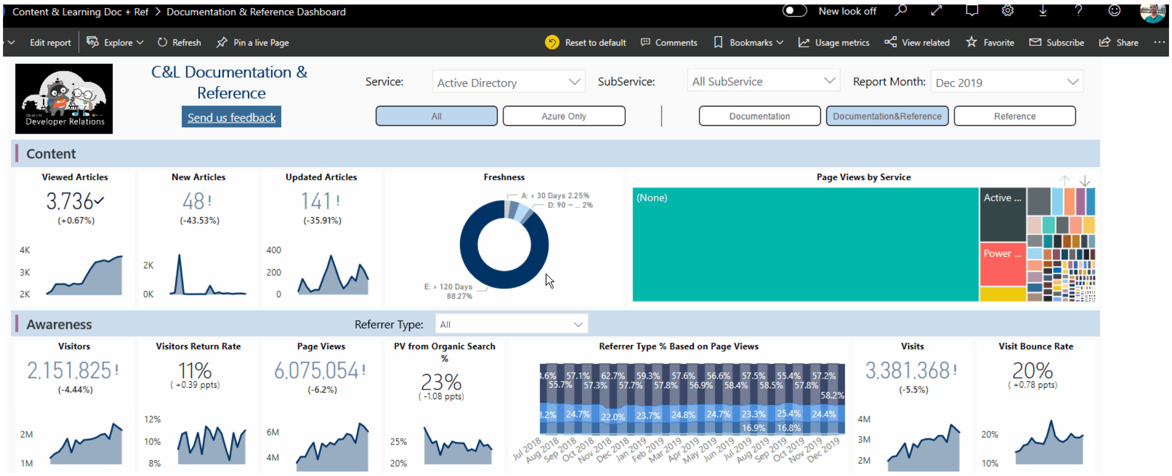
What's the definition for each KPI?

The report provides the Content, Awareness, and Engagement KPIs. For more metric definitions, see Metrics definitions.



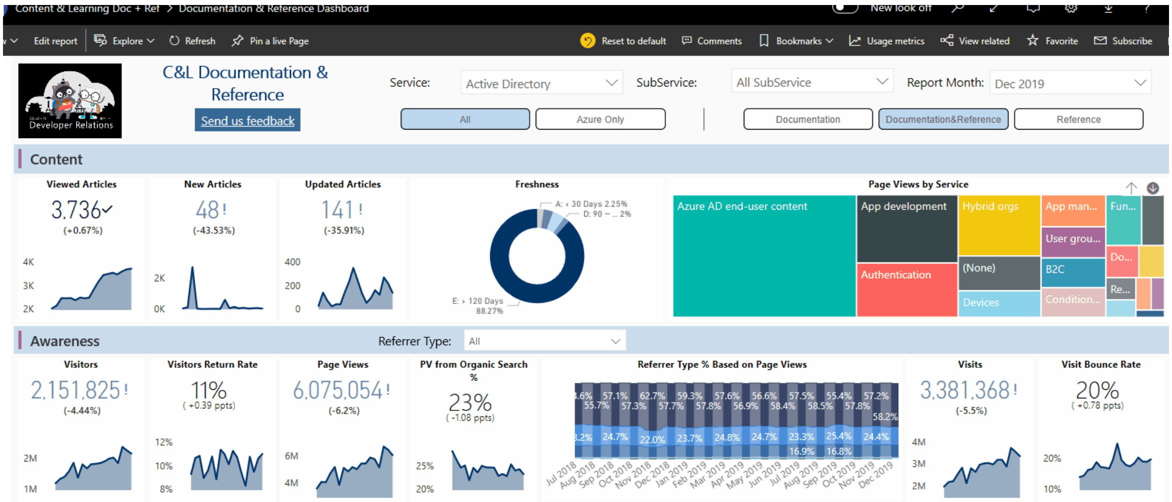
Area 1 (Content)

- **Viewed Articles**: # of articles viewed in the reporting month based on `ContentID`. Non-en-us locales are included with en-us.
- **New Articles**: # of articles first published in the reporting month, based on `ContentID`. Non-en-us locales are included with en-us.
- **Updated Articles**: # of articles updated published in the reporting month, based on `ContentID`, `Last Reviewed Date`. Non-en-us locales are included with en-us.
- **Freshness**: Distribution of the articles based on last updated articles, we count the days gap from `Last Review Date` to the end of reporting month, and categorize them to A,B,C,D,E based on days gap of 0-30, 31-60, 61-90, 91-120, >120 days. Metrics definitions: Freshness.
- **Page View by Dimensions**: A treemap by `PageViews` for level 1 dimension and Level2. For example, in the image below, we can find out the `Active-Directory` is the top 1 `Service` with most Page Views in Dec 2019, and we can Drill down `SubService` and find `Azure Ad end-user content` is the top 1 subservice under `Active-Directory`. The same treemap also shows on other 3 pages.



Area 2 (Awareness)

- **Visitors**: # of unique visitors, based on `VisitorsId`. You can also pivot by `Referrer Type` in **Area 4**. Also see Unique visitors.
- **Visitor Return Rate**: % of visitors in last month returned this month. # of visitors who show in both previous and current reporting month / # of visitors in previous month.
- **Page Views**: # of Page Views, based on `PageViewId`. Also see Metrics definitions: PageViews.
- **PV from Organic Search %**: # of Page View from `Organic Search` / total Page Views. Also see Unique visitors. Also see Metrics definitions: OrganicSearchRate
- **Visits**: # of visit sessions, based on `enrich_session_id`. Metrics definitions: PageViews.
- **Visit Bounce Rate**: we define the visits with only one Page view and `active page view length` < 5 seconds as bounce visits, and we calculate the visit bounce rate as `# of bounce visit/total visits`. Also see Metrics definitions: PageViews.
- **Referrer Type % Based on Page Views**: This is to show the Page View trend for each Referrer type. In the image below, `Microsoft` is the number 1 referrer channel, and `Organic Search` is the second. And there are no big changes in the last 12 months for their Referrer Type %.



Area 3 (Engagement)

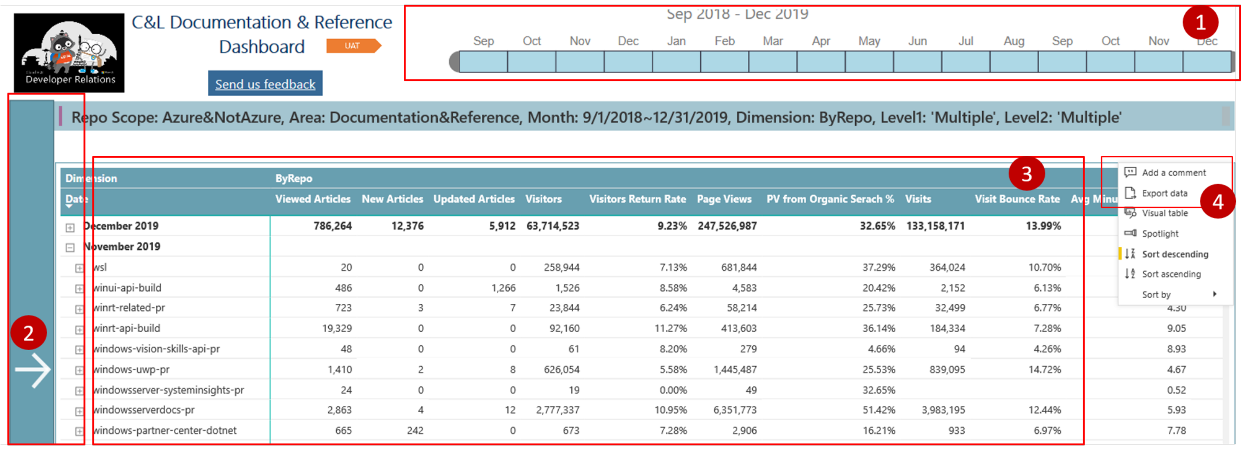
- **Avg Minutes Per Visitors**: The Avg Active Page View Length(minutes) per visitors, we sum the `Active Page View Length` by visitors.
- **Avg Page View Minutes**: The Avg Active Page View Length(minutes) per page views, we sum the `Active Page View Length` by Page View.
- **Engaged Visitors**: We defined the visitors who have valid page view as engaged visitors, the visitors who

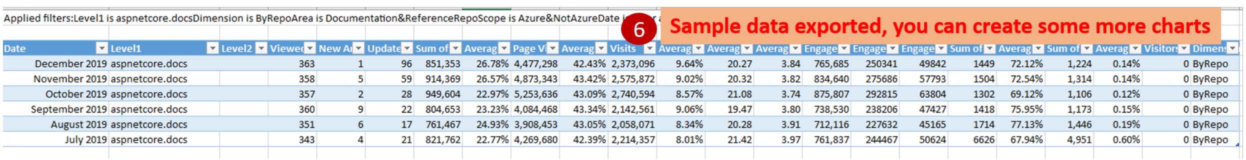
only have bounce visits will not be counted. We call the visitors who come to docs in different `N` days on a certain area as a `N Days Engaged Visitors` in reporting month.

- **Engagement Funnel**: To show the funnel for `Total Visitors`, `Engaged Visitors`, `Engaged Visitors >=2 Days`, `Engaged Visitors >=3 Days`, `Engaged Visitors >=4 Days`, `Engaged Visitors >=5 Days`.
- **Voted Visitors rate**: The # of visitors who rated / total visitors.
- **CSAT**: The # of Helpful ratings / total Ratings
- **Visitors Click Free Trial**: # of visitors who clicked `Azure Free Trial`.

How to export data for customized reporting?

The dashboard provides a way to export the same data we are using the Dashboard from DB, you can choose time range in **Area 1**, and choose **Repo Scope**, **Areas** and **Dimensions** in **Area 2**. After reviewing the data in your selected scope in **Area 3**, you can **Export Data** in **Area 4** after clicking `More Options` (... icon).





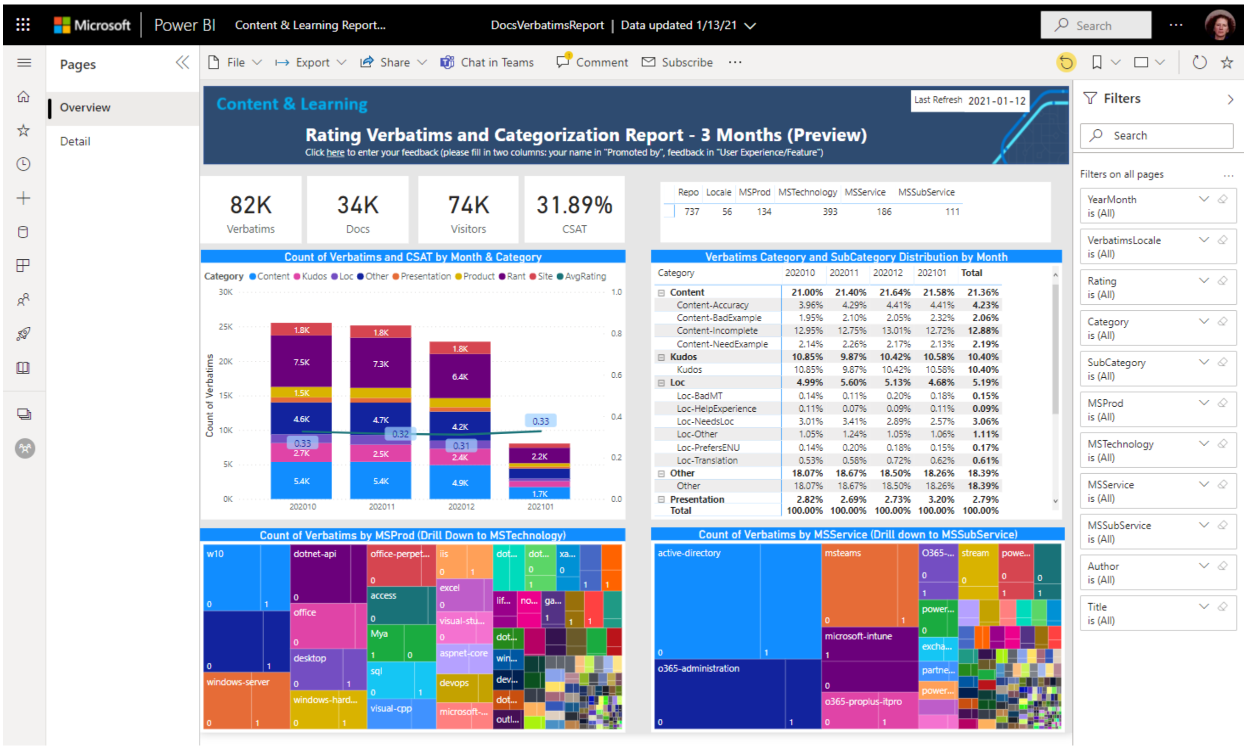
Area 6 provides a sample of data that was exported. You can create additional charts based on the CSV file.

> **WARNING**
>
> If you filter the Dimension field by URL, Product, or Service, and if you do not want to further filter by a Level2 filter (SubURL, Technology, or SubService), then we recommend that you select the "All" filter in Level2 (AllSubUrlLevel2, All Technology, or AllSubService).
>
> If you don't select the "All" filter in Level2, then if you collapse the rows of the table, the roll-up row at the URL, Product, or Service level will be inaccurate. It will sum the "All" Level2 row with all the Level2 rows, effectively doubling the counts.

Verbatims report (Preview)

3/5/2021 • 2 minutes to read

We're rolling out a preview version of a new Verbatims report to replace the previous report. The data in the previous version hadn't updated since September 2020, and we don't own it.



The new Verbatims report https://aka.ms/DocsVerbatimsReport has two pages:

- **Overview** displays summary data for the past three full months, plus the current month to date. It shows total verbatims, number of docs, and the average rating (the 0 or 100 "Is this page helpful?" rating). It segments count of verbatims in a few different ways:
  - By MSProd or MSService.
  - By category and subcategory. These subcategories are assigned using a machine learning text classification tool. They may or may not be accurate. We're planning more training for the tools.
- **Detail** shows the actual comments left by customers via the rating control. You can use these comments to improve your docset. See How to view and act on customer verbatim comments.

Overview page

KPIs

The KPI numbers across the top:

- **Verbatims**: Total number of verbatims in the filtered or unfiltered page.
- **Docs**: Total number of articles with verbatims in the filtered or unfiltered page.
- **Visitors**: Total number of visitors (MAU) with verbatims in the filtered or unfiltered page.
- **CSAT**: The average rating that these articles have received. Note that:
  - Customers can't provide a verbatim without selecting a rating.
  - This average is lower than the overall rating average. Possible explanation: Customers who add verbatims are on average less satisfied.

Column + line chart: "Count of Verbatims & Average Rating by Month & Area"

- Each column in this chart represents the total number of verbatims for a month: three full months, plus the fraction of the current month to date. Month names are year plus month: yyyymm. For example, 202011 is November 2020.
- The legend divides the verbatims by category. The eight areas are a roll-up of 20+ subcategories. The Content category is of most interest to authors.
- The line shows the average rating by month.

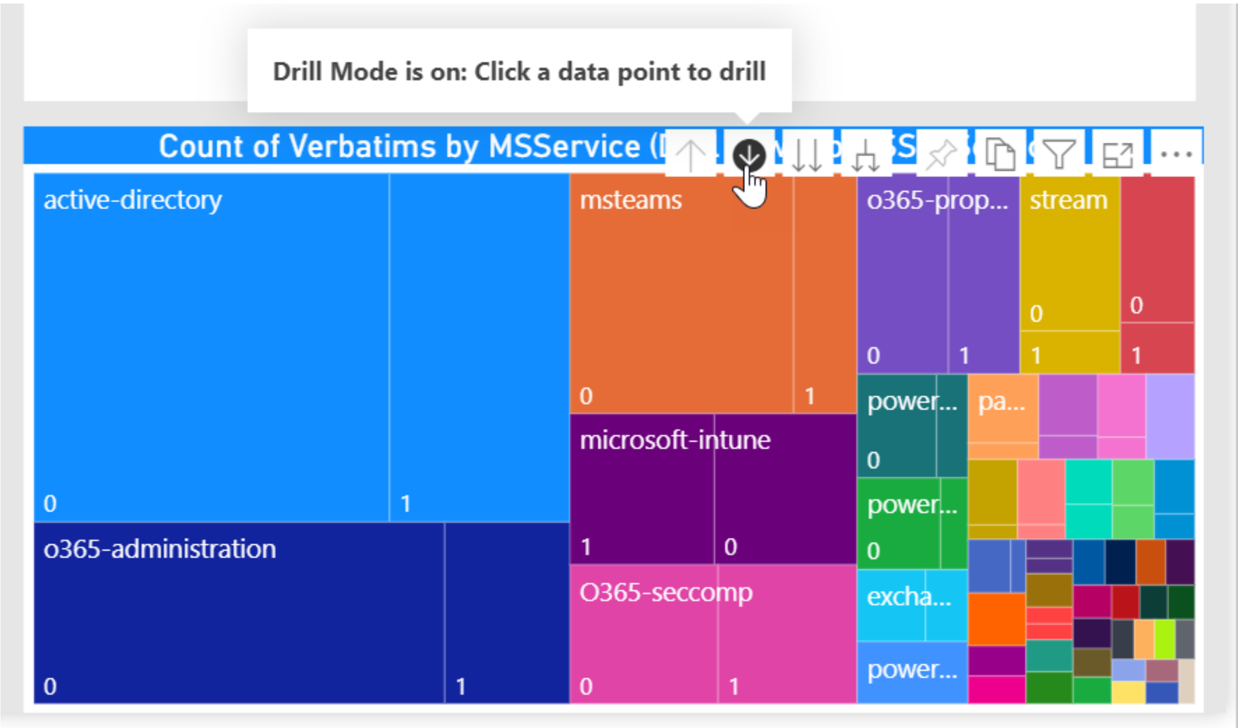
Matrix: "Verbatims Area & Category Distribution by Month"

This matrix shows the same data as the column + line chart, but it lets you drill down the the subcategories in each category.

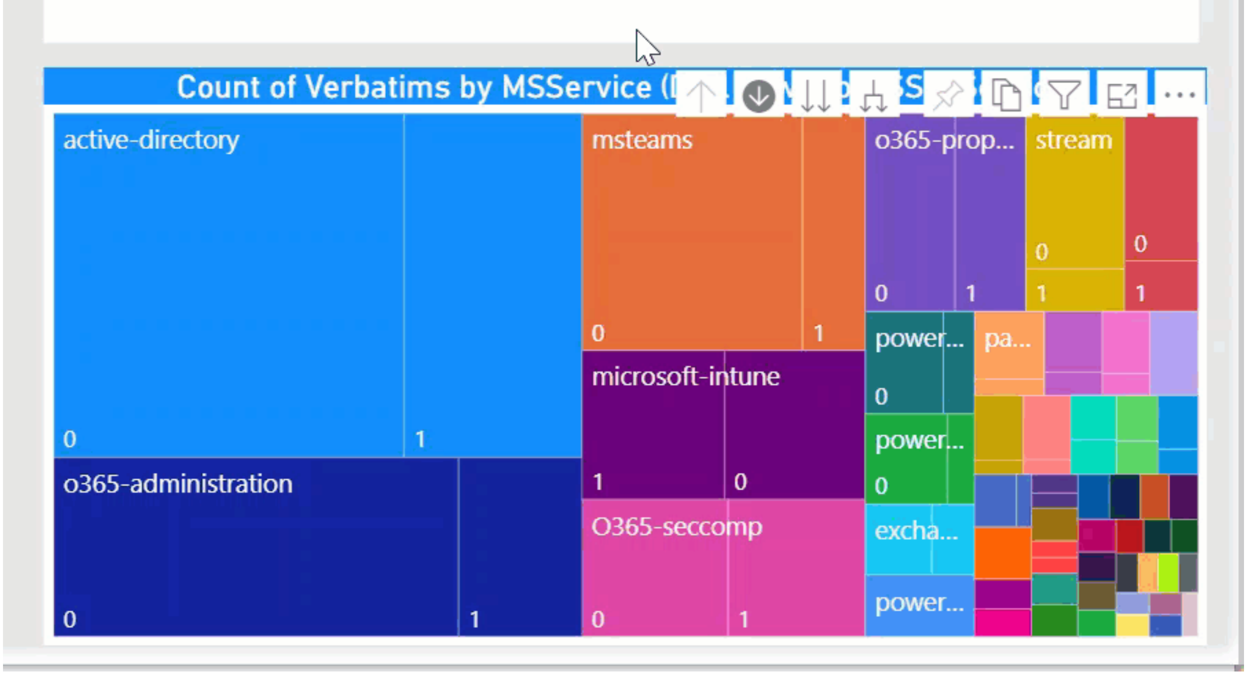
Treemaps for MSProd and MSService

The two treemaps at the bottom break the verbatims out by product on the left or service on the right. You can drill down in either of them to see the technologies/subservices that make up a product/service.

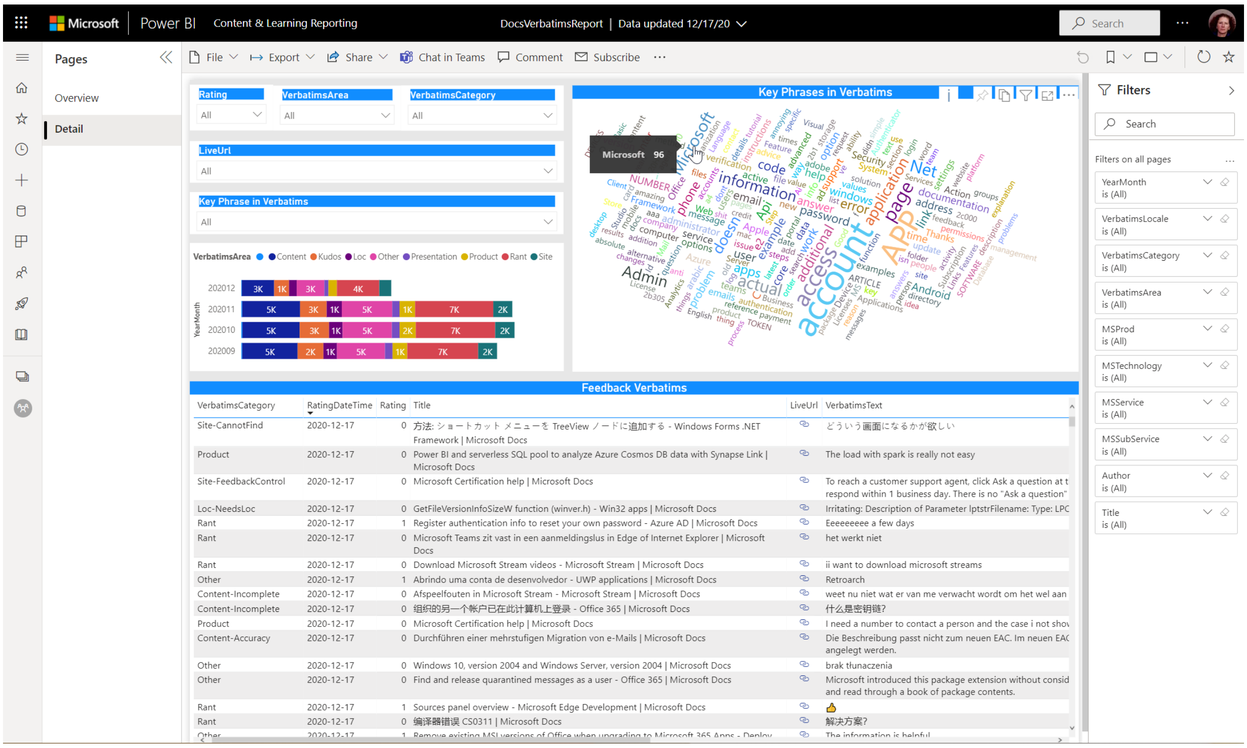
Hover over the treemap and select the down arrow to **turn on Drill down**.



Now select one of the products/services to see its technologies/subservices.



Detail page



Word cloud

The word cloud pulls out key phrases from the verbatims. Size indicates the number of times the word comes up -- the larger, the more frequent. Hover over a word to see the actual number. Selecting a word filters the other visuals on the page, including the table.

Feedback verbatims table

The table lists all the verbatims for the articles that match the filters on the page. If the verbatim isn't in English, scroll right in the table to see the autogenerated English translation.

Next steps

This Verbatims report is in preview. If you have feedback, please enter it in this Verbatims Report Feedback Excel file.

- Learn How to view and act on customer verbatim comments

GitHub issue and pull-request report

3/5/2021 • 3 minutes to read

The GitHub issues and pull-request report provides both portfolio-level metrics - on a repo, department, product, or service - and details on specific Issues.

Pages

- **ReadMe**: A collection of important details and known issues for the current version of the report.
- **GitHubIssueSummary**: A collection of high-level power metrics and visualizations representing the health of GitHub issues within the scope defined by filter settings. This page is currently scoped to public repos within DevRel: Content & Learning that have GitHub issues enabled.
- **GitHubIssueDetails**: Provides issue-level details within the scope defined by filter settings. This page is currently scoped to public repos within DevRel: Content & Learning that have GitHub issues enabled.
- **GitHubPRSummary**: A collection of high-level power metrics and visualizations representing the health of pull-requests within the scope defined by filter settings. This page is currently scoped to all repos within DevRel: Content & Learning.
- **GitHubPRDetails**: Provides PR-level details within the scope defined by filter settings. This page is currently scoped to all repos within DevRel: Content & Learning.
- **WorkArea_**: These pages are testbeds for new visualizations and report functionality.

Filters

Pages: GitHubIssueSummary, GitHubIssueDetails

FILTER	REPORT LOCATION	DESCRIPTION
State	Summary/Details	Open and/or Closed issue states
IssueType	Summary/Details	Type of issue. Possible values are defined by labels applied during the triage process.
RepositoryName	Summary/Details	Name of one or more GitHub repositories
YearMonth	Summary/Details	Years and Months included in the measures.
Priority	Summary/Details	Priority of issues. Possible values are defined by labels applied by the GitHub automation app and are based on page view for the associated topic.
OKRRepo?	Summary/Details	Repositories included in the aggregate Content Health KR measure surfaced in the Monthly business review.
OrganizationLogin	Summary/Details	GitHub Organization (for example, MicrosoftDocs, DotNet)

FILTER	REPORT LOCATION	DESCRIPTION
MSProd	Summary/Details	ms.product value of the topic the issue was filed against
MSTechnology	Summary/Details	ms.technology value of the topic the issue was filed against
MSService	Summary/Details	ms.service value of the topic the issue was filed against
MSSubService	Summary/Details	ms.subservice value of the topic the issue was filed against
AssigneeOrg	Summary/Details	Assigned issues by Organization.
AssigneeDepartment	Summary/Details	Assigned Issues by Department
AssigneeManagerAlias	Summary/Details	Assigned issues by manager assignee(s) report to
AssigneeMSAlias	Summary/Details	Assigned issues by MS alias
GitAlias	Summary/Details	Assigned issues by Git alias
ClosedMonth	Details	Issues closed in each month(s)
SubmitterType	Details	Internal or External (non-MSFT) submitter
Submitter	Details	Individual that created the issue
MSFTSubmitterOrg	Details	MSFT submissions by Organization.
MSFTSubmitterDepartment	Details	MSFT submissions by Department
MSFTSubmitterManager	Details	MSFT manager submitter(s) report to
MSFTSubmitterAlias	Details	Submissions by MS alias
Labels	Details	By issue label(s)
ContentId	Details	By the ContentId defined in a topic's metadata

Pages: GitHubPRSummary, GitHubPRDetails

FILTER	LOCATION	DESCRIPTION
YearMonth	Summary/Details	Years and Months included in the measures.

FILTER	LOCATION	DESCRIPTION
OKRRepo?	Summary/Details	Repositories included in the aggregate Content Health KR measure surfaced in the Monthly business review.
OrganizationLogin	Summary/Details	GitHub Organization (for example, MicrosoftDocs, DotNet)
RepositoryName	Summary/Details	Name of one or more GitHub repositories
MSProd	Summary/Details	ms.product value of the topic the PR was opened against
MSTechnology	Summary/Details	ms.technology value of the topic the PR was opened against
MSService	Summary/Details	ms.service value of the topic the PR was opened against
MSSubService	Summary/Details	ms.subservice value of the topic the PR was opened against
AssigneeOrg	Summary/Details	Assigned pull-request by Organization.
AssigneeDepartment	Summary/Details	Assigned pull-request by Department
AssigneeManagerAlias	Summary/Details	Assigned pull-request by manager assignee(s) report to
AssigneeMSAlias	Summary/Details	Assigned pull-request by MS alias
GitAlias	Summary/Details	Assigned pull-request by Git alias
ClosedMonth	Details	Issues closed in each month(s)
OpenDays	Details	How long the PR has been, or was, open
SubmitterType	Details	Internal or External (non-MSFT) submitter
SubmitterGitAlias	Details	Submission(s) by GitHub alias
SubmitterDepartment	Details	MSFT submissions by Department
SubmitterManagerAlias	Details	MSFT manager submitter(s) report to
SubmitterMSAlias	Details	MSFT submissions by alias

Providing feedback

Similar to other released content reports, feedback can be provided via the Content Data Advocacy Group dashboard.

Next steps

- Learn the GitHub issues process for Azure authors.
- Receive GitHub issue and public pull request reports.

Get page-level metrics in your browser: Microsoft Docs Metrics

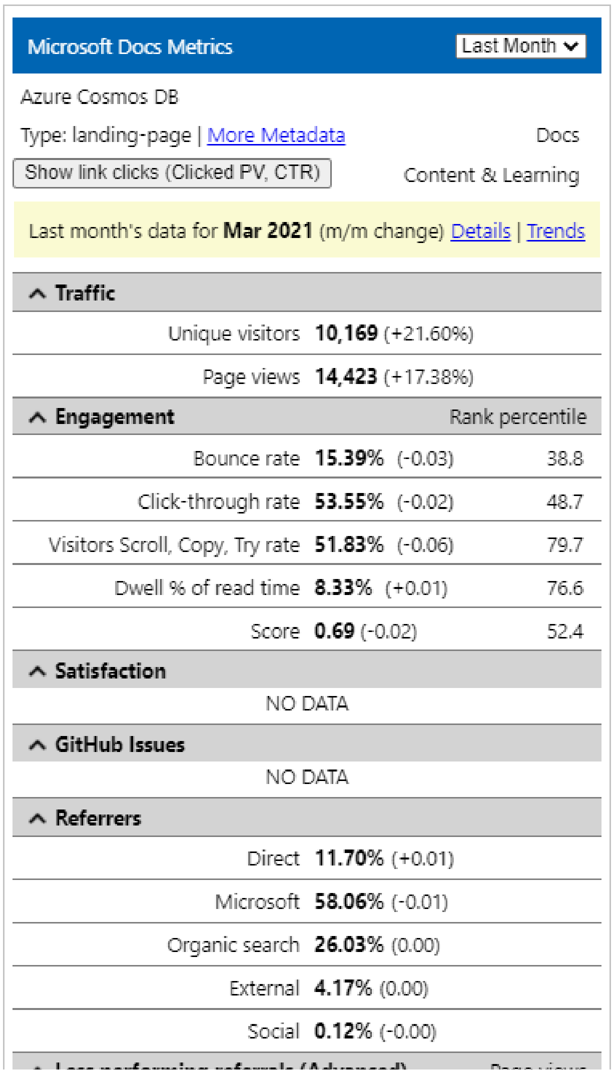
5/24/2021 • 12 minutes to read

The Microsoft Docs Metrics (MDM) tool that allows you to get high-level metrics for a page on docs.microsoft.com right from your browser. It's convenient and useful when you want easy access to traffic, engagement, and referrer data, or you want to see verbatim comments for a specific page.

MDM is a replacement for SkyRay, support for which ended in Feb 2021.

> **WARNING**
>
> EARLIER VERSIONS of this extension stopped working in late April due to CGA changes. Update as soon as possible. **Last release: 2021-04-22** (release notes & past versions).

For content on approved tech sites, the MDM tool opens directly in Azure Data Explorer instead. Those sites include: `azure.microsoft.com` , `blogs.msdn.microsoft.com`, `blogs.technet.microsoft.com`, `cloudblogs.microsoft.com` , `channel9.msdn.com` , `code.msdn.microsoft.com` , `code.visualstudio.com` , `devblogs.microsoft.com` , `developer.microsoft.com`, `gallery.technet.microsoft.com`, `msdn.microsoft.com` , `social.msdn.microsoft.com`, `social.technet.microsoft.com`, `techcommunity.microsoft.com`, `technet.microsoft.com`, `techprofile.microsoft.com` , `www.azure.cn`, `docs.azure.cn` , and `www.microsoft.com`



^ Less performing referrals (Advanced)		Page views
Direct Traffic		1,631
Microsoft	portal.azure.com/	862
Microsoft	docs.microsoft.com/en-us/learn/modules/azure-database-fundamentals/summary	587
Microsoft	docs.microsoft.com/en-us/azure/architecture/data-guide/technology-choices/data-storage	142
Microsoft	docs.microsoft.com/en-us/azure/cosmos-db/graph-introduction	82

^ SEO

Page views from search **3,754** (+18.09%)

Google Keywords (Top 5)	Page views
cosmos db tutorial	291
azure cosmos db tutorial	222
cosmos db	175
azure cosmos db	157
cosmos db documentation	120

^ Operations [View GitHub Content]

Freshness	**E: >120d**
Last Reviewed	**2019-09-10** (590d)
Last Commit	**2021-04-17** (5d)
Lines Changed	**11**

^ Resources

@microsoft.com | log out | View my topics | Privacy

See also: How to use this tool | C&L reports | Send feedback

3.0.1

Install and upgrade version

MDM on Edge is GA (offering auto-updates), but remains unsigned in beta on Chrome. We expect to have GA for Chrome later this year.

Use the following browser-specific steps to remove the preview (beta) extension and install the GA version.

For Microsoft Edge

1. Uninstall any older beta versions of this tool before continuing, if applicable.

2. Get the latest extension from version: 3.0.1

> **NOTE**
>
> - The **Extensions lab** page can only be accessed if you're signed in with your Microsoft alias and connected to Microsoft's corporate network.
> - Scroll down to locate the MDM extension because the **Extensions lab** page is not searchable.

3. Enable auto update from extensions lab:

   a. In your browser, open the extension management tab.

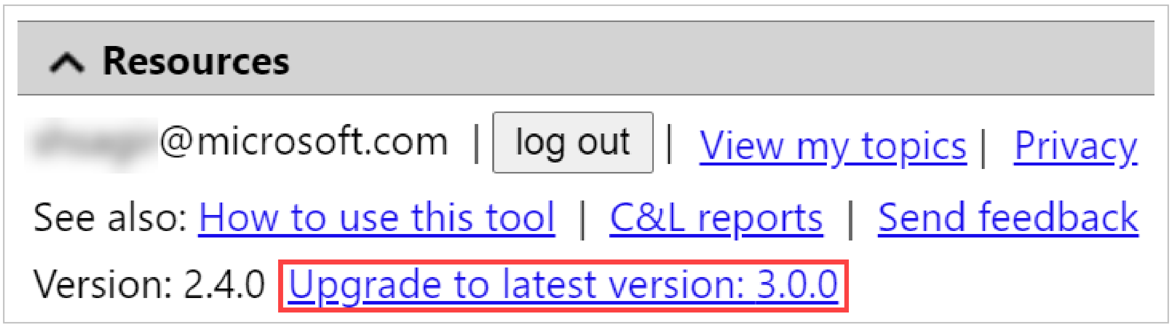
   b. Turn on **Auto install extensions from Extension Lab** to automatically get newer versions of the extension when they are released.

For Google Chrome

WARNING

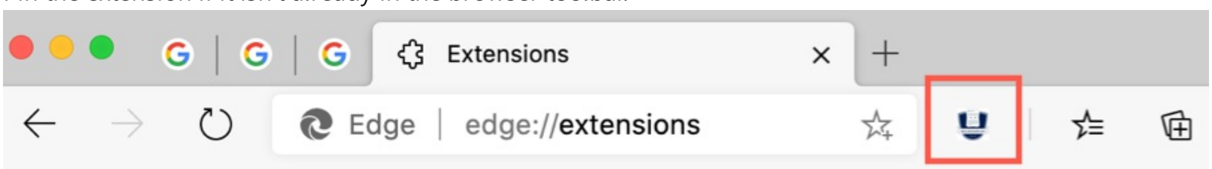
The MDM extension does not auto-update on Chrome. Chrome users are responsible for keeping MDM up to date. The latest version is shown at the bottom of the tool window in your browser at all times.

1. Uninstall any older beta versions of this tool before continuing, if applicable.
2. Download and unzip the MDM extension zip file from version: 3.0.1
3. In your browser, open the extension management tab.
4. In the Extensions tab:
   a. Turn on **Developer mode** in Chrome.
   b. Select **Load unpacked**.
   c. In the dialog box, select the folder for the unzipped extension.
   d. 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, make sure to **allow pop-ups** in (Edge or in Chrome) from `https://skyray-api-public.azurefd.net:443` so that you can authenticate.
2. Pin the extension if it isn't already in the browser toolbar.



3. Open a page on https://docs.microsoft.com.
4. Click the **Microsoft Doc Metrics** extension icon in the top-right corner. The extension opens on the right side of the page.

WARNING

TROUBLESHOOTING SIGN ONS

If you aren't automatically signed in, try the following fixes:

- In the Login window, enter your Microsoft domain account.
- If the window doesn't appear, check your pop-up blocker.
- If you are a **Google Chrome users** and signed into your browser, go to **Chrome** > **Preferences** in the menu. In the **You and Google** section, log out.

Permissions

To view MDM metrics in the Azure Data Explorer dashboard, you will need the following permissions.

1. If you are a vendor, join the security group, Content Learning Dashboard User Group.
2. Request **read permissions** for the CGA Kusto Public cluster (cgadatamall.westus.kusto.windows.net). You can also initiate the request from the dashboard by clicking **cga kusto public** (top-right corner).
3. **DevRel org only**: Also request **read permissions** for the CGA KustoFollow DevRel cluster (followercgadataout.westus.kusto.windows.net).

To access the source file in GitHub, you must:

Enable your GitHub account for access to our private internal organization, MicrosoftDocs.

> **WARNING**
> 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.

Time range choices

Use the drop-down in the upper right of the tool to choose the time period for the data.

> **TIP**
> It can take up to a week for the previous month's data to become available. For example, you might not see December data until after January 5th.

OPTION	DESCRIPTION
Last month	The latest complete month of data
Last week	The latest complete week of data

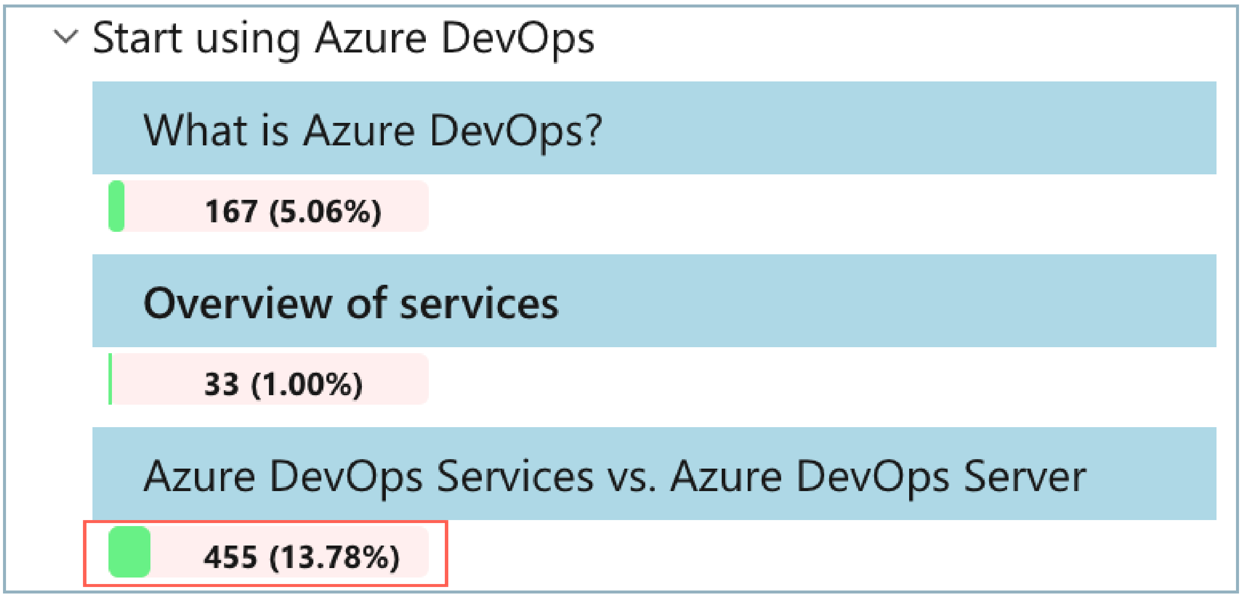
Outgoing click counts

To see what outgoing links were clicked (click-through), select the button **Show link clicks (Clicked PV, CTR)**. When you select this button, all the links clicked during the selected time range appear on the page.

The data appears on the page with this syntax `Clicked PV (CTR %)`:

VALUE	DESCRIPTION
Clicked PV	This value shows the # of page views with outgoing traffic to the URL behind the link, and **not the actual clicks on the link**. If you have 3 links on that page pointing to the same URL, their counts will be aggregated.
CTR	The percentage of page views that visited the linked URL (click-through rate).

You'll also see how many page views a URL got by the visual bar.



TIP

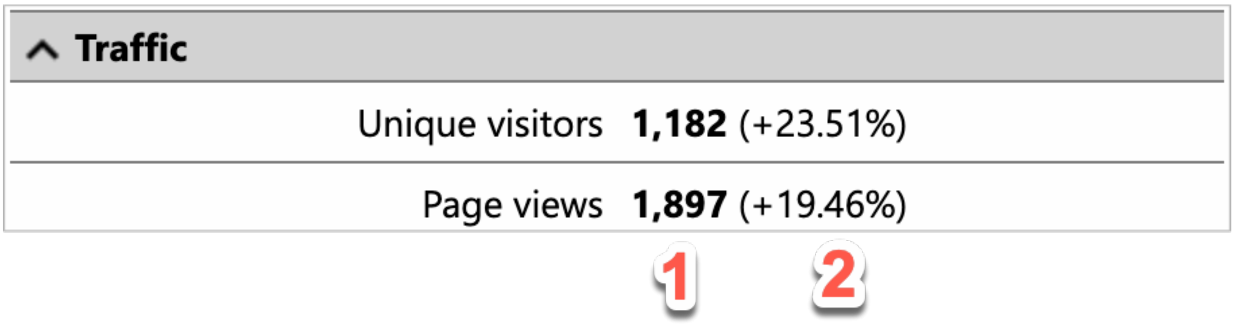
If you can't see your page well, click the MDM icon in the toolbar again to hide the extension while keeping the link counts.

Traffic metrics

This section provides page traffic information for the selected period.

For each metric, you can review:

1. The overall volume of **page views** and **unique visitors**
2. The **growth rate** (month-over-month or week-over-week)



Engagement metrics

This section provides content metrics on how readers engaged with this page. Use these metrics to understand:

VALUE	GROUP	DESCRIPTION	RANK PERCENTILE
Bounce rate	All	Left too quickly & bounced in <5 seconds	The lower, the better
Click-through rate	All	Clicked through to another page on our site	The higher, the better
Visitors Scroll, Copy, Try rate	Docs	Engaged (copy, try, scroll) with page content	The higher, the better

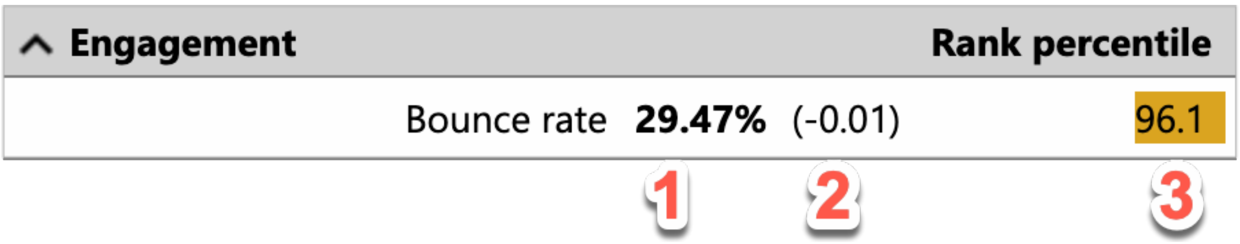
VALUE	GROUP	DESCRIPTION	RANK PERCENTILE
Dwell % of read time	All	Dwelt on the page for a while	The higher, the better
Monthly completion rate	Learn	Completed the learning path	The higher, the better
Score	Docs	Roll-up of the engagement metrics into a single number	The higher, the better

In the **Rank percentile** column, you can see the relative performance of this page against other pages of the same type group across all of https://docs.microsoft.com. For example, all tutorials use the same primary metrics. If you see multiple ranks, it may mean that this page appears for two groups, such as in a C&L doc set *and* a BAG doc set.

The **Score** value is a roll-up of the primary engagement metrics for a given page type (see formulas) into a single number. Identifying pages that are performing lower than others can help you spot potential issues. and troubleshoot your page's performance.

TIP

If **Bounce** is high, check the **Underperforming referrers** and run the **Referral Analysis query**. See the Referrer section.



As shown in the graphic, the data is presented as:

1. Rate of PVs exhibiting this behavior
2. The growth rate month-over-month or week-over-week)
3. The percentile ranking of this page against other pages like it, with highlighting wherever the article is performing significantly worse that its peers in this area.

Satisfaction (CSAT) & Verbatim feedback

Not applicable to **Learn** content.

Customer satisfaction (**CSAT**), is measured by the % of responses with helpful votes divided by total votes. This section is particularly helpful when the page has received many votes. If you only get a few, it might not be as insightful. It shows "NO DATA" if there aren't any CSAT responses on the page.

You can also select the **Details** link to view an Azure Data Explorer dashboard filtered to the article you were looking at, and the time period you had selected: one week or one month. You can update the dates in the query to see more verbatims.

Use the verbatims 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**.

Change log

3.0.1

Version 3.0.1, released on **2021-04-22**, and contains the following major fixes:

- Fix a potential security issue for token exposure.

- Update backend API service endpoint which is secured by PME tenant.
- Remove '(beta)' characters for new release version.
- Add privacy statement url in extension UI.
- Signed extension for Edge version (not signed yet for Chrome).

2.202102

Version 2.202102, released on **2021-02-28**, and contains the following major fixes:

- Switched to content performance V3 data to align the ranking and latest engagement metrics (**CopyTryScroll rate**) used in the Content Performance Dashboard
- Added support for *docs.microsoft.com/archive* (*Other Docs Page*) not in the scope of the Content Performance Dashboard for all metrics except ranking
- Added support for Tech sites to view common page KPIs in a new ADX dashboard tab
- Removed drilldown links to **RatingVerbatim** (under Satisfaction) and **ReferralDrilldown** (under Referrers) links as the data is available in the dashboard opened using the **Details** link
- Typos and label clarifications fixed

2.202012

Version 2.202012, released on **2020-12-10**, and contains the following major fixes:

- Improved the outgoing links match % with some work arounds the dynamic links on tabs and cards links
- Improved the least performing referral results, only add the enrich_url_page_query_string to Direct Traffic referrals
- Added notification feature to inform users on MDM if new version available, and users can also find the local version and latest version on server, follow the links to install the latest version
- Replaced Trend and details in MDM to match content perf dashboard.
- Query data from Kusto when hitting empty cache, this is to fix the Google search data shows empty when it's refreshed on the 4rd day while other content performance data ready on 2nd day
- Improved the CSS style when embedding the MDM to work around the TOC overlayer problem
- Typos and label clarifications fixed

Note: **Version 1.9** will stop working in late December due to CGA changes. Updates as soon as possible.

1.9

Version 1.9, released on **2020-10-22**, and contains unspecified UAT fixes.

Contributor Insights

4/19/2021 • 5 minutes to read

The Contributor Insights Dashboard helps you understand the types of contributors who are publishing their contributions to the docs.microsoft.com platform. To see the Contributor Insights Dashboard, visit https://aka.ms/clcontributordash.

Our active community across Microsoft is a diverse group of contributors working together across the company and uses input coming from outside Microsoft as well. This dashboard categorizes our community as members into one of three groups:

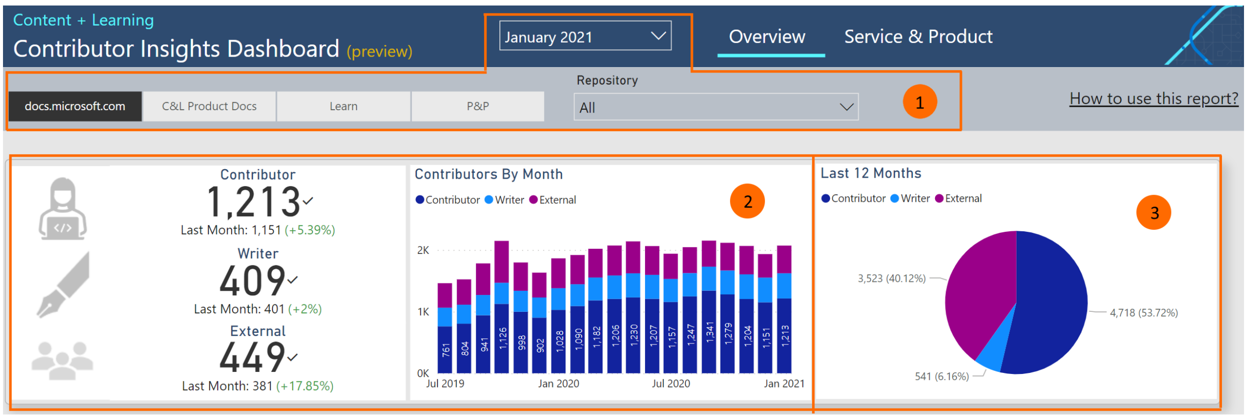
- **Writer** - a community member for whom content development is their primary job responsibility (including vendors and publishing staff)
- **Part Time Contributor** - a community member for whom content development is not their main job role (for example product team Program Managers, Engineers, etc.; includes vendors)
- **External** - a community member who is outside Microsoft, possibly a customer, a partner, or a developer

> **IMPORTANT**
> A **Contributor** is defined as anyone who has authored a commit in the live branch of a docs.microsoft.com repository, e.g. Contributors have had their work *published*.

Understand the overview page

The Overview page shows contributor information for all content repositories for the most recent completed month. You can change the month of the report, or use the filters to narrow down contributor information to only certain repositories.

For more information on the areas highlighted in this image, read the corresponding sections below.



Choose your scope - Area 1

The overview page offers a few selections for how to scope the rest of the page.

Month - Change the month of the report to show contributor participation in previous months.

> **IMPORTANT**
>
> A **Contributor** is counted in a given month if their commits joined the live branch of a docs.microsoft.com repository in that month, regardless of when the commits were authored.

Repository selection - use the buttons or the Repository menu to set the scope of the report. You can either use the buttons to select groups of repositories, or the item selection to choose a single repo.

The four buttons filter the report to a view of groups of repositories.

- `docs.microsoft.com` - Choose to see information for all repositories for docs.microsoft.com
- `C&L Product Docs` - Choose to see information for only technical docs for C&L Products
- `Learn` - Choose to see information for only Learn repositories
- `P&P` - Choose to see information for only P&P Repositories

Repositories are categorized into one of these groups using Repo Tracker, which categorizes and tracks active repositories. Please contact Ryan Thompson with questions about repo tracker at this time.

Mirrored Repositories - For repos in a private-public pair, *only the private repo is listed* in the drop-down list of repos. The public repo is not listed because content is not published from public repositories. Commits from external users are mirrored to the private repo before they're published, so they appear in the dashboard when the private repo is selected.

> **TIP**
>
> The `Repository` selection is affected by the buttons, so if you're looking for a repository but can't find it, try changing your selection back to `docs.microsoft.com`.

Contributor count by Month - Area 2

The contributor counts shows the number of contributors of each type for the selected filter.

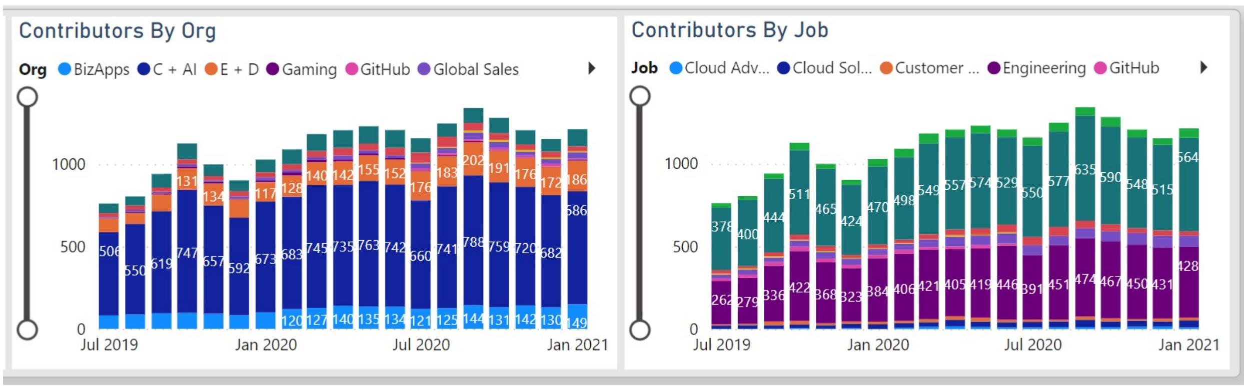
Contributor count over 12 months - Area 3

The contributor engagement window shows the count of contributors that have published content on docs.microsoft.com in the previous 12 months.

In this screenshot, 1900 part-time contributors contributed in only one month of the previous 12 to all repositories, and around 300 full-time contributors have contributed in 11 or 12 months in the past 12.

Digging deeper into contributors

The Contributor Insights dashboard answers basic profiling questions for the selected scope over the past 12 months.



Contributors by job

The contributor insights dashboard categorizes by contributors using their job title in their Microsoft dashboard.

The categories it shows are: - Product PM - Engineer - IT & Service Operations - Cloud Solution Architects & Sales & Marketing - Support - Customer Engineer - Cloud Advocates - Others

Contributors by organization

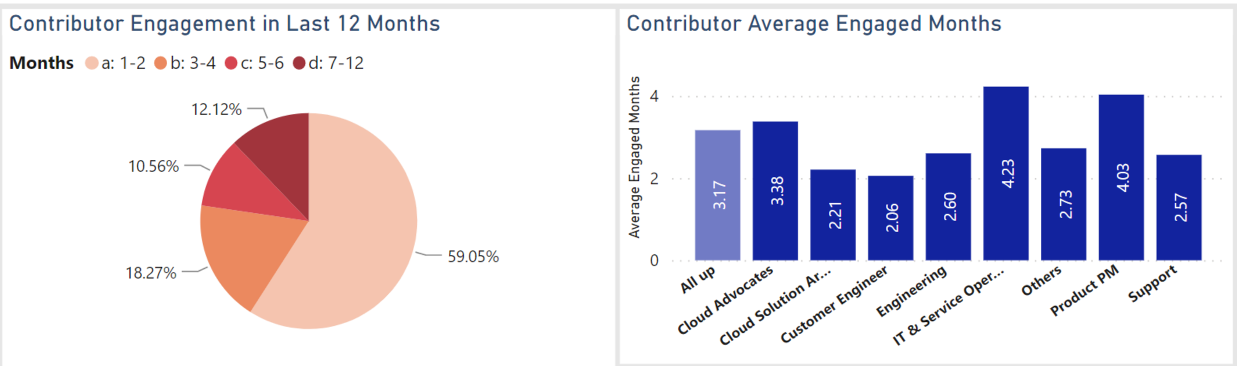
The contributor insights dashboard shows a view of which organization within Microsoft a contributor belongs to. We ask this question by asking which senior leader within Microsoft an individual contributor reports up to.

This table shows the designation of organization and who the contributor reports up to for this categorization:

ORGANIZATION	CONTRIBUTOR REPORTS UP TO
C + AI	Scott Guthrie
E + D	Rajesh Ja
WW Commercial Business	Judson Althoff
Global Sales	Jean-Philippe Courtois
Gaming	Phil Spencer
BizApps	James Phillips
Linkedin	Ryan Rolansky
Github	Nat Friedman

How often are part-time contributors contributing

Two visualizations exist to tell the story about how often part time contributors are contributing. The first looks how many of the past 12 months each Part Time contributor showed up in. Since they do not contribute every month, this view shows how active our community is. The second shows the average number of months for all part-timers, but then splits out that average to isolate the activity of the different jobs.



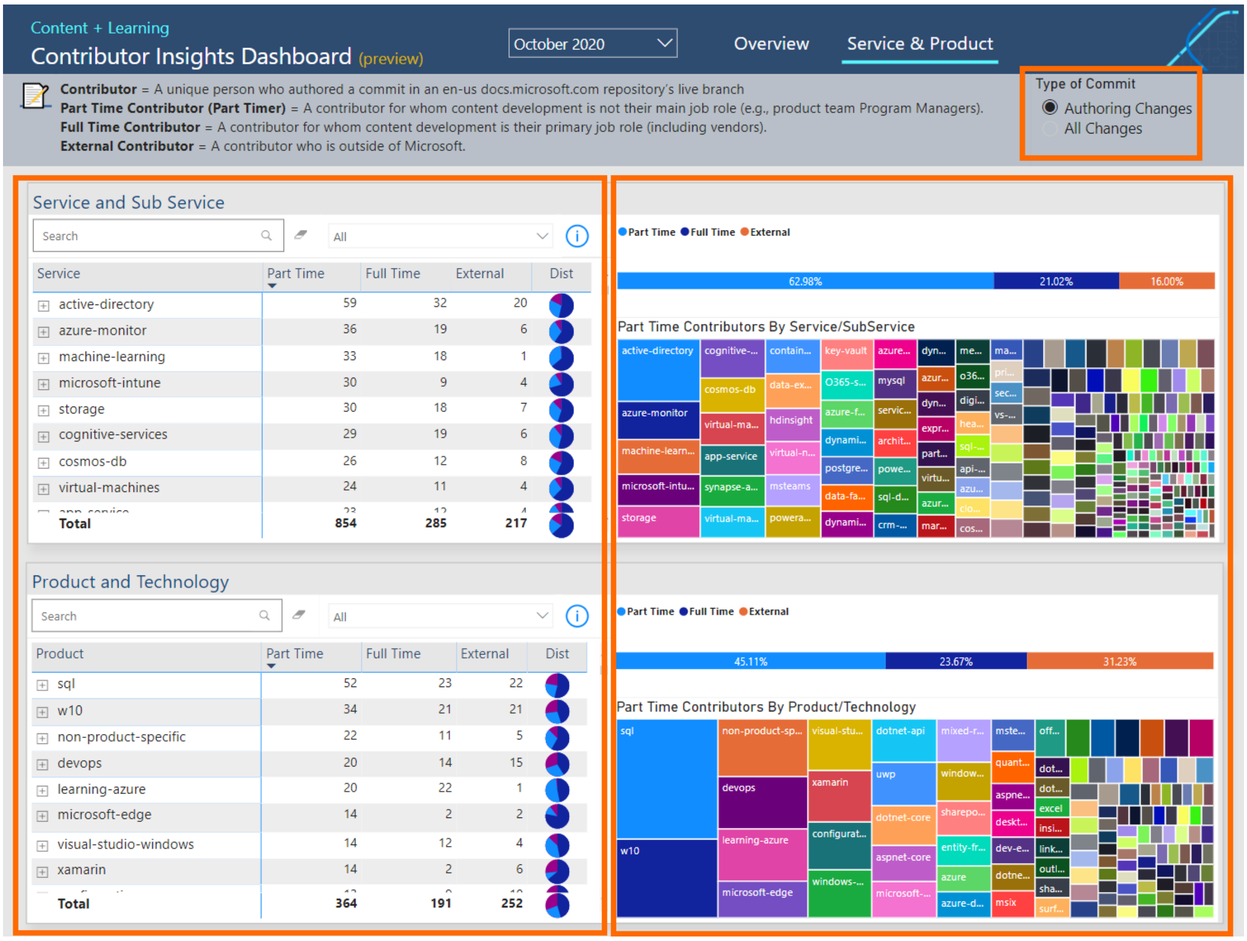
How many contributors this month also contributed last month

A final visualization asks "of the contributors in the current month, how many also contributed last month?" to understand our month over month *return rate* for contributors. It shows both the number of part-timers who returned from the previous month, and the % value of Part Time contributors who returned.

Contributors by product or service

On the Service & Product tab of the report, the contributor insights dashboard shows the count of contributors that committed to individual service and product tags.

> **IMPORTANT**
>
> Use the **Type of Commit** control to toggle between a view that shows *Authoring Changes* only, or *All Changes* which also includes merge commits. The default is *Authoring Changes*.

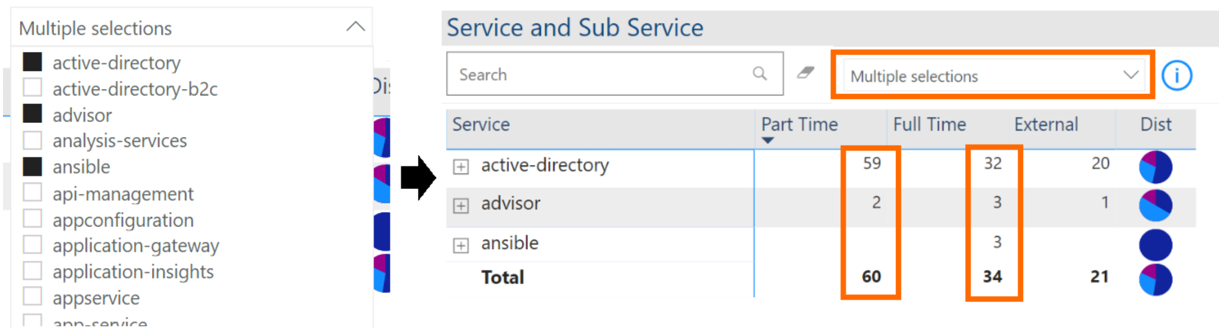


Contributor Counts - This window shows each service/subservice tag and product/technology tag in use on docs and shows the counts of contributors to each service.

Heat Map - The heat map shows a graphical view of the selected tags, and the contributor count of the selection at the top.

Choosing multiple services or products

The Service and Product page allows the selection of multiple entries using the control button.



When multi-selecting services, a count of unique individual contributors is calculated and shown in the **Total** row.

In this screenshot, 59 part-time contributors contributed to Active Directory content and 2 part-time contributors contributed to advisor content. In total, 60 part-time individuals contributed across these two services, because one of the individuals contributed to both services.

Under the hood: Contributor Insights Dashboard data sources

The contributor insights dashboard relies on the following data sources:

- Github commit history to identify authors of all commits that are published to docs.microsoft.com
- 1ES engineering system to correlate github authors with internal Microsoft users, learn their title, and then find which organization they belong to
- HR Information that categorizes Job role into a set of disciplines

How to use the page-level trends & drill-down dashboards

5/21/2021 • 2 minutes to read

You can use the Trends and Drill-down dashboards to find weekly or monthly data trends or to get referrer, verbatim, GitHub issues, or click-through data for a single page.

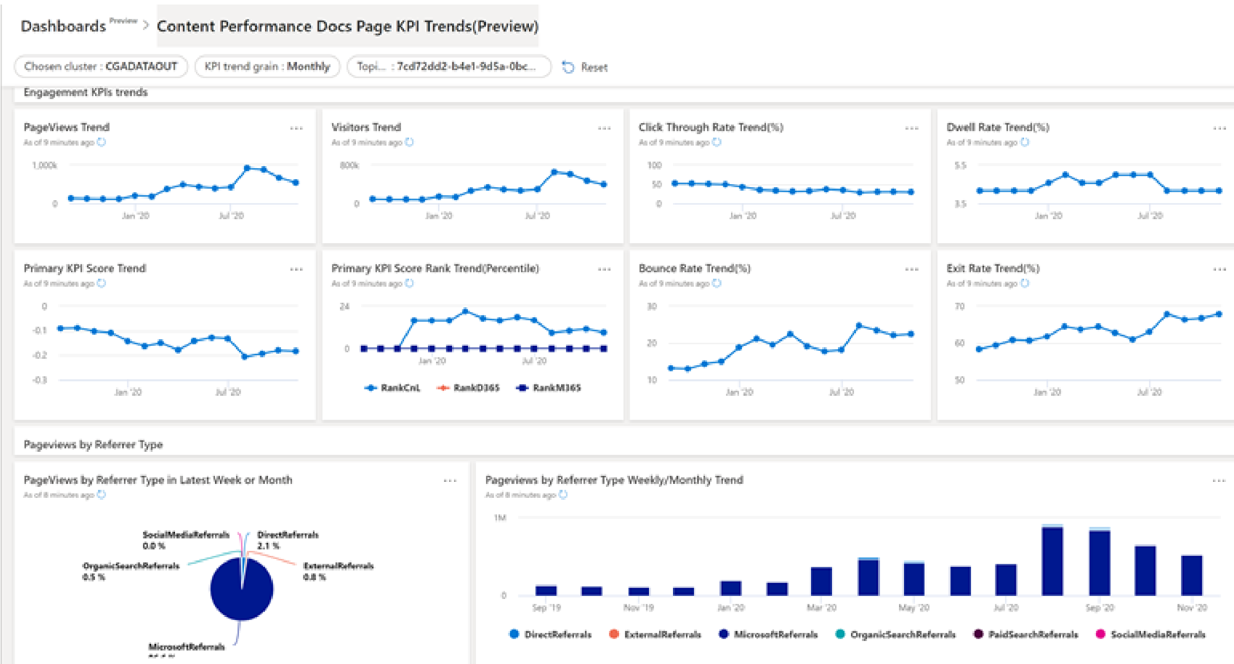
Weekly & monthly trends dashboard

Purpose: Check trends for a single article (up to 14 months or 14 weeks)

How to access: Follow these steps to access from both the Content Performance report and the Microsoft Docs Metrics.

Direct report link: https://aka.ms/contentperfkpitrends

Resources: Check out ways to improve your topics with troubleshooting tips or find definitions for these metrics.



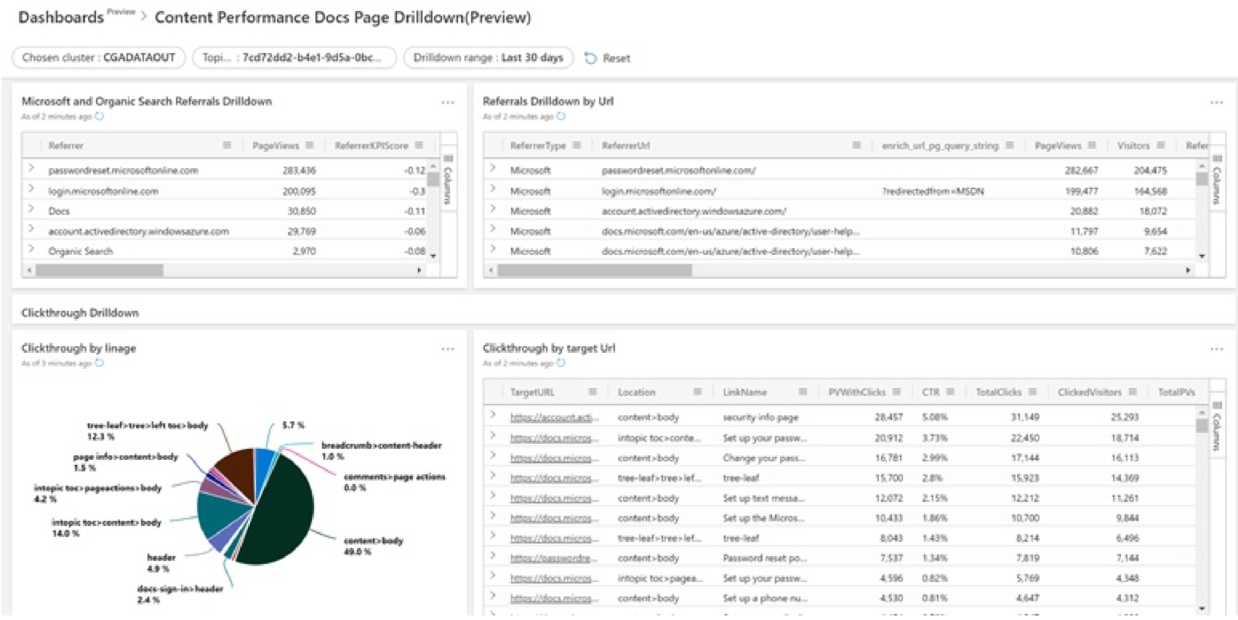
Topic drill-down dashboard

Purpose: Get all the drill-down reports for your topic in one single view (CTR, Referrers, Verbatim, and GitHub issues). You can change the time range to a preconfigured time like LastWeek, Last30Days, or choose Start and End to customize the date range by clicking the Drill-down range filter.

How to access: Follow these steps to access from both the Content Performance report and the Microsoft Docs Metrics.

Direct report link: https://aka.ms/contentperfpagedrilldown

Resources: Check out ways to improve your topics with troubleshooting tips or find definitions for these metrics.



How to access the Trends & Drill-down dashboards

To use these dashboards, you must request access to a Kusto data cluster. Which cluster(s) you choose depends on your organization. Requests require manual approval from the Customer Growth Analytics (CGA) team, which might take a few days.

Select the links in the following table to submit a MyAccess request. A prepopulated form should appear, but if not, search by the Cluster name, which is the link name.

KUSTO CLUSTER	ROLE	NOTES
CGA Kusto Public (cgadatamall.westus.kusto.windows.net)	Read-only	**Required** for all users. It's the default cluster
CGA KustoFollow DevRel (followercgadataout.westus.kusto.windows.net)	Read-only	**Optional** for DevRel team members. For use if having performance issues with Public cluster
1es CloudMine cluster (https://1es.kusto.windows.net/)	CloudMine-Data reader	**Required** for all users to view GitHub issue information in the Drill down dashboard

From the Content Performance report

1. Select a topic in either the **Documentation** or **Reference** pages of the C&L report or the BAG or Microsoft 365 page.
2. Click the **View Trends** or **Page drill-down** button. The Kusto query runs in Azure Data Explorer.

From the Microsoft Docs Metrics tool (MDM)

1. From the MDM tool window, click the **Trends** or **Details** link. The Kusto query runs in Azure Data Explorer.

Purpose of zero and low page views reports

3/31/2021 • 2 minutes to read

You can use zero and low page views (PVs) reports to help you decide to retire or archive low-performing articles. A large volume of zero and low PVs articles reside on the Docs site. These docs are of little or no value to customers. They consume resources, bring down the quality metrics for the site, and compete for search visibility.

Why use this report?

Zero and low PVs reports help writers, contributors, and stakeholders:

- Understand where they should or shouldn't invest time.
- See top and bottom articles for PVs.
- Make archiving or unpublishing decisions easier.

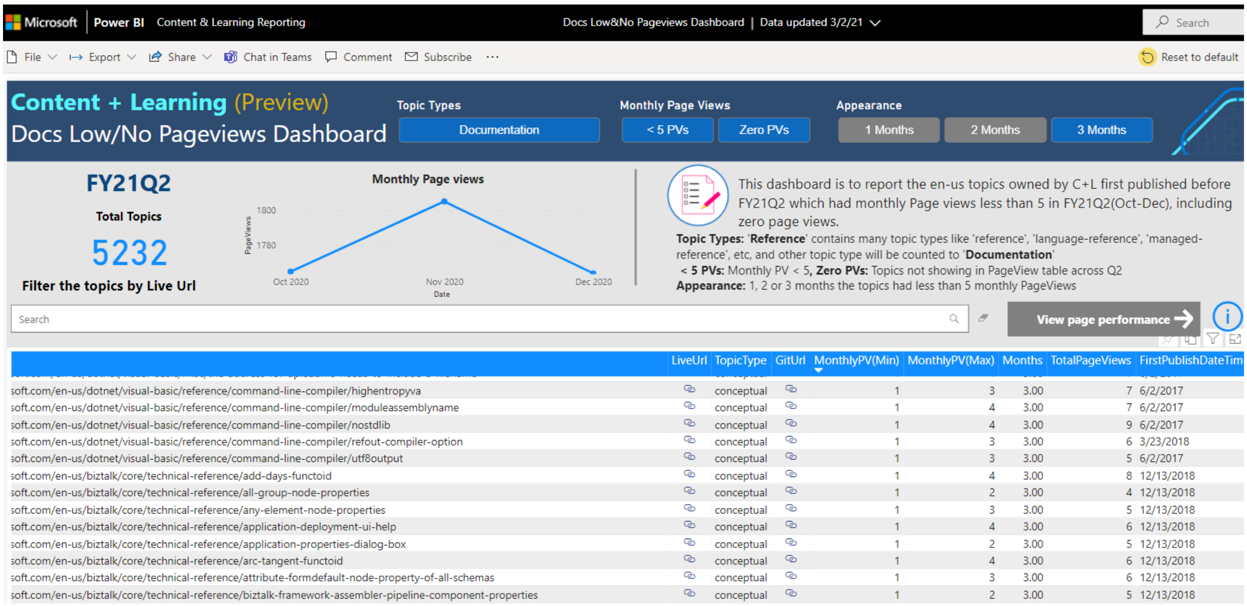
Removing content with zero and low PVs:

- Improves the SEO of the Docs site.
- Reduces costs related to OPS, such as localization, storage, and compute time for builds.

Use zero and low page views reports

The following image of a dashboard shows a list of zero and low PVs reports:

- The report only includes all C+L repos and has the data for both conceptual and reference content. By default, conceptual content is selected. On the report UI, you can select the **Reference** button or cancel the selection to see the data for reference articles.



- Under **Appearance,** three buttons are available. Selecting **1 Month** provides the data for zero and less than five PVs for at least one month in a quarter. Selecting **2 Months** provides the data for zero and less than five PVs in two months of the quarter. Selecting **3 Months** provides the data for zero and less than five PVs for all three months. If you want to get the zero and less than five PVs data for the quarter, select all three buttons. Data for an entire quarter is also the default for this report.
- The report allows filtering by repo name, topic type, ms.service, ms.subservice, ms.prod, ms.technology,

author, GitUrl, liveUrl, and last published date.

- To drill down on a particular URL, enter the URL in the **Search** field and select **View page performance**. Azure Data Explorer displays the performance of the article over a period of time.

FAQ

Why does content that was already redirected appear in this report?

This situation can happen if an article is redirected but the document ID wasn't preserved. In situations where content is moved within a repo or to another repo and the file name is changed, set `redirect_document_id` to true instead of false.

```
      "source_path_from_root": "/articles/multi-factor-authentication/multi-factor-authentication-microsoft-authenticator.md",
      "redirect_url": "./end-user/microsoft-authenticator-app-how-to",
      "redirect_document_id": false
```

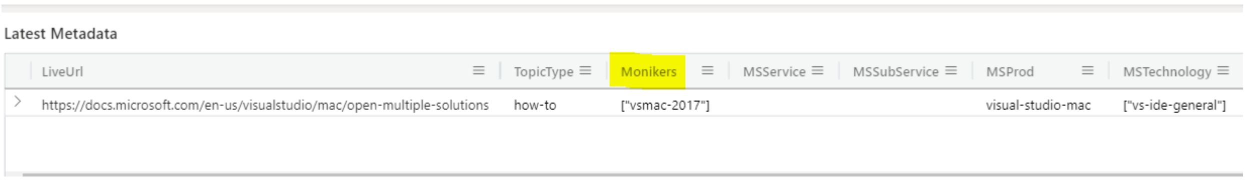
For more information about redirection and how it works, see the contributor guide article How to redirect obsolete articles

Why am I seeing reference API content even though I haven't selected the Reference button?

This situation can happen if reference content is missing Topic_type metadata.

Why am I seeing unexpected or duplicate results for versioned content?

When you find an article that you didn't expect to see in a report, select **View page performance**. The article with zero and low PVs that appears in Azure Data Explorer is probably based on a specific version. Content for each version gets its own topic key, and the report is based on the topic key metadata.



In this example, the vsmac-2017 version had less than five PVs, but not the other version of the same article.

The Doc Subset dashboard

5/21/2021 • 2 minutes to read

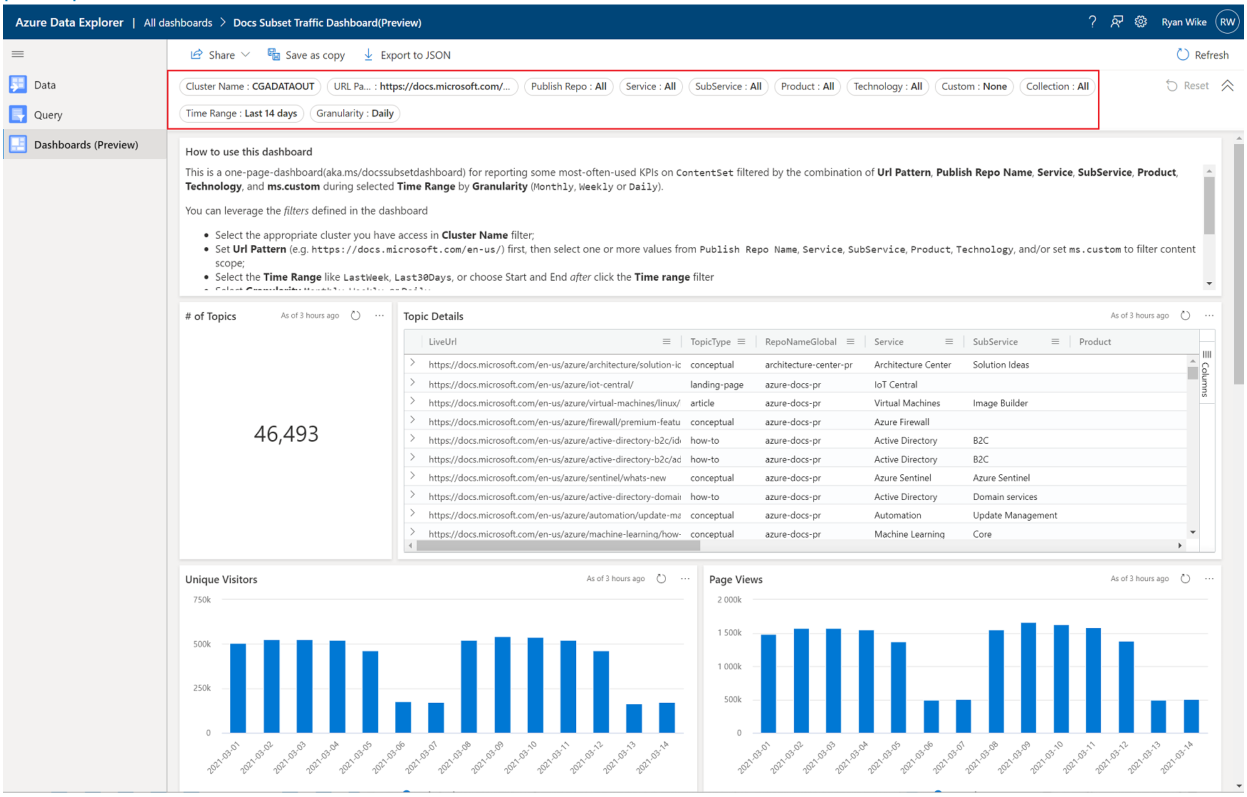
This report shows the most-often-used KPIs for a content set or set of articles (not for individual articles).

Why use this report?

- Useful for identifying trends in your content.
- Provides fine-grained time and granularity controls.
- Search more than two months into the past.
- Easy to filter on lang-local in the URL.

How to use the Doc Subset dashboard

Direct report link: https://aka.ms/docssubsetdashboard. For info on getting access to the dashboard, see the prerequisites.



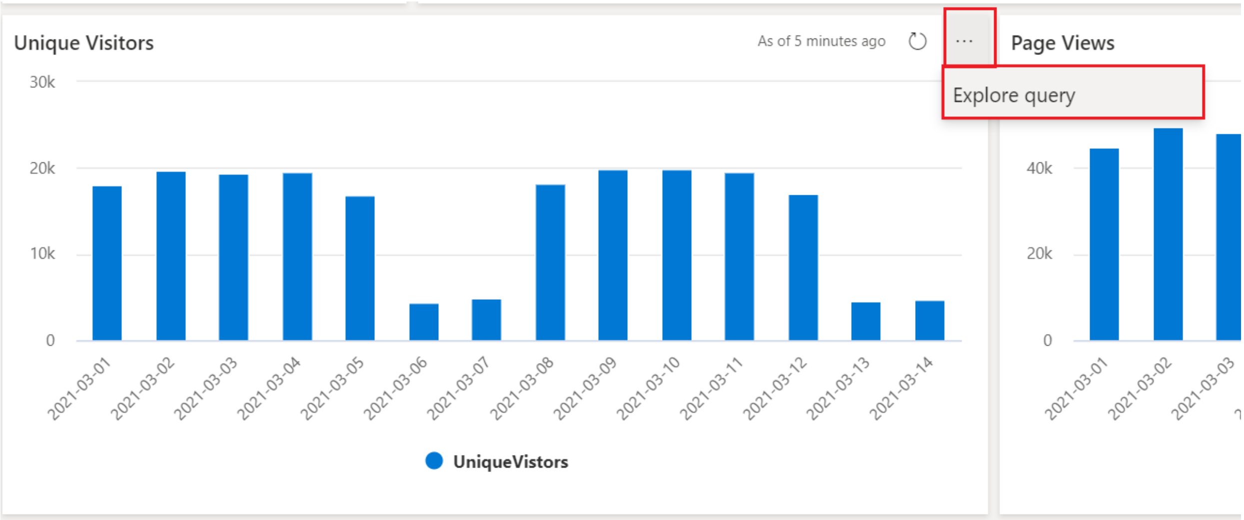
1. Select the Kusto cluster you have access to (FOLLOWERCGADATAOUT or CGADATAMALL) in **Cluster Name** filter.
2. Set **Url Pattern** with language-locale (for example, `https://docs.microsoft.com/en-us/`) first, then select one or more values from **Publish Repo Name**, **Service**, **SubService**, **Product**, **Technology**, and/or set **Custom** (ms.custom metadata value) to filter content scope.
3. Select the **Time Range** (for example, last 14 days, last 60 days, or a custom date range).
4. Select **Granularity** (Monthly, Weekly, or Daily).

> **NOTE**
>
> When querying a large content scope, it may take some time to refresh the dashboard due to Kusto performance limitations.

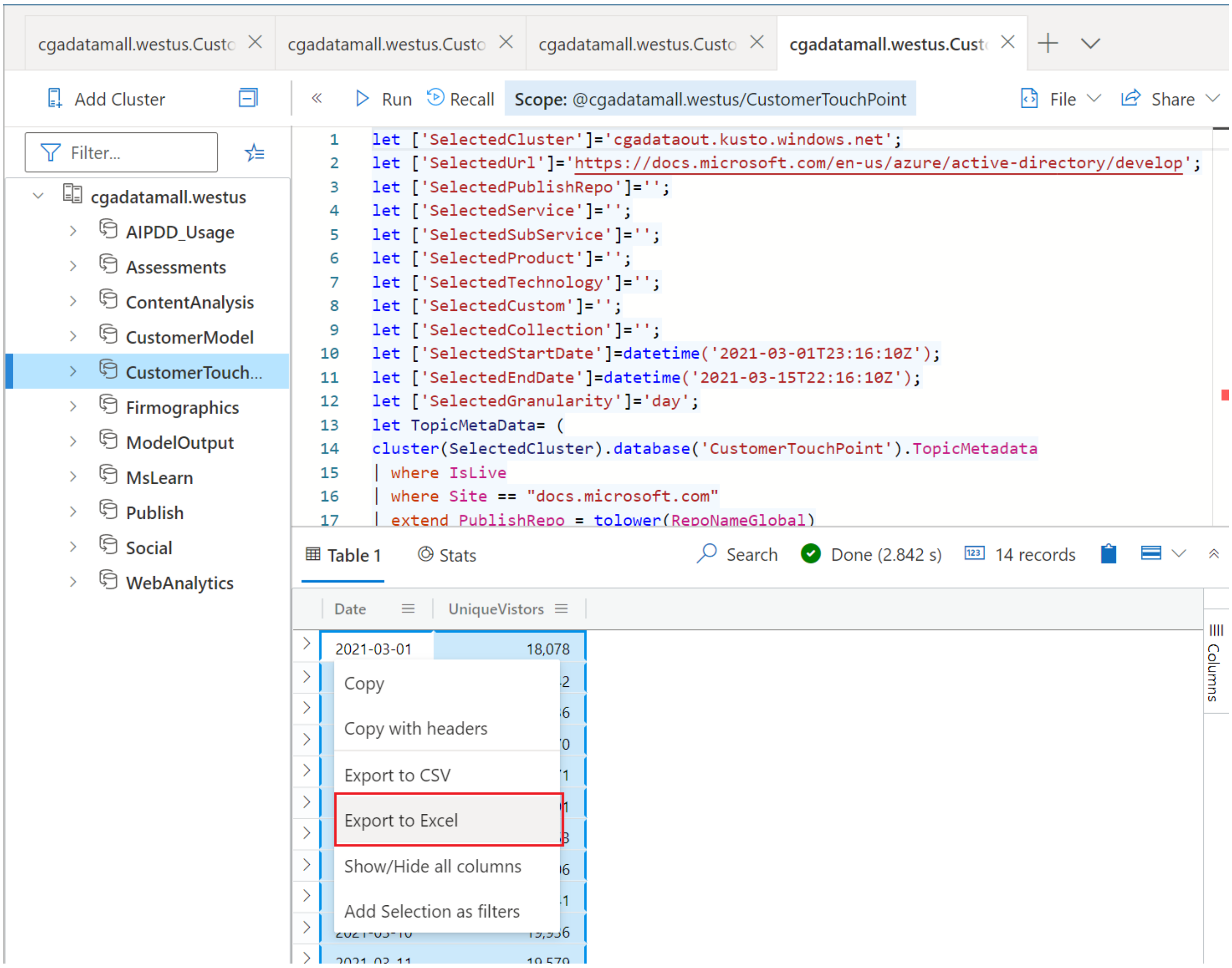
How to export data

You can export data from the charts in the Doc Subset dashboard, for example **PageViews** or **Unique Visitors**.

Select the ellipses (...) in the upper right corner of the chart and select **Explore query**.



A query window opens in Azure Data Explorer. In the results window select the data, right click, and select an export option.



How to get support or submit feedback

You can get support or route feedback through the analytics teams channel: Analytics teams channel.

How to access the DocSubset dashboard

You can access this report from within the Doc & Ref Dashboard, or directly at https://aka.ms/docssubsetdashboard.

This dashboard requires access to a Kusto data cluster. Which cluster(s) you choose depends on your organization. Requests require manual approval from the Customer Growth Analytics (CGA) team, which might take a few days.

Select the links in the following table to submit a MyAccess request. A prepopulated form should appear, but if not, search by the Cluster name, which is the link name.

KUSTO CLUSTER	ROLE	NOTES
CGA Kusto Public (cgadatamall.westus.kusto.windows.net)	Read-only	**Required** for all users. It's the default cluster
CGA KustoFollow DevRel (followercgadataout.westus.kusto.windows.net)	Read-only	**Optional** for DevRel team members. For use if having performance issues with Public cluster

How to troubleshoot lower-performing articles

5/21/2021 • 14 minutes to read

After you've identified a lower-performing article, you can use the steps in this article to make improvements to it, depending on which metrics might be causing the performance issue.

To identify lower-performing articles, follow these steps:

1. In the C&L Content Performance report, navigate to the **Documentation** or **Reference** page. In the BAG report, navigate to the **Business Applications Group** page, and in the Microsoft 365 report, use the **Microsoft365** page.
2. In the far-right of the Power BI report, use the **Filters** pane to scope to a content set based on your preference; for example, by URL, author, repo.
3. Look for articles with a KPI score rank of 25% or less, highlighted in light or dark yellow. Or in the **Filter** pane, in the **PrimaryKPIScoreRank** filter, select **Advanced filtering**, select **is less than or equal to**, enter **25**, and then click **Apply filter**.
4. Sort the table by highest page views.

PrimaryKPIScoreRank	BounceRate	ClickThroughRate	DwellRate	ExitRate
4.88	29.98%	34.34%	4.17%	42.56%
2.36	38.59%	19.43%	1.25%	65.80%
65.38	13.81%	59.66%	7.67%	77.25%
8.49	24.73%	29.00%	4.17%	67.88%
70.14	10.4			15.27%

Dark yellow: bottom 10%
Light yellow: bottom 10-25%
Gray: Not included in KPI formula

Identify the correct topic type and corresponding formula

Make sure you assign the correct ms.topic value for the article. Because the KPI Score is based on the topic type, it's important to correctly label your topics so it is compared to similar topic types. Review the primaryKPIScore calculations in the Primary KPI metric article.

For example, if the title of the article is "How to troubleshoot 2-factor authentication problems in Windows Intune" and it contains troubleshooting procedures, and the article is tagged as **conceptual**, then **Troubleshooting** would be more appropriate so that troubleshooting articles are compared to other troubleshooting articles.

If you change the ms.topic type, the KPI Score Rank or score (if topic type has different formula) will be recalculated in the next monthly report. Correct classification could change your score, so you may want to change only the ms.topic type before you invest more effort.

Tips for high bounce rate

If BounceRate is flagged, it means that the topic didn't meet customer expectations and they left within 5 seconds without engaging with anything on the page. They didn't click a link, and they didn't scroll. Often a particular referrer can cause poor performance. How is BounceRate calculated?

How to influence bounce rate:

- Check referrer traffic. Are there any referrers that have high bounce rates, signaling that the link to your topic isn't meeting customer intent. If possible, work with referrer owners (for example, product or website owners) to point to better content or change the content to meet the expectations of that referred traffic.

- Review customer feedback. Are there any verbatim or GitHub issues that provide hints to customer issues?
- Consider clearer headings and intro paragraphs to set expectations. If customers can't tell what's in it for them right away, they'll bounce.

Tips for low click-through rate

If the ClickThroughRate is flagged, it might indicate that there aren't links in the body of the topic or the links that are present aren't visible or relevant. How is ClickThroughRate calculated?

How to influence click-through rate:

- Improve bounce rates. If you can improve your bounce rate by decreasing it, overall page engagement should also improve.
- Cross linking. Ensure the topic has relevant cross links to related topics or next steps. Review the click patterns and verbatim for possible clues.
- Reposition links. On pages designed for navigation, like hub and landing pages, it's useful to identify links/buttons lower on the page that get more clicks and move them up. Remove links that customers aren't clicking to simplify choice. For longer articles, consider moving high value links earlier in the topic.
- Consider clearer link text and calls to action so users know what to expect before they click. Link text should be short and descriptive.

Check what and where users are clicking

For poor performance on click-through, it's useful to look at what customers are clicking on a page. You can check click activity for a page in two ways:

- Use the Microsoft Docs Metrics extension for a quick visual of where users are clicking on the page. Note that the click totals are aggregated by target URL, so if the same link appears multiple times on the page, the total all of the clicks for that URL, not the specific link.
- Use the click-through drill-down report that's in the Drill down dashboard. On the **Documentation** or **Reference** pages, select a single article from the table, and then click the **Page drilldown** button in the top-right corner. (Requires Kusto permissions)

Click report details

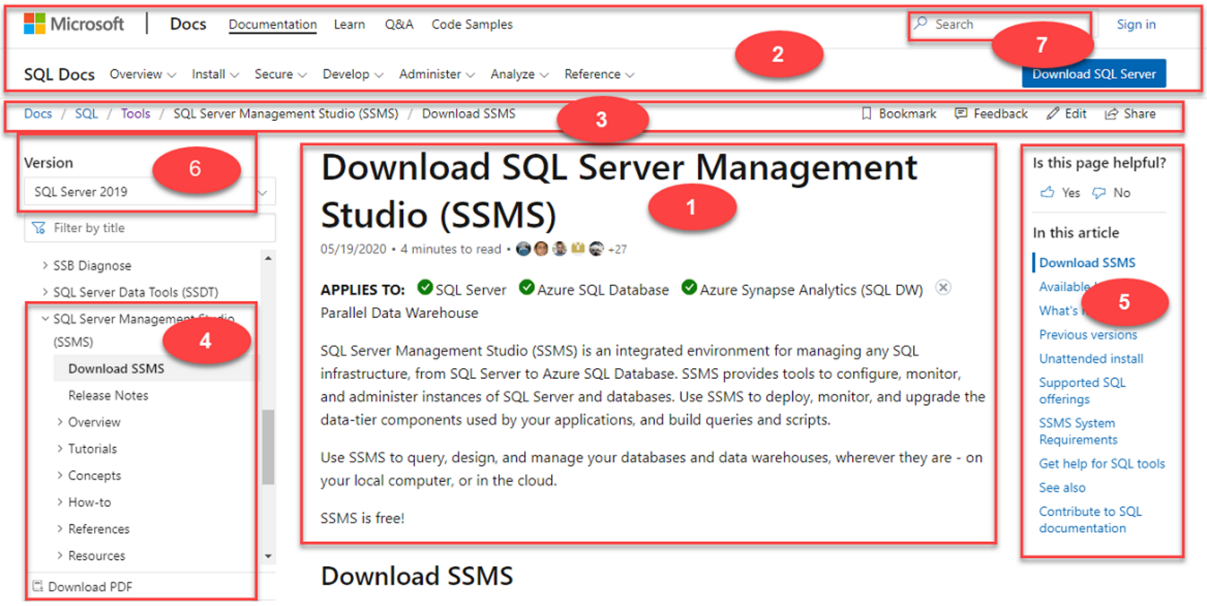
The Page drilldown report includes a pie chart of click activity by area and a table of each link with relevant metrics.

COLUMN HEADING	DEFINITION
Location	Where the link is located on the page See Link location
LinkName	Text of link
Target URL	URL associated with URL
Clicks	All clicks on the particular link
ClicksPVs	All clicks on a particular link by page view. Will be less than "clicks" as the links are deduped
ClicksVisitors	All clicks on a particular link by unique visitor. Will be less than clicks or clicks by PV as this value is deduped by user
TotalPVs	Number of times the selected page was loaded

COLUMN HEADING	DEFINITION
TotalVisitors	Number of individual users to the page identified by a web cookie
CTRbyPV	Number of clicks by page view as a percentage of total page views for this page (pageClicksPageViews/TotalVisitors)
CTRbyVisitor	Number of clicks by visitor as a percentage of total visitors (pageClicksVisitors/TotalVisitors)

Link location

Here's a key for where links are located on the page. If you right-click the page and view source you can search for "data-bi-name" and that typically provides the location information.



LOCATION	AREA
1	content>body
2	header
3	content-header (breadcrumb and comments)
4	TOC: **tree-leaf>tree>left toc** (mobile version) and if includes >**body** (tablet resolution or larger)
5	In this article: intopic toc
6	moniker-picker
7	search (blank, use target URL for docs.microsoft.com/search to identify)
8	footer (terms of use, language selector)
9	recommendations

Tips for low copy-try-scroll rate

If CopyTryScrollRate is flagged, it means that customers didn't frequently scroll, copy code, or use the Try button. How is CopyTryScrollRate calculated?

How to influence copy-try-scroll rate:

- Check your BounceRate. If BounceRate is high, address BounceRate and SEO issues.
- Check your ClickThroughRate. If ClickThroughRate is high, and CopyTryScroll is low, this is usually expected (customers may click a link instead of scrolling). In this scenario, generally Primary KPI is good, so this is a good result.
- Check your topic length. Very short topics have low CopyTryScrollRate, because for some window sizes there isn't opportunity to scroll. Evaluate whether the topic should be combined with other topics, or is a candidate for removal.
- Check the customer focus. Is the customer finding what they need before scrolling? You can use MedianMaxScroll to help investigate.

Tips for low dwell rate

If DwellRate is flagged, it means that while customers aren't bouncing, they aren't staying on the page as long as expected either. How is DwellRate calculated? DwellRate is no longer a calculation in the new performance report.

How to influence dwell rate:

- Review customer feedback. Are there any verbatim or GitHub issues that provide any hints to customer issues?
- Run the referrer drill-down query. Are there any referrers that have high bounce rates, signaling that the link to your topic isn't meeting customer intent. If possible, work with referrer owners (for example, product or website owners) to point to better content or change the content to meet the expectations of that referred traffic.
- Consider content structure. Customers often scan webpages to find information.
  - Can you tighten the topic to focus on the customer intent? Can you break up large paragraphs into bullets or use visuals to explain concepts?
  - Is the article visually appealing? Does it have an image "above the fold," or is it a "wall of words"?
  - If there are distinct topics that won't compete in search, consider breaking a larger article into smaller articles with more focused intents.

Tips for high exit rate

If ExitRate is flagged as high for a landing, hub, or sample index page, it means that customers aren't clicking through, so the page is not serving its purpose, which is to drive customers to other pages within the content set. How is ExitRate calculated?

How to influence exit rate:

- Optimize for Search. Review the Search-related metadata to make sure your title, description, and H1 are optimized for the right audience.
- Review click patterns. Review your link selection to find areas to improve and consider clearer link text and calls to action so customers know what to expect before they click. Link text should be short and descriptive.

Check referrer traffic

A common problem with content is a mismatch between the expectation set by the referrer and the content on the page. It's often useful to look at a breakdown of all the referring traffic sources, and how performance metrics vary across each one. You may find a referrer that is driving many page views and causing high bounce rates compared to other referrers.

1. To run the referrer report from the Content Performance report, on the **Documentation** or **Reference** page, select a single article from the table, and then click the **Page drilldown** button in the top-right corner. To run the referrer report from the MDM extension, click the **Details** link.
2. An Azure Data Explorer dashboard runs for your page (based on TopicKeyID). If you don't get results, check your cluster permission or change the **Chosen cluster** at top of report.
3. To identify which referrers are causing the low page performance, sort the **PageView** column by the highest traffic, and then look at the bounce rate column to see which referrers have the highest bounce rates. Alternatively, you can use the Referrer KPIScore. Each referral is assessed by using the same Primary KPI Score formula as the topic.

Search referrer traffic with high bounce rate

If the high bounce rate is coming from search traffic, such as Google, then optimizing your page for Search is the best bet.

- Use the steps outlined in SEO: Basic techniques for good web writing.
- Does your topic share common search queries with a different audience type; for example, could consumers be landing on it because of how it displays in search results? Can you update the metadata title or the opening paragraph to match the article's intent better or add a Note that directs them to a relevant Support topic?

Direct referrer traffic with high bounce rate

High bounce rate caused by direct traffic is tough to pinpoint as direct traffic doesn't have a referring URL. What is Direct traffic?

- Check to see if there are any fwlinks, aka.ms links, or other links from the product UI to understand the context customers are being routed to your page. If your topic changed URL structures, be sure to look for those variations in the fwlink tool and update the target URL. You might need to change the page the link is pointing to or work with the product team to remove the link.
- Add a campaign ID to your in product links, which will add a parameter you can see in the Referrer drill-down report. More details about this approach are outlined in the topic Links to articles from the UI.
- Server-side redirects. Legacy blogs, forums, or Support sites might contain links to your topic that are now 301 redirects; for example, with "msdn.microsoft.com" in them. If possible, reach out to the site owner to get those links updated or removed.
- Bot or other fraudulent traffic. High levels of direct traffic may also be an indication of fraudulent bot traffic that needs to be reported. Some testing systems are also known to generate large amounts of direct traffic, particularly for landing pages. If you suspect this might be the case, reach out to the C&L Insights and Analytics team so they can investigate and add to the exclude list.

External or Microsoft referrer traffic with high bounce rate

If external or Microsoft-owned web pages are responsible for the high bounce rates compared to other traffic sources, then you should check the context of the backlink on those pages. Is it misleading? If so, modify the link text or remove or replace the link so the correct context is set.

Check verbatim comments and GitHub issues

Incorporating customer feedback from feedback sources is always a good thing to do when you are troubleshooting article performance.

- **Verbatim in Content Performance**. In the Content Performance report, check the **Verbatims** column to see if there are any for the month. If there are, click the topic to highlight it, and then click the **Page**

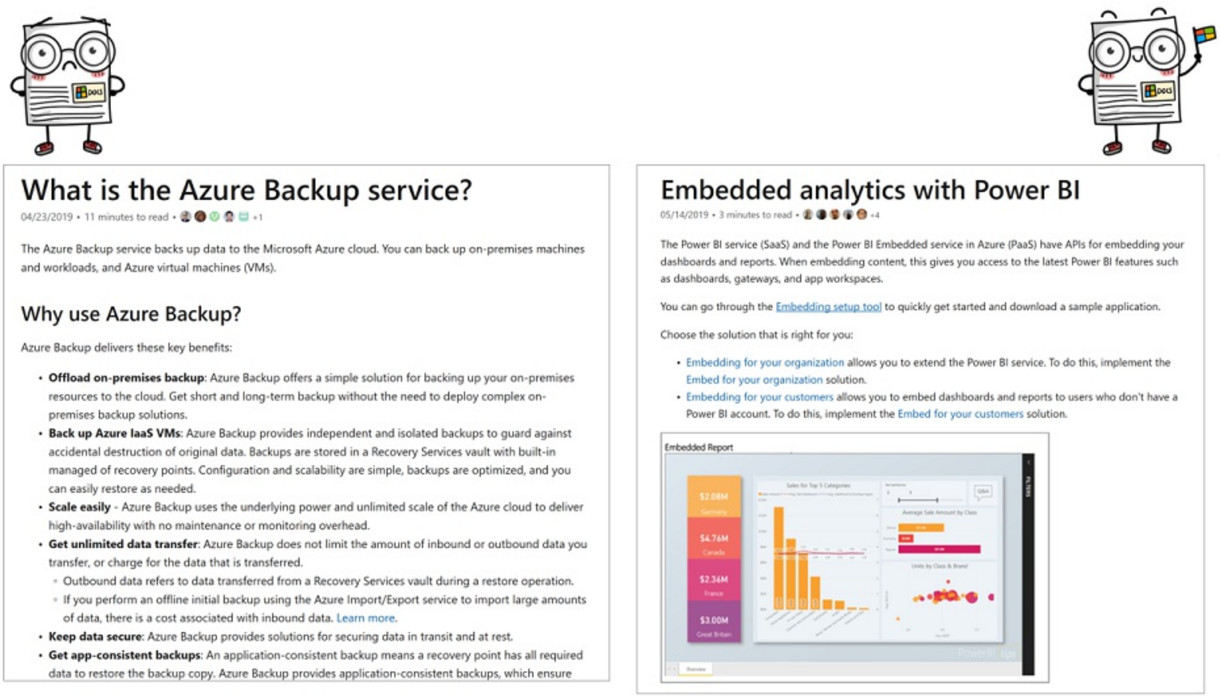
drilldown button in the top-right of the page to open the Drilldown dashboard for the topic. (Quick tip: In the Drilldown dashboard, if you click the ellipses in the right side of the tile, and then click **Explore query**, you can manually change the date range in the Kusto query to include previous months).
- **Verbatim in the MDM tool**. In the Microsoft Docs Metrics tool, check the **Satisfaction** section. To access the verbatim, click the **Details** link to open the Drilldown report.
- **Verbatims for multiple pages or months**. To see customer feedback across a content set or just one page for the past 90 days, use the Verbatims report.
- **GitHub issues**. Check the GitHubOpenIssueCount column, highlight the topic, and then click the **Page drilldown** button in the top-right of the page to open the Drilldown dashboard for the topic. This requires access to the **CloudMine-Data reader role** in the 1es CloudMine cluster (https://1es.kusto.windows.net/).

Verify content follows best practices

Frequently, low performing articles simply don't comply with our core content authoring, voice or SEO guidance, and that lack of compliance drives low performance. You may need to update the article to better address a single customer intent, to include images, or to reduce extraneous text. If the article is long, you may need to break it into smaller articles.

- Voice and writing guidance is available, and it makes a difference!
- Request a developmental edit to obtain feedback on voice and writing guidance.
- SEO: Basic techniques for good web writing.
- Consult the writer's SEO cheat sheet. Pay particular attention to: titles, meta descriptions, H1s
- Freshness + maintenance checklist. Check the Freshness column to see if it's been awhile since the content was updated. Screenshots, links, or outdated information might be causing the underlying issues with the topic.

Looking at a higher performing article (not flagged in yellow) of the same type may provide insight. For example, let's compare stats for the following articles.



We see that the articles, while of the same type and with similar monthly page views, were different from one another. The higher performing article follows our modern voice guidance and content guidelines for well-written content. The lower performing article does not.

ARTICLE 1	ARTICLE 2
Appx page views 10,000/month, page type = overview	Appx page views 10,000/month, page type=overview
11 minute estimated read time	3 minute estimated read time
Wall of text	Attractive, even unusual graphics, one of which is above the fold
Little white space, not easy to scan	Noticeable white space, easier to scan
Tons of tables with a lot of minutiae	
Next steps section uses text-based links	Next steps section uses large clickable buttons
Low levels of click through	High levels of click through
Seems to lack a single customer intent	Clearly focused customer intent

In this case, the recommendations to fix Article 1 included:

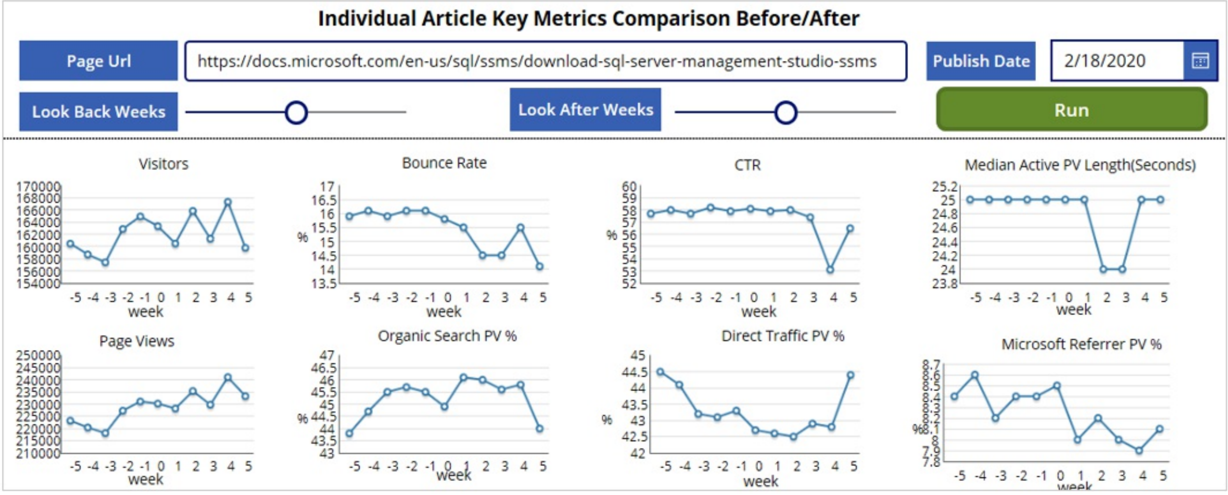
- Identify clear customer intents for the different parts of the article and break it up into multiple, smaller, more focused articles.
- Add some visuals as appropriate – you need them to draw the user in.
- Add relevant links in more prominent places in the article to encourage click through

Bubble up product pain points

You may be using the steps in this article as part of trying to raise CSAT for a technical article on docs.microsoft.com. After analyzing the article's performance, you may conclude that the right users are finding the article but they are very unhappy because the document covers a product feature or area that generates a lot of customer pain. That pain is then expressed through negative verbatim, angry GitHub issues, and low CSAT that cannot be addressed by changes in the article. In these cases, use the negative data around page views, low CSAT, and customer feedback to try to drive a product fix. We can fix bad documentation, but we cannot fix a broken product experience with a doc.

Article Assessment: Measure the impact of your changes

With the new Article Assessment tool page in the Topic-Level Content Performance Dashboard report, you can compare the performance of a single article, before and after you make changes to it.



When you use the Article Assessment tool, keep the following in mind:

- You must manually specify the publish date in the assessment tool.
- Remove version monikers from the URL. For example, trim `https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver15` down to `https://docs.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms`.
- Week 0 in the results represents the data for the full calendar week that includes the publish date. Week 1 is the first full calendar week after the week of publish. Weeks run Sunday to Saturday for this report.
- This page of the report can return data for the current month; whereas, the Documentation and Reference pages show aggregated monthly data.

While a minimum of 3 to 4 weeks is needed before SEO improvements can be discerned, the core bounce, dwell, clickthrough, and exit rate metrics typically will show a response during the first full week after the publication week (week 1 in the assessment tool).

As you make more changes within a content set, month over month, you should monitor that content set for overall improvement of the percent of page views that go to higher performing content. On the **Overview** tab of the content performance dashboard, filter to the content set you care about and note the monthly changes overall.

Resources

- Content performance report
- Learn how to use the Content performance report

Content performance reporting principles

11/2/2020 • 3 minutes to read

The most common question or concern about the content performance reporting is that a particular metric is unfair or inappropriate because we can imagine scenarios when the metric's high or low score isn't necessarily a good or bad thing. This article describes the principles that inform our approach to content performance.

We compare articles to their peers and look for outliers.

If you think it's unfair to measure, say, the ClickThroughRate on Overviews because Overviews are just walls of prose with few links, keep in mind that when we're determining under- or over-performance, we're comparing that rate against the rate of other Overviews. Those imagined scenarios where the metrics don't seem “true” to the experience apply to all Overviews, and therefore factor out. In the end, we still have to wonder and investigate why some Overviews have such anomalously low rates among their peers. This is why we compare articles to their peers and then look for outliers

Engagement is a good thing to see in aggregate data.

ClickThroughRate, to continue the example, is a good indicator of a good experience in nearly all situations. And so are the many other metrics we use (CopyTryEngagementRate, StartTutorialRate, DeployClickRate...); they show users further engaging with our content. In the ClickThroughRate example, it's hard to argue that it's a bad thing when users choose to click and view another page (or copy code, or start a tutorial...) after reading an Overview that introduces them to a service, especially when we're also measuring whether users dwelt long enough to have read the Overview (DwellTimeDelta) and combining that into the performance score. These behaviors are good things to see, and they're easily and reliably measured. Yes, it's possible lots of clicking is people who are lost. But behaviors like being lost are easily detected with other metrics we employ, and with other user research methods we employ.

Reporting is only as good as your metadata.

We certainly do have limited metadata accuracy in some cases (lots of different content types lumped into 'article' and 'conceptual'), which prevents us from measuring some of them appropriately or "fairly". However, the solution here is not "hold off on measuring content performance until we can fix the metadata." The solution is to fix the metadata. We need good metadata for lots of reasons beyond reporting: personalization, search, browse...

We can change the metrics, but only if we've done the work to know we should.

That said, we do routinely look for opportunities to try new metrics, or stop using some metrics that don't prove to be actionable. However, proving a metric is not actionable requires that we actually try to improve it on a large number of articles, and be ready to show our work. Then we can really know when a metric isn't worth our while. If we're willing to put in the work, we will definitely find some duds down the road, but that's not the point of putting in the work.

Content performance isn't personal performance.

We need to remember that with content performance, we're not measuring ourselves nor our work. We're measuring engagement and using it as a tool to help us do our work. Content performance is not tied to our

Connects. If anything, what's tied to our Connects should be what we learn together from using a common framework for evaluating our users' experience with our content. Authors should have individual goals around using analytics and iteration to identify generalizable insights so that we can discover tips and tricks that drive performance when applied in various contexts and educate the team about it.

Release notes and known issues for the content performance dashboard

5/4/2021 • 15 minutes to read

Release notes

May 2021

- Kusto cluster update. The endpoint for **DevRel read-only** users of the CGA Kusto cluster **cgadataout** changed on April 20. This includes the CustomerTouchpoint, WebAnalytics, MsLearn, and Publish databases. The new location is **CGA KustoFollow DevRel** (followercgadataout.westus.kusto.windows.net). If you already had permissions to **cgadataout**, your permissions should have been automatically migrated to the new cluster.
- Added Page Views (PVs) month over month (MoM) to show % change, exposed in a couple of places in the report:
  - On the **Overview** page, in the bar chart as a Tooltip. Hover over each month to expose the Tooltip or export the monthly data by clicking the ellipses in top-right corner, and then **Export data**.
  - On the **Overview** page, in the Performance drilldown section, as a new column titled **PVs MoM**.
  - On the **Documentation**, **Reference**, and Partner-specific pages, as a new column titled **PVs MOM**.
- To show whether a page changed its rank from the previous month, look for the new column **PrimaryKPIScoreRank Change** in the **Documentation**, **Reference**, and Partner-specific pages.
- Added notices to highlight any months with data loss on the **Overview** page. Any month with a data issue will have info in the Tooltip for that month and an alert in the Performance drilldown section of the page.

February 2021

- **High bounce rates** (Feb 2021) - In February, some engagement events related to dwell on pages were under-counted, resulting in over-counted bounce rates, which will show a spike in this metric. This data gap impacts all of docs.microsoft.com and related reports. See Known issues for root cause
- Added Title as a filter on the Documentation, Reference, BAG, and Microsoft 365 pages
- Filtered out nine low-priority repos for the BAG report
- Fixed the ability to export underlying data so that service and product columns are included

January 2021

New formula for most pages. Replaced DwellRate with a new metric CopyTryScrollRate. See more details.

- Non-nav pages: (CopyTryScrollRate + ClickThroughRate) – BounceRate
- Reference: CopyTryScrollRate – BounceRate
- No change to hub/landing formula: 2 * ClickThroughRate - (ExitRate + BounceRate)

New version of reports. The new formulas required us to rerun the data for all pages, which updated our baseline trends, and we've created a new version of the report so there's an archive of the legacy report for comparison. As a result, any updates to metadata or page redirection might impact your historical page view data. The new version uses the same aka links and the legacy report can be accessed through these links:

- https://aka.ms/contentperf-legacy (Content & Learning)
- https://aka.ms/contentperf-bag-legacy (Business Application Group)
- https://aka.ms/contentperf-m365-legacy (Microsoft 365)

Overview page updates. A few notable updates:

- Trend chart now shows 13 months instead of 7.
- New filter for URL (level 1 and level 2).
- The Performance drill-down report is now on this page rather than a standalone page and includes a column for % of high performing articles.

Documentation and Reference pages.

- CopyTryScrollRate is part of the KPI formulas and other metrics are for available for diagnostics:
  - CopyTryScrollRate
  - CopyRate
  - TryRate
  - MedianMaxScrollRate
  - OrganicSearchReferrers
- Weekly and Monthly trend data is available through the **View Trends** button in the top-right corner.
- Referrer, Verbatim, Click-through data, and GitHub issues are available through the **Drill-down** button in the top-right corner rather than in individual columns in the table.

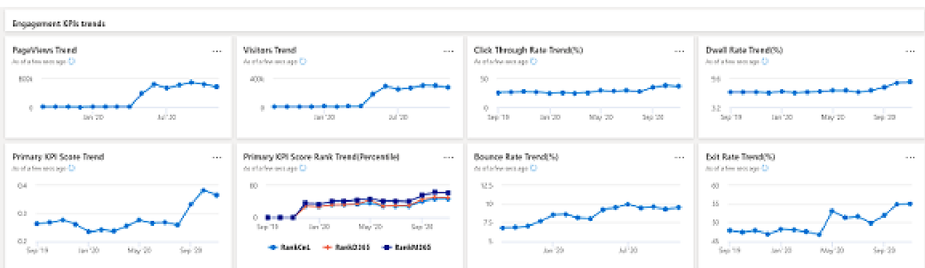
Report scope changes. Several URL paths were either excluded from all reports or added to a different report based on the organization the content belongs to. The overall page views will differ between the new and legacy reports.

ORGANIZATION	EXCLUDED	ADDED
Content & Learning	/search (3.6M PVs) /locale (173K PVs) /answers (171K PVs) /compliance (155K PVs) /ai-builder (33K PVs) /teamblog (29K PVs) /contribute (23K PVs)	No change
M365	No change	/kaizala /stream /openspecs (400K PVs) /compliance (155K PVs)
BAG	/dynamics	/ai-builder (33K PVs) /industry (6K PVs)

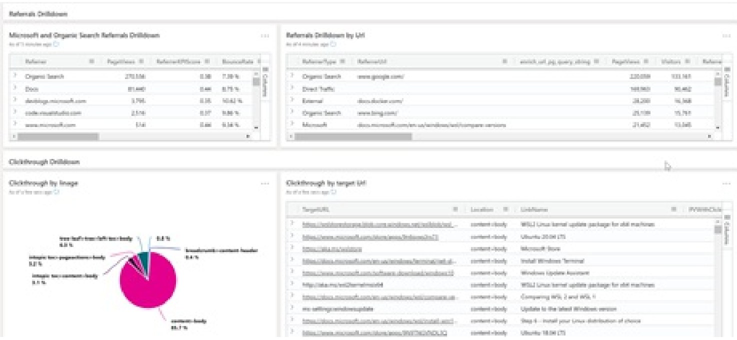
November 2020

New Trend and Drill-down dashboards. Try out these two dashboards available in the Content Performance report for a quick snapshot of your topic.

- **Trend dashboard**. Get weekly + monthly topic trends for latest 14 weeks or months.



- **Drill-down dashboard**. Get all the drill-down reports for your topic in one single view (CTR, Referrers, Verbatims, and GitHub issues). You can Customize **Time Range** like LastWeek, Last30Days, or choose Start and End by clicking the **Drilldown range** filter.



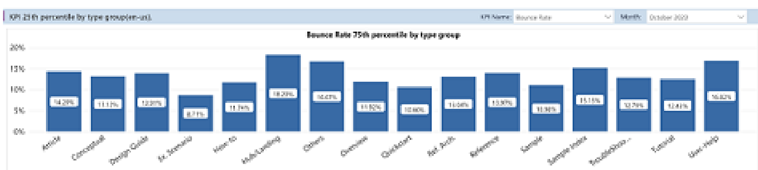
- **How to use**. Select a topic in either the Documentation or Reference pages, and then click one of the buttons in the top-right of the page.

KPI benchmark charts. Located on a new KPI Benchmark page, these charts and table show the 25th and 10th percentile by Topic Types for all topics in the report-not by content set. Use the 25th or 10th percentile buttons and Month drop-downs to flip between core metrics across topic type groups to see the monthly trends.

Change to KPI score baseline. ~36K articles were moved from C&L reporting to BAG/M365 (openspecs and Power platform). With fewer articles in C&L, the new competitive landscape changed and inevitably impacted article ranking. The impact mostly occurred with *Article*, *Conceptual*, and *Others Type* groups. We saw close to a 6-point decrease in the all-up KPI score for C&L (72.41% to 66.69%).

October 2020

- New benchmark chart added to the Overview page. It shows the 25th percentile by Topic Types for all topics in the report-not by content set. Use the KPI name and Month drop-downs to flip between core metrics across topic type groups to see the monthly trends. Note that this chart will likely move to a separate page in November. You can use this chart to:
  - Identify the bottom 25th percentile for a particular metric by type group and what you need to achieve to move out of the bottom.
  - Get a quick idea of how one topic type compares to another. It can help set expectations when changing your ms.topic to the appropriate ms.topic value.

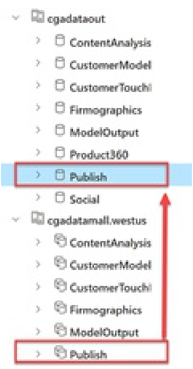


- Article assessment page. A Cluster Name drop-down was added, allowing you to switch between two Kusto clusters to find the one that resolves faster.

- Kusto cluster permissions. Many features within the Content Performance report require you to have cluster permissions for them to run. The CGA team owns the management of these permissions.
  - If you are in the Dev Relations organization, you can request access to both the cgadataout (cgadataout.kusto.windows.net) and the CGA Kusto Public (cgadatamall.westus.kusto.windows.net) clusters.
  - If you aren't a member of Dev Relations, you can request access to the CGA Kusto Public (cgadatamall.westus.kusto.windows.net) cluster.
- Excel reports. By request, the downloadable versions of the Docs monthly report were separated by repo (azure, win32-pr, windows-driver-docs, and open-specs_windows, and other) instead of high and low page views to make them faster to download.

September 2020

- **Referrer drill-down query**. This query is currently failing. As a workaround, you can cancel the query and instead run it by using the cgadataout cluster. Request access to the cgadataout cluster if you don't already have it (`cgadataout.kusto.windows.net`). There's an issue with the underlying function in the cgadatamall cluster and an active bug with cgasupport.



- **New colors to signal low performers**. We heard your feedback about the red and pink conditional formatting being too harsh and have changed to yellow to match the low performing category on the Overview page. Ranking within the bottom 10% is now dark yellow and ranking in the bottom 10- 25% is light yellow.
- **Additional metrics available**. You can now see metrics, such as click-through and exit rates, even when they aren't used to calculate the KPI score. In those cases, the value appears on a grey background. The extra data points can help you with sleuthing.

PrimaryKPIScoreRank	BounceRate	ClickThroughRate	DwellRate	ExitRate
4.88	29.98%	34.34%	4.17%	42.56%
2.36	38.59%	19.43%	1.25%	65.80%
65.38	13.81%	59.66%	7.67%	77.25%
8.49	24.73%	29.00%	4.17%	67.88%
70.14	10.4			15.27%

Dark yellow: bottom 10%
Light yellow: bottom 10-25%
Gray: Not included in KPI formula

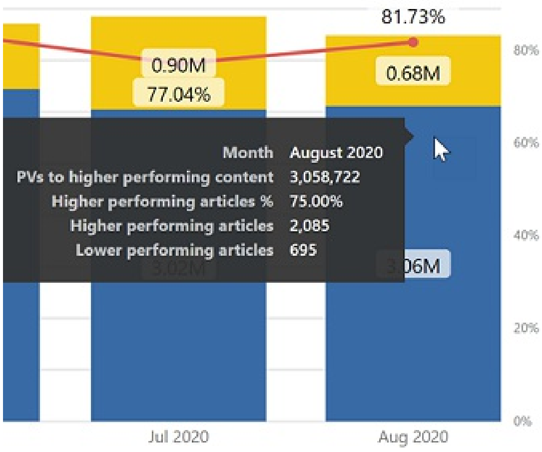
- **New direct traffic details exposed**. A new column in the referral drill-down titled “enrich_url_pg_query_string” can help identify traffic coming from MSDN redirects or F1 Help.

ReferrerType	enrich_url_pg_query_string	PageViews
Direct Traffic	?redirectedfrom=MSDN	175
Direct Traffic		6,350
Direct Traffic	?OCID=WinClient_Ver1703_Settings_DevMode	81,272

August 2020

- We needed to reset the historical data on the Overview page for June and July. Impact is to hub & landing pages. For more information, see Known issues.

- On the Overview page, added the total articles for the low and high-performing groups and the % of articles that are high performing. The totals include both the docs and ref articles. To access, hover over the column to view the tooltip. This update helps align the article totals with the totals in the “Articles shown” total at the top of the Documentation and Reference pages.



- Moved the Windows Console topics from Microsoft 365 to the C+L report.

- Added Microsoft Teams topics to the C&L report in addition to keeping it in the M365 report.

July 2020

- Added the Acrolinx clarity score next to the Acrolinx score. If the Clarity score is low, content owners can use recommendations in the report to improve their article.
- Added the MSAuthor column on the Documentation and Reference pages for easier content owner identification.
- Permissions update. To use many of the features in the content performance report, you need the right permissions. Based on CGA guidance, the permissions have changed to the cgadataoutmall cluster (public). Links to the permissions form are updated in the report and all the drill-down Kusto reports now point to this cluster (previously: cgadataout). This update should fix any “permissions denied” issues.
- Downloadable Excel reports available from Overview page. Added all primary metrics data for articles such as clickthrough and exist rates to the spreadsheets and added grey color formatting when the metric aren’t used in score calculation. The added metrics might help content owners with more sleuthing. Also added columns that were recently added to the Power BI report that weren’t in exported reports.

UI changes:

- Removed timestamp in the LastReviewed column to tighten up space.
- Article assessment page now has a link to how to use it.
- Expanded the scope for the GitHub OpenIssues link. Originally, we only provided the link for the articles that had open issues in the last 365 days.

June 2020

New and updated functions:

- Click-through report. Clicking the link in the **ClickThroughsDrilldown** column launches Azure Data Explorer and provides the number of clicks for each link by link name or target URL and its location on the page. See the Troubleshooting article
- Referrers report. Added CurrentPageKPIScore, ReferrerKPIScore, and KPIScoreDelta to the ReferralsDrilldown function. Both scores are using the same formula according to the current topic type, and a positive KPIScoreDelta means the referral is good. You can leverage the Page Views and KPIScoreDelta in the ReferralDrilldown when you analyze the referrals. See the Troubleshooting article.

UI changes:

- Increased the font size by 1 pt in the table to make it more readable
- Removed "|Microsoft Docs" from Title column
- Moved OrganicSearchRate after "Visitors"
- Moved CSAT-related columns ahead of GitHub related columns
- Added the Help (?) icon to the Documentation and Reference pages to link to Metrics definitions.

Metrics definition change:

- Changed the **DwellTimeDelta** to **DwellRate** to measure the median dwell time as a percentage of estimated time to read. Formula is MedianDwellTime/Estimated reading time using a positive % to make it more explainable. The score for a topic changed to a positive value, but doesn't impact the score ranking. Original formula (MedianDwellTime – Estimated reading time)/Estimated reading time)

May 2020

- Expanded the content scope to cover all Topic types, including **How-to**, **Troubleshooting**, and **End-user-help**. See the latest score formula and grouping at Primary KPI Score.
- Expanded the content scope to cover all topic keys except /archive and /previous-versions, and moved learning path and learn module to MS Learn Report.
- The **Dynamics** and **Office** content were separated from **Content & Learning** scope, and new reports created for BAG Content (Dynamics 365, Power Platform, and Data Integration) and Microsoft365 using the same metrics as C&L.
    - C&L: https://aka.ms/contentperformancedashboard
    - BAG Content: https://aka.ms/contentperformance-bag
    - Microsoft365: https://aka.ms/contentperformance-m365
- Flipped the charts on the **Overview** page, showing the **Page Views to lower-performing articles** on the top of **Page Views to higher-performing articles** to make it more readable.
- Snapshot the monthly **Page View** to higher-performing, lower-performing articles on en-us by the dimensions defined in Overview filters to avoid the **% of PV** change from time to time when metadata gets changed.
- **Thousand Eyes Testing** traffic was excluded beginning on April 18, 2020, which caused 1.5M decrease in monthly page views for hub/landing pages.
- Added **Total Page Views** on the **By Service/By Product* page so we can leverage the volume of Page Views when evaluating the **% of PV to higher-performing articles** by Service or Product.
- Added **GitHubNewIssues** on the left of **GitHubOpenIssuesLink** on **Documentation** and **Reference** page to know how many new issues were created in the last month before clicking the GitHubOpenIssuesLink to view the detailed issues in GitHub.
- Added **RatingVerbatims** on the left of **RatingVerbatims** link on **Documentation** and **Reference** page to know how many **Verbatim** were created in the last month before clicking the **RatingVerbatims** and to view the detailed verbatim in Azure Data Explorer.
- Updated the **Canonical Functions** for Content Performance dashboard V2 data. Some query examples

were updated at How to use Content Performance Data.

January 2020

- Some new canonical functions are available that you can use to drill down for a specific article
  - ReferralsDrilldownByTopicKey: Drill down to the most important performance KPIs by referrer name, referrer domain, and referrer URL
  - HubLandingPageDrilldown: Drill down the content performance for the page opened for a specific hub-landing page
  - TopBeforeAndAfterPagesDrilldown: Provide a customer journey for the most popular pages viewed before and after a certain page, immediately, in the same session, and within last N days respectively.
- Article Assessment (Beta) is available as a new page on the Content Performance Dashboard so you can view the impact of your published changes for a single article. You specify the page, the publish date, and the weeks before and after publish for which you want to see the data. Then, when you run the report, you get the results to see whether your changes affected change in traffic, bounce, click-through, etc.

December 2019

- On the **Documentation** page and the **Reference** page:
  - Added GitHub issues links. When you click the link in the **GitHubOpenIssuesLink** column, another browser tab opens GitHub.com and displays the list of open GitHub issues for the article.
  - Added verbatim links. When you click the link in the **RatingVerbatim** column, another browser tab opens Azure Data Explorer and automatically runs a query to retrieve all new verbatim that were logged during the reporting period.
- The report UI has been cleaned up to remove redundant or confusing items and to highlight key resources for providing feedback and learning to use the report.
- The article type selector buttons have been moved to a standard filter on the right panel, and the tiny hard-to-decipher charts on the main page have been removed.
- The PrimaryKPIScore metric article in the contributor guide has been updated to list the article types covered by the content performance model. More information has been added to document how the model groups certain content types together based on factors beyond the ms.topic value.

November 2019

- Added the **Performance by Service/Product** page to provide a ranking of services and products by the percentage of page views that go to the higher performing content. You can expand the plus signs to view subservice and technology performance as well.
- The weekly and monthly trend reporting has been changed to use Azure Data Explorer (the web version of Kusto). Mac users and anyone without Kusto installed can now use the Kusto queries for content performance.
- Instead of having to follow 12 manual steps to obtain referrer details, you can now click an icon in the ReferralsDrilldown column of the Documentation, Learn, and Reference tabs to obtain the referrer information. The function behind this feature loads Azure Data Explorer, populates the query, connects to the right database, all in one click. More info

October 2019

The **reference-architecture** topic type is now available as the **Ref. Arch** filter button on the Documentation page.

- You can now filter by employee departments on all report pages using the **Department** filter.

- (BETA) Trend reporting for a given topic has been partly automated. You can now click the **Weekly/Monthly** button to start Kusto and get the trend lines for key metrics.

Known issues

- **High bounce rates** (Feb 2021) - In February, some engagement events (dwell) on pages were under-counted, the result of which is over-counted bounce rates. This data gap impacts all of docs.microsoft.com.
  - Period of data impacted: 2/9-3/3
  - Background: A frontend docs bug introduced on 2/9 prevented the following custom telemetry events from firing: page-focus-changed, page-visibility-changed, select-value-changed, print, secondary-content-scroll, unload
- **Redirected pages** (Nov 2020) - Service and Product metadata values weren't preserved when redirect_document_id was set to false in the redirect.json file prior to November 2020. The code bug in the Metadata Store was fixed so these values are preserved moving forward.
- **Historical data on the Overview page** (July 2020) - We needed to refresh the historical data for both June and July 2020 based on the following reasons:
  - Root cause: The .md version of hub and landing pages, which are no longer live, were being calculated as part of the historical snapshot query for the Overview page (details are in the Docs Build v3 migration changes. The same page is using two topic IDs (one for .yml version and one for .md version). Both topic IDs are showing traffic, and the expected behavior is that the not-live version (.md file) shouldn't have any traffic. Although the traffic is small, it impacted the overall ranking.
  - Impact: Overview tab-June and July only. Type group most impacted is Hub & landing. This update didn't impact the monthly data on Docs and Reference report pages.
  - Short-term fix: Update snapshot query and refresh June and July data only on the Overview page.
  - Long-term fix: CGA looking into root cause in the Kusto page view table.
- **Duplicate records - one article with multiple topic keys**: In the content performance dashboard data, duplicate entries may exist for individual article URLs which is a known issue in the CGA data source and will be addressed in the new unified topic metadata store.
- **Public/private repository pairs**: In the RepoName filter, both public and private repositories are listed for repository pairs such as azure-docs and azure-docs-pr. If both appear for your repository, select them both to get a full view of the content performance data for that repository pair.
- **Test-tool traffic**: Some hub-landing pages contain traffic caused by the testing tool, the direct traffic bounce rate is super high. You can use “ReferralsDrilldown” function to evaluate their performance before we filter out the testing traffic.

> **NOTE**
>
> The **Thousand Eyes** testing introduced around 1.5M Page Views monthly on 30+ docs hub-landing, index pages were excluded from April/18/2020.

- **Data differences between Content Performance and Documentation dashboards**: You might notice differences in data points between the Content Performance Dashboard and the Documentation Dashboard. The main difference is that the documentation dashboard includes data for all locales, while the content performance dashboard contains only EN-US data. Additionally, there are differences in how some of the data is calculated. For example, bounce is measured differently. For the Documentation Dashboard, we use the industry standard definition for bounce rate: the % of sessions on docs with only one page view, and time on site is less than 5 seconds. There is no session concept for page-level performance measurement. So, in the Content Performance Dashboard, we measure bounce rate based on page view behavior instead.

Connect to CGA Kusto data clusters

5/7/2021 • 2 minutes to read

Use Kusto to deep-dive into content data metrics for non-standard reporting needs. In this article, you learn how to request access to the Kusto data clusters and connect to them using client applications.

You should check to see if your reporting needs can be met by existing dashboards before querying Kusto directly. If your needs aren't met, help us improve by submitting a dashboard request.

NOTE

Data clusters are supported by CGA (Cloud+AI Customer Growth Analytics). For more information, see CGA's website.

Request access to the Kusto data clusters

NOTE

As of April 27th, 2021, the **cgadataout** cluster has been replaced by the **followercgadataout** cluster. Users with **cgadataout** access will automatically have access to the replacement cluster. Request access.

To view content health and engagement metrics, you must request access to one of two Kusto data clusters. Which cluster you choose depends on your organization. If you're in Developer Relations (DevRel), you can request access to **CGA Kusto Public** and **followercgadataout**. If you are not in DevRel, you can only request access to **CGA Kusto Public**.

Select the links in the following table to submit a MyAccess request. When creating the request, select **reader-only** for your permissions level.

DEV RELATIONS ORGANIZATION	NON DEV RELATION USER
CGA Kusto Public (cgadatamall.westus.kusto.windows.net)	CGA Kusto Public (cgadatamall.westus.kusto.windows.net)
followercgadataout (followercgadataout.westus.kusto.windows.net)	

Some reports, like the Page-level trends & drill down dashboard, require you to join an additional data cluster.

CLUSTER NAME	DESCRIPTION
1es CloudMine cluster (https://1es.kusto.windows.net/)	CloudMine aggregates data from diverse sources including Azure DevOps, HeadTrax, Stack Overflow, and GitHub. Request to join the **CloudMine-Data reader role** to view GitHub information in the Page-level trends & drill down dashboard.

Public cluster vs followercgadataout

The CGA Public (cgadatamall) cluster is intended for exploratory queries. It is not scaled for computationally expensive queries or production workloads.

The followercgadataout cluster is higher capacity, but not infinite capacity. There are no guarantees on performance.

Regardless of which cluster you use, we recommend limiting the output using constraints like date ranges and content filters to improve performance and avoid memory limits.

Connect to a Kusto cluster

You can connect to Kusto using many different applications. In this section, you learn how to connect using Kusto.Explorer (Windows only) and Data Explorer (web browser).

Before you attempt to connect to the Kusto cluster, make sure you have read-access to the clusters. For more information, see Request access to the Kusto data clusters

Connect with Azure.Explorer

Azure.Explorer offers a rich desktop experience for Windows users to explore data in the Kusto Query Language. For more information on installation and setup, see Kusto.Explorer installation and user interface.

After you install Azure.Explorer connect to the Kusto cluster:

1. Select the **Connection** tab.
2. Select **Add connection**.
3. For **Cluster connection**, enter your cluster address.

   For example, `cgadatamall.westus.kusto.windows.net` or `followercgadataout.kusto.windows.net`

4. For **Security**, select `AAD -Federated`.

Connect with Data Explorer

Data explorer lets you query Kusto clusters directly from your browser.

1. In your browser, enter the address of the cluster you want to connect to.

   For example, `cgadatamall.westus.kusto.windows.net` or `followercgadataout.kusto.windows.net`

2. Your browser will redirect you to dataexplorer.azure.com.
3. After you authenticate with your Microsoft credentials, you will be taken to the interactive querying tool.

Next steps

In this article, you learned how to connect to the CGA Kusto clusters. To learn how to query docs content data, see How to use content performance canonical functions.

How to use content performance canonical functions?

3/5/2021 • 5 minutes to read

In this article, you learn about canonical functions that provide answers to commonly asked data questions. Canonical functions output data that has been processed using CGA's business logic and can reduce your reliance on manual data processing.

In this article you learn how to use the following canonical functions:

FUNCTION NAME	DESCRIPTION
ContentKPI functions	Aggregated content performance data for each topic
ReferralsDrilldown	Understand how users came to a given topic and the quality of those referrals
HubLandingPageDrilldown	Understand where users go immediately after visiting a hub or landing page
TopBeforeAndAfterPagesDrilldown	Understand other docs that a user went to before or after visiting a given page (user journey)

WARNING
CGA doesn't guarantee performance in their Kusto clusters. Permission requests to the cgadataout.kusto.windows.net cluster may be denied due to cluster management policies. CGA recommends using the content performance dashboard before querying Kusto directly. Create a dashboard request for needs that aren't served by existing dashboards.

Prerequisites

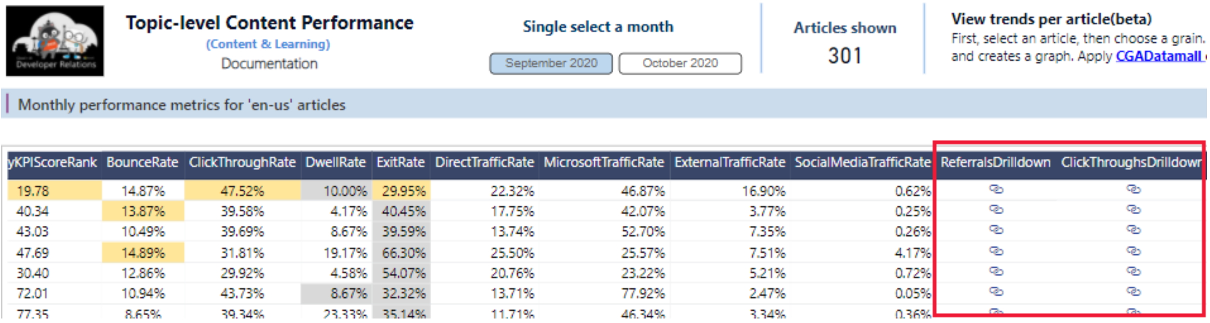
You must have access to the **cgadataout** or **cgadatamall** cluster. For more information, see Connect to CGA Kusto data clusters.

How to use the canonical functions

Canonical functions are stored in the **Publish** database of both the **cgadataout** and **cgadatamall** clusters. If you have read-access to the clusters, you have access to the functions.

Use the canonical functions in the following ways:

- Select the **Try-it** link above the code snippets in this article to run the query immediately in your browser.
- Copy the code snippets in a query client like Kusto.Explorer and run the query.
- Select the links in the Content Performance Dashboard for drill down queries.



You should update parameters and include filters to narrow the result-set. This improves performance and helps you avoid memory limits and timeouts.

ContentKPI functions

ContentKPI functions join the TopicMetadata table and fact tables. They aggregate the results by `TopicKey` or `ContentID`. You can set the temporal granularity to `Weekly` or `Monthly`. The functions return topic metadata and all the metrics reported in the Content Performance Dashboard.

The functions are available at cluster('cgadataout.kusto.windows.net').database('Publish'). And have the following names:

- ContentKPIByTopicKeyMonthlyV2
- ContentKPIByTopicKeyWeeklyV2
- ContentKPIByContentIdMonthlyV2
- ContentKPIByContentIdWeeklyV2

Example: Top 100 lower-performing articles by Page Views from a docset

Try-it

```
cluster('cgadataout.kusto.windows.net').database('Publish').ContentKPIByTopicKeyMonthlyV2
| where RepoName == 'azure-docs-pr'
| where Locale =='en-us'
| where Date == datetime(2020-04-01)
| invoke GetPrimaryKPIScoreV2()
| as T
| join kind=inner
(
    T
    | summarize Score25Pctile = percentile(PrimaryKPIScore, 25) by PageType, Date
) on PageType, Date
| extend IsLowerPerforming = PrimaryKPIScore < Score25Pctile
| where IsLowerPerforming
| order by PageViews desc
| extend Rank = row_number(1, prev(Date)!=Date)
| where Rank <= 100
| project    Date,
            TimeGrain,
            TopicKey,
            TopicType,
            Title,
            LiveUrl,
            PageViews,
            BounceRate,
            ClickThroughRate,
            DwellRate,
            ExitRate,
            OrganicSearchRate,
            DirectTrafficRate,
            PaidSearchRate,
            ExternalTrafficRate,
            MicrosoftTrafficRate,
```

```
        MicrosoftTrafficRate,
        SocialMediaTrafficRate,
        CSATHelpfulRate,
        CSATResponseRate,
        UserEngagementRate,
        StartTutorialRate,
        AvgCopyTryEngagement,
        DeployClickRate,
        Freshness,
        LinesChanged,
        LastCommitDateTime,
        GitHubOpenIssueCount,
        GitHubTotalIssueCount,
        DaysSinceLastReview,
        LastReviewed,
        Locale,
        RepoName,
        TranslationType,
        DevLang,
        MSProd,
        MSTechnology,
        MSService,
        MSSubService,
        Custom,
        Author,
        MSAuthor,
        Manager,
        Bounces,
        CSATHelpfulRatings,
        CSATTotalResponses,
        GitHubNewIssueCount,
        ClickThroughs,
        CopyTryClicks,
        EstimatedSecondsToRead,
        Exits,
        FirstSteps,
        MedianDwellTimeSeconds,
        OrganicSearchReferrals,
        TutorialLinkClicks,
        Visitors,
        RatingVerbatims,
        PrimaryKPIScore,
        PageType
```

ReferralsDrilldownByTopicKey

The ReferralsDrilldownByTopicKey function drills down on content metrics based on where a user came from. It shows how different users interacted with page depending on how they got to the page.

Users are grouped by `ReferrerType`, `ReferrerDomain`, and `ReferrerUrl` for a given `TopicKey` by weekly or monthly. The function returns all referrals with more than 10 page views.

Parameters

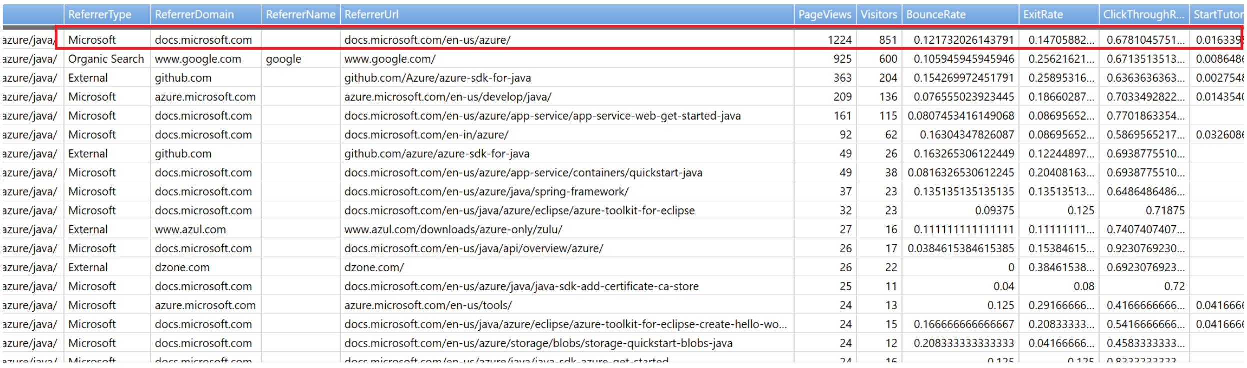
Update the following parameters to narrow your query:

- TargetTopicKey: *TopicKey* for a target page, you can copy this from the Content Performance Dashboard
- StartDate, EndDate: *datetime*, the start and end date you want to track. For example `datetime(2019-10-01)`, `datetime(2019-12-1)`
- TimeGrain: *string*, `month` or `week`

Example: Evaluate referral performance

Here is an example to drill down on `https://docs.microsoft.com/en-us/azure/java/` and exclude `Direct Traffic` referrals. Sort by `PageViews` and evaluate the performance for different referrer links.

Here is a screenshot for the data return by the function, you can try-it and check more KPIs.



	ReferrerType	ReferrerDomain	ReferrerName	ReferrerUrl	PageViews	Visitors	BounceRate	ExitRate	ClickThroughR...	StartTutor
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/	1224	851	0.121732026143791	0.14705882...	0.6781045751...	0.016339
azure/java/	Organic Search	www.google.com	google	www.google.com/	925	600	0.105945945945946	0.25621621...	0.6713513513...	0.008648
azure/java/	External	github.com		github.com/Azure/azure-sdk-for-java	363	204	0.154269972451791	0.25895316...	0.6363636363...	0.002754
azure/java/	Microsoft	azure.microsoft.com		azure.microsoft.com/en-us/develop/java/	209	136	0.076555023923445	0.18660287...	0.7033492822...	0.014354
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/app-service/app-service-web-get-started-java	161	115	0.0807453416149068	0.08695652...	0.7701863354...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-in/azure/	92	62	0.16304347826087	0.08695652...	0.5869565217...	0.032608
azure/java/	External	github.com		github.com/azure/azure-sdk-for-java	49	26	0.163265306122449	0.12244897...	0.6938775510...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/app-service/containers/quickstart-java	49	38	0.0816326530612245	0.20408163...	0.6938775510...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/java/spring-framework/	37	23	0.135135135135135	0.13513513...	0.6486486486...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/java/azure/eclipse/azure-toolkit-for-eclipse	32	23	0.09375	0.125	0.71875	
azure/java/	External	www.azul.com		www.azul.com/downloads/azure-only/zulu/	27	16	0.111111111111111	0.11111111...	0.7407407407...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/java/api/overview/azure/	26	17	0.0384615384615385	0.15384615...	0.9230769230...	
azure/java/	External	dzone.com		dzone.com/	26	22	0	0.38461538...	0.6923076923...	
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/java/java-sdk-add-certificate-ca-store	25	11	0.04	0.08	0.72	
azure/java/	Microsoft	azure.microsoft.com		azure.microsoft.com/en-us/tools/	24	13	0.125	0.29166666...	0.4166666666...	0.041666
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/java/azure/eclipse/azure-toolkit-for-eclipse-create-hello-wo...	24	15	0.166666666666667	0.20833333...	0.5416666666...	0.041666
azure/java/	Microsoft	docs.microsoft.com		docs.microsoft.com/en-us/azure/storage/blobs/storage-quickstart-blobs-java	24	12	0.208333333333333	0.04166666...	0.4583333333...	

Try-it

```
cluster('cgadataout.kusto.windows.net').database('Publish').ReferralsDrilldownByTopicKey(
    TargetTopicKey = '0981839c-30f2-6a8a-2b38-2078eb69b302',
    StartDate = datetime(2019-10-01),
    EndDate = datetime(2019-10-31),
    TimeGrain='month')
 | where ReferrerType <> 'Direct Traffic'
 | order by PageViews desc
```

HubLandingPageDrilldown

The HubLandingPageDrill down shows the next page that a user visited immediately after visiting a hub landing page.

Use this query to assess link quality on hub-landing pages.

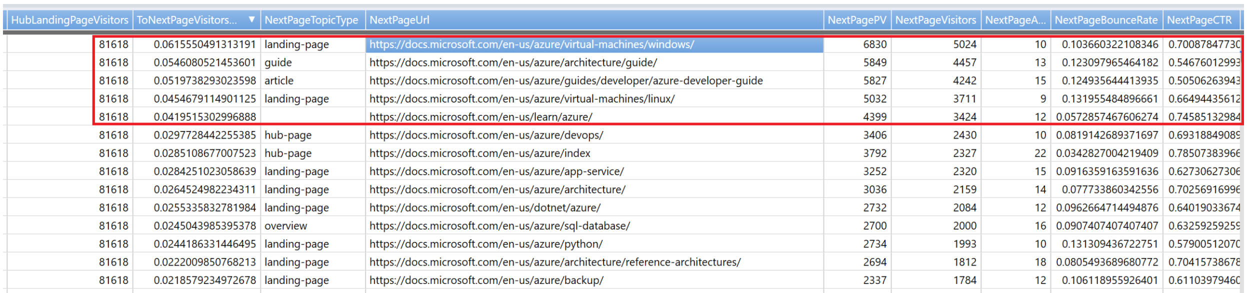
Parameters

- HubLandingPageUrl: *Url*, hub-landing page url, for example `https://docs.microsoft.com/en-us/azure/`
- StartDate: *datetime*, such as datetime(2019-10-01)
- EndDate: *datetime*, such as datetime(2019-10-31)

Example

The following example drills down on the Azure hub-landing page 'https://docs.microsoft.com/en-us/azure/'. We can find:

- The most popular links on the Azure Home page
- The bounce rate, view length, and CTR on each page opened



HubLandingPageVisitors	ToNextPageVisitors...	NextPageTopicType	NextPageUrl	NextPagePV	NextPageVisitors	NextPageA...	NextPageBounceRate	NextPageCTR
81618	0.0615550491313191	landing-page	https://docs.microsoft.com/en-us/azure/virtual-machines/windows/	6830	5024	10	0.103660322108346	0.7008784773
81618	0.0546080521453601	guide	https://docs.microsoft.com/en-us/azure/architecture/guide/	5849	4457	13	0.123097965464182	0.5467601299
81618	0.0519738293023598	article	https://docs.microsoft.com/en-us/azure/guides/developer/azure-developer-guide	5827	4242	15	0.124935644413935	0.5050626394
81618	0.0454679114901125	landing-page	https://docs.microsoft.com/en-us/azure/virtual-machines/linux/	5032	3711	9	0.131955484896661	0.6649443561
81618	0.0419515302996888		https://docs.microsoft.com/en-us/learn/azure/	4399	3424	12	0.0572857467606274	0.7458513298
81618	0.0297728442255385	hub-page	https://docs.microsoft.com/en-us/azure/devops/	3406	2430	10	0.0819142689371697	0.6931884908
81618	0.0285108677007523	hub-page	https://docs.microsoft.com/en-us/azure/index	3792	2327	22	0.0342827004219409	0.7850738396
81618	0.0284251023058639	landing-page	https://docs.microsoft.com/en-us/azure/app-service/	3252	2320	15	0.0916359163591636	0.6273062730
81618	0.026452498223431	landing-page	https://docs.microsoft.com/en-us/azure/architecture/	3036	2159	14	0.077733860342556	0.7025691699
81618	0.0255335832781984	landing-page	https://docs.microsoft.com/en-us/dotnet/azure/	2732	2084	12	0.0962664714494876	0.6401903367
81618	0.0245043985395378	overview	https://docs.microsoft.com/en-us/azure/sql-database/	2700	2000	16	0.0907407407407407	0.6325925925
81618	0.0244186331446495	landing-page	https://docs.microsoft.com/en-us/azure/python/	2734	1993	10	0.131309436722751	0.5790051207
81618	0.0222009850768213	landing-page	https://docs.microsoft.com/en-us/azure/architecture/reference-architectures/	2694	1812	18	0.0805493689680772	0.7041573867
81618	0.0218579234972678	landing-page	https://docs.microsoft.com/en-us/azure/backup/	2337	1784	12	0.106118955926401	0.6110397946

Try-it

```
cluster('cgadataout.kusto.windows.net').database('Publish').HubLandingPageDrilldown(
    HubLandingPageUrl = 'https://docs.microsoft.com/en-us/azure/',
    StartDate = datetime(2019-10-01),
    EndDate = datetime(2019-10-31))
```

TopBeforeAndAfterPagesDrilldown

TopBeforeAndAfterPagesDrilldown shows a customer journey from the pages viewed before and after viewing a target page. This function returns results for 3 different time periods:

- ImmediatelyBefore/After: How many visitors viewed a given page immediately before and after visiting the target page. Pages with high visits indicate a strong relationship with the target page.
- SameSessionsBefore/After: How many visitors viewed a given page during the same session. Pages with high visits are correlated with the target page, since many users viewed them in the same session.
- NDaysBefore/After: How many visitors viewed a given page within `N days` before, or after, viewing the target page. This data suggests that these pages may be related.

Parameters

- TargetUrl: *Url*, the page you want to evaluate, such as `https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-your-first-function-visual-studio`
- StartDate, EndDate: *datetime*, such as datetime(2019-10-01), datetime(2019-10-31)
- DaysBeforeAndAfter: *int*, N days before or after, for example 30 here means last 30 days before, and next 30 days after viewing current page
- TopNUrls: *int*, the Top page with most visitors, 100 means top 100 pages by its visitors

Example

This example drills down on the following page `https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-your-first-function-visual-studio`.

TopBeforeOrAfterUrls	VisitorsImmediatelyBefore ▼	VisitorsImmediatelyAfter	VisitorsSameSessionBefore	VisitorsSameSessionAfter	VisitorsNDaysBefore	VisitorsNDaysAfter
https://docs.microsoft.com/en-us/azure/azure-functions/	2337	115	2698	746	3484	1819
https://www.google.com/	1655					
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-azur...	546	10	833	356	1536	956
https://docs.microsoft.com/en-us/azure/azure-functions/functions-overview	421	123	933	464	1669	1077
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-func...	347	209	436	519	769	813
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-first-func...	327	216	431	385	663	645
https://docs.microsoft.com/en-us/azure/azure-functions/functions-develop-vs	279	321	770	722	1555	1679
https://docs.microsoft.com/en-us/azure/azure-functions/index	189		208		278	
https://docs.microsoft.com/en-us/dotnet/azure/	177		196	89	381	267
https://docs.microsoft.com/en-us/azure/azure-functions/functions-scale	165	79	297	174	789	837
https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-function-...	140		228	79	463	289
https://docs.microsoft.com/en-us/azure/azure-functions/functions-add-output-bin...	137	690	273	652	473	977

Try-it

```
cluster('cgadataout.kusto.windows.net').database('Publish').TopBeforeAndAfterPagesDrilldown(
    TargetUrl = 'https://docs.microsoft.com/en-us/azure/azure-functions/functions-create-your-first-
function-visual-studio',
    StartDate = datetime(2019-10-01),
    EndDate = datetime(2019-10-31),
    DaysBeforeAndAfter = 30,
    TopNUrls = 100)
```

Contacts

- If you have any question regarding there functions, contact Edison Dai. You can create a ticket using the dashboard data requests process. @mention msmbaldwin in the work item for awareness.

Next steps

In this article, you learned how to run canonical functions in CGA's Kusto clusters.

Learn how to write your own queries, by looking at example queries submitted by our content-community. For more information, see Community Kusto queries.

Community Kusto queries

5/10/2021 • 4 minutes to read

In this article, you learn about community-submitted Kusto queries that other contributors have found useful.

These queries are not officially supported and are intended to help you understand different Kusto queries. For a list of supported queries, see How to use canonical functions.

Prerequisites

You must have access to the **cgadataout** or **cgadatamall** cluster. For more information, see Connect to CGA Kusto data clusters.

Queries

Trends over time for a particular URL

Run the query in Lens Explorer

```
let startDate = datetime(2020-1-1);
let endDate = datetime(2020-5-1);
cluster("Cgadataout").database("CustomerTouchPoint").PageView
| where StartDateTime between(startDate .. endDate)
| where Url == "https://docs.microsoft.com/en-us/azure/architecture/"
| summarize Visits=dcount(enrich_session_id), PageViews = count(), FromSearch = countif(ReferrerType has
'Search'), NotSearch = countif(ReferrerType !has 'Search'), Bounces = countif(IsBounce == 1) by
bin(StartDateTime, 1d)
| render timechart
```

Low-to-zero-visits query

Run the query in Lens Explorer

You can change the matchURL variable to narrow down to a particular docset. Start and end dates determine the time window. The results are ordered by PageViews descending, so you can see what got no views in a given time range.

```
let urlPattern="https://docs.microsoft.com/en-us/sql/";
let allLivePages =
cluster('cgadataout.kusto.windows.net').database("CustomerTouchPoint").TopicMetadata
| where LiveUrl contains urlPattern
| where IsLive == 1
| where isempty(RedirectUrl)
| project TopicKey, TopicType, Title, LiveUrl;
let viewedPages =
    cluster('cgadataout.kusto.windows.net').database('Publish').ContentKPIByTopicKeyMonthlyV2
    | where LiveUrl contains urlPattern
    | where Date > ago(60d)
    | project TopicKey, LiveUrl, PageViews;
allLivePages
| join kind = leftanti (viewedPages) on TopicKey
```

Page views and visits for specific services in last N days

```
cluster("Cgadataout").database("CustomerTouchPoint").PageView
| where Site == "docs.microsoft.com" and Locale == "en-us"
| where StartDateTime >= ago(90d)
| where MSService contains "sentinel"
| summarize PageViews = count(), Visits = dcount(enrich_session_id) by Url
```

All content for Content & Learning published between two dates

Use these queries to find all content published for tech docs, Learn, and the architecture center.

The output will contain three columns - all articles, learn, and architecture. To determine the tech doc totals, add a column to calculate the total using this formula: `=b2-(c2+d2)`.

For reporting, we usually separate reference from conceptual. For standard reporting, the following types are considered reference:

- managed-reference
- reference
- language-reference
- error-reference

Updated articles: all articles that existed prior to the first date specified in the query and that were updated between the specified dates.

```
cluster("Cgadataout").database("CustomerTouchPoint").TopicMetadata
| where Site == 'docs.microsoft.com' and IsLive == '1' and Locale=='en-us'
| where TopicType != 'language-reference' and TopicType != 'managed-reference' and TopicType != 'reference'
and TopicType != 'error-reference' and TopicType !='page-not-found' and TopicType !='portal' and TopicType
!='struct' and TopicType !='interface' and TopicType !='enum' and TopicType !='callback' and TopicType
!='function' and TopicType !='class' and TopicType !='method'and TopicType !='ioctl' and TopicType
!='structure' and TopicType !='enumeration'
| where startofday(LastReviewed) between(datetime(2019-02-01) .. datetime(2019-02-28))
| where startofday(FirstPublishDateTime) !between(datetime(2019-02-01) .. datetime(2019-02-28))
| summarize updatedTotal=dcount(TopicKey),
    updatedLearn=dcountif(TopicKey, LiveUrl matches regex "^http?s://docs.microsoft.com/([^/]*)/learn/"),
    updatedAAC=dcountif(TopicKey, LiveUrl matches regex "^http?
s://docs.microsoft.com/([^/]*)/azure/architecture/")
    by Product,Service
```

New articles: all articles published for the first time between the specified dates.

```
cluster("Cgadataout").database("CustomerTouchPoint").TopicMetadata
| where Site == 'docs.microsoft.com' and IsLive == '1' and Locale=='en-us'
| where TopicType != 'language-reference' and TopicType != 'managed-reference' and TopicType != 'reference'
and TopicType !='page-not-found' and TopicType !='portal' and TopicType !='struct' and TopicType
!='interface' and TopicType !='enum' and TopicType !='callback' and TopicType !='function' and TopicType
!='class' and TopicType !='method'and TopicType !='ioctl' and TopicType !='structure' and TopicType
!='enumeration'
| where startofday(FirstPublishDateTime) between(datetime(2019-02-01) .. datetime(2019-02-28))
| summarize updatedTotal=dcount(TopicKey),
    updatedLearn=dcountif(TopicKey, LiveUrl matches regex "^http?s://docs.microsoft.com/([^/]*)/learn/"),
    updatedAAC=dcountif(TopicKey, LiveUrl matches regex "^http?
s://docs.microsoft.com/([^/]*)/azure/architecture/")
    by Product,Service
```

Courtesy of: Presley/Jian

All abandoned articles in Content & Learning content sets

```
let startDate = ago(30d); //Start date of page view window
let endDate = now(); // End date of page view window
let TopicsTable = cluster("Cgadataout").database("CustomerTouchPoint").TopicMetadata //scopes to only
content owned by C+L
| where Site == 'docs.microsoft.com' and IsLive == '1' and Locale=='en-us' //scopes to docs site and en-us
content
| where LastReviewed < ago(90d) //set datetime to 90 or 120 days ago to list all stale and/or abandoned
content
| where TopicType !in ("managed-reference", "reference") //excludes most reference content from this query
| where Pillar == "Azure"; //choose the pillar to scope results to avoid exceeding the query record limit
let cachedPageViewTable = materialize(cluster('Cgadataout').database("CustomerTouchPoint").PageView
| where Site == "docs.microsoft.com" and Locale == "en-us"
| where IsMSInternalTraffic != true and enrich_rip_isp != "microsoft corporation"
| where StartDateTime between(startDate .. endDate)
| project StartDateTime, TopicKey, enrich_session_id, IsBounce, PageViewId, Title, Site, Locale, TopicType,
Url, IsMSInternalTraffic, enrich_rip_isp, LinkClickEvents, LinkClickEventsJson, VisitorId);
let metricsByTopicKey = cachedPageViewTable
| summarize PageViews = count(), Visits = dcount(enrich_session_id),  Bounces = countif(IsBounce == true)
//, Title = any(Title), TopicType = any(TopicType), Url = any(Url)
by TopicKey
| extend ['Bounce %'] = round(Bounces * 100.0 / PageViews, 1);
let output = TopicsTable
| join kind = leftouter (
metricsByTopicKey
) on TopicKey
| project
  Locale
, Service
, Author
, TopicType
, Title
, LastReviewed = format_datetime(LastReviewed, 'yyyy-MM-dd')
, LiveUrl
, Visits = iff(isnotempty(Visits), Visits, 0)
, PageViews
, Bounces
, ['Bounce %'] = strcat(tostring(iff(isnan(['Bounce %']), todouble(0), ['Bounce %'])), '%')
| sort by Visits desc;
output
```

Next steps

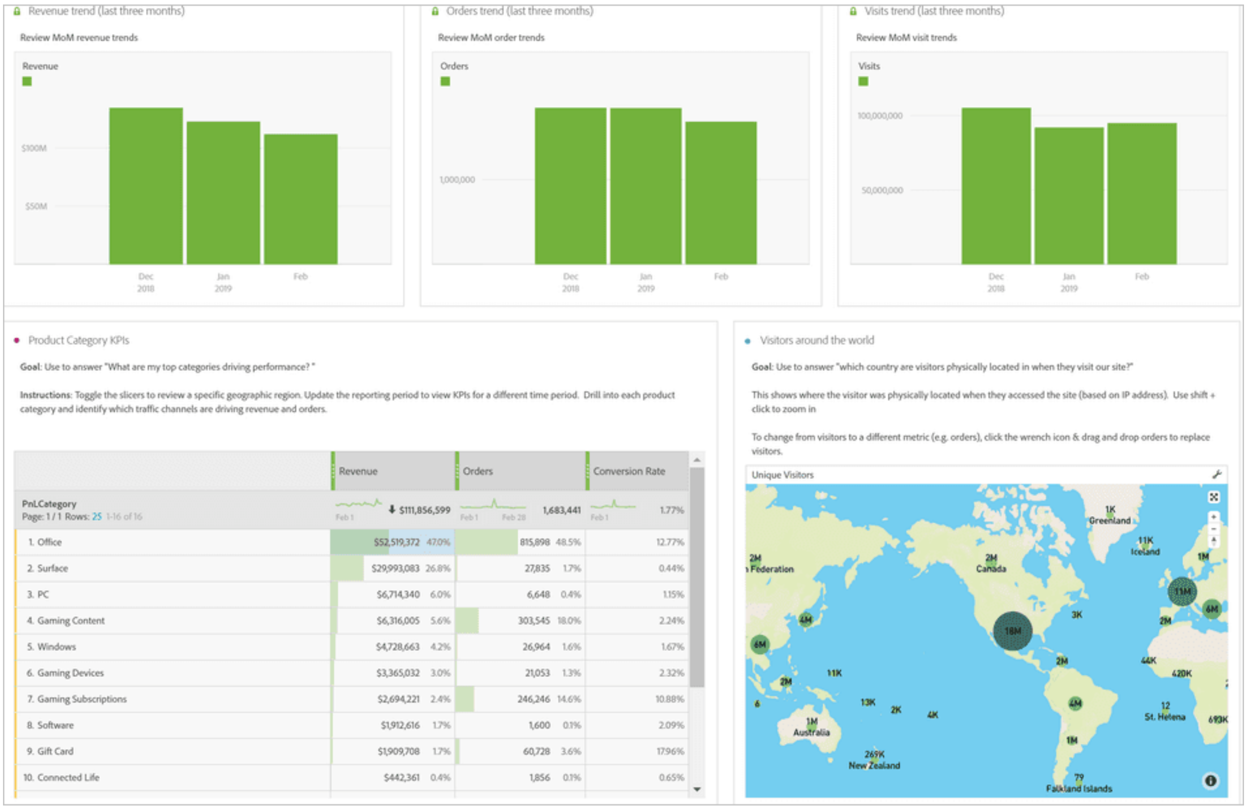
For more sample queries using the Kusto Data Out, see CGA Kusto Cluster "CGADataOut" Data Models.

Adobe Analytics

4/23/2021 • 2 minutes to read

Contributors can use the Adobe Analytics platform to explore web analytics for Microsoft domains. This is available to all Microsoft employees, free of charge.

For more information, see Get started - Microsoft Adobe Analytics.



Adobe Analytics vs Kusto and PowerBI

Adobe Analytics is an alternative analytics platform to Kusto and PowerBI that contributors can use to generate reports and explore web metrics.

Adobe Analytics uses the same JSLL (Java Script Logging Library) tagging system that feeds into the CGA reporting Kusto clusters that provide the data for most official C&L PowerBI reports. As a result, the data across platforms should be comparable, though data discrepancies may appear due to differences in data processing.

Metadata and global markets

Metadata information such as ms.author, ms.service, ms.technology, etc. are not available in Adobe Analytics. These metadata values **are** available in Kusto, and their downstream PowerBI reports, due to additional processing that occurs in the Kusto cluster data pipeline. As a result, Adobe Analytics queries cannot use metadata values to filter datasets, you must rely on URL patterns in Adobe Analytics.

Many official C&L PowerBI reports only display metrics from the `en-us` lang-locale. Although global data is available in the Kusto clusters, they require custom Kusto queries to. You can use Adobe Analytics to query data on global markets without learning the Kusto Querying Language.

Low barrier to entry

Unlike Kusto, you don't need to learn a new querying language to use Adobe Analytics. Instead, you create queries with a visual explorer to filter and transform data.

For more learning resources, see Adobe Analytics onboarding videos and guided learning.

Support

For help and support, reach out to the Adobe Analytics support alias, AskMarTech@microsoft.com.

Training and office hours are also available. For more information on dates and events, see Training & Events.

How to view and act on customer verbatim comments

3/5/2021 • 4 minutes to read

To find the verbatim comments for the articles you own, use one of these methods:

- For a single article, use the Microsoft Docs Metrics report (Preview) **Details** link to view an Azure Data Explorer dashboard filtered to the article you were looking at, and the time period you had selected: one week or one month. You can update the dates in the query to see more verbatims.
- The new preview version of the Verbatims report https://aka.ms/DocsVerbatimsReport shows verbatim comments for all content in docs. You can filter it by author, product, service, and other values, to show the respective verbatim comments. For more information, see Metrics: Verbatims report.

The Verbatim reports use an ML model that categorizes the verbatim comments. See Verbatim categories in this article for descriptions of those categories and subcategories. In those reports, the ML-determined subcategory is shown next to the verbatim. Both reports also show international verbatim comments machine translated to EN-US. The accuracy of the ML model for categorizing verbatim comments isn't 100% accurate. We have a backlog item to improve the machine-learning categorization.

What customers expect when they leave feedback

Understanding customer needs is key to building great content and helping drive adoption and customer satisfaction. Because of scale, this understanding is always limited – but we can do a lot to improve the "signal" with data in ways that are clear and actionable. One of the many channels for better understanding our customers is through customer feedback verbatims.

Customers' biggest expectation in the feedback loop is that we address the issues expressed in the verbatim. Feedback and comments can also be directly actionable for content owners, localization teams, and site engineering teams. We need to act on feedback, if possible. If not, customers will cease to give feedback. Unfortunately the verbatim comments are anonymous, so you can't respond directly to a customer comment.

Few examples of the broad categories of customer feedback verbatim comments include:

REQUEST	EXAMPLE
Request for more information	- Please explain like you would to a kid what Microsoft Authenticator does!(**Multi-Factor Authentication**) - Any tips on how to do repository authentication? Can't seem to find any docs on it. (**Service Fabric**) - How many IPs one can add to one NIC? How many NICs one can add to a machine? (**Virtual Network**)
Requests for specific fixes	- you should update the correct installation link to the sap NetWeaver library.(**Power BI**)
Requests for specific examples	- Would be nice to have an example demonstrating interop for custom struct types and methods for a library, for example, gmp, gtk etc. (**.NET Core**)

REQUEST	EXAMPLE
International experience	- WHY I SEE SQL COMMANDS TRANSLATED IN ITALIAN??? (**SQL Server**) - Please display Microsoft sites in English when I navigate to English sites, don't magically transform them to Dutch without asking for it.(**Azure Active Directory**)

Impact of verbatim fixes and responsiveness on CSAT

Analysis show that addressing an actionable verbatim on a timely manner has a direct correlation with article CSAT value. The more customer feedback verbatim we address on a timely manner, the highest the article satisfaction rate will be. The following two figures show the impact of addressing actionable verbatim on customer satisfaction. The green shade is the actual CSAT value of the article while the yellow shade shows the predicted CSAT. If all the actionable verbatim comments for an article are addressed, we'd expect to see a **2%-47.8%** increase in CSAT. The improvement depends on the count of actionable verbatim comments for the article during the week you're measuring.

Verbatim categories

Our content gets more than 1M verbatim comments a year from customers. We would need extra resources to read and address all of this feedback. To help, we use a machine learning text classification tool that autogroups verbatim comments into 22 predefined subcategories of issue types, which we have grouped into categories. The following table describes each verbatim category.

CATEGORY	SUBCATEGORY NAME	DESCRIPTION
Form or Presentation	Presentation-StyleVisuals	The user is commenting about the look of the website.
	Presentation-BrokenLink	The article contains broken links.
	Site-CannotFind	User cannot find what they are looking for.
	Site-Feature	Comment about the website.
	Site-FeedbackControl	Comment about the feedback control on the website.
Content	Content-Accuracy	The article content is inaccurate.
	Content-BadExample	The example is not working or inappropriate.
	Content-Incomplete	The content is incomplete.
	Content-NeedExample	The article needs an example.
Localization	Loc-BadMT	The machine translation is inaccurate.
	Loc-HelpExperience	The localized content is hard to use.

CATEGORY	SUBCATEGORY NAME	DESCRIPTION
	Loc-NeedsLoc	The topic needs translation.
	Loc-Other	Miscellaneous localization issues that do not fit any other subcategory.
	Loc-PrefersENU	The user prefers English language content.
	Loc-Translation	The translation is inaccurate.
Product	Product-Experience	The comment is about the product user experience.
	Product-International	The comment is about the product from a foreign perspective.
	Product-Link	A download link is broken.
	Product-Sentiment	General product feedback.
	Product-SubscriptionSupport	Feedback about the subscription model or usability.
Kudos	Kudos	The article was helpful.
Other	Other	The comment does not fit any of the other subcategories.
Rant	Rant	Offensive comment, spam, or other remarks that are not actionable.

Next steps

Learn more about the preview version of the Verbatims report.

Have feedback about the preview version of the Verbatims report? Please enter it in this Verbatims Report Feedback Excel file.

Request enhancements to data, dashboards, or queries

11/2/2020 • 3 minutes to read

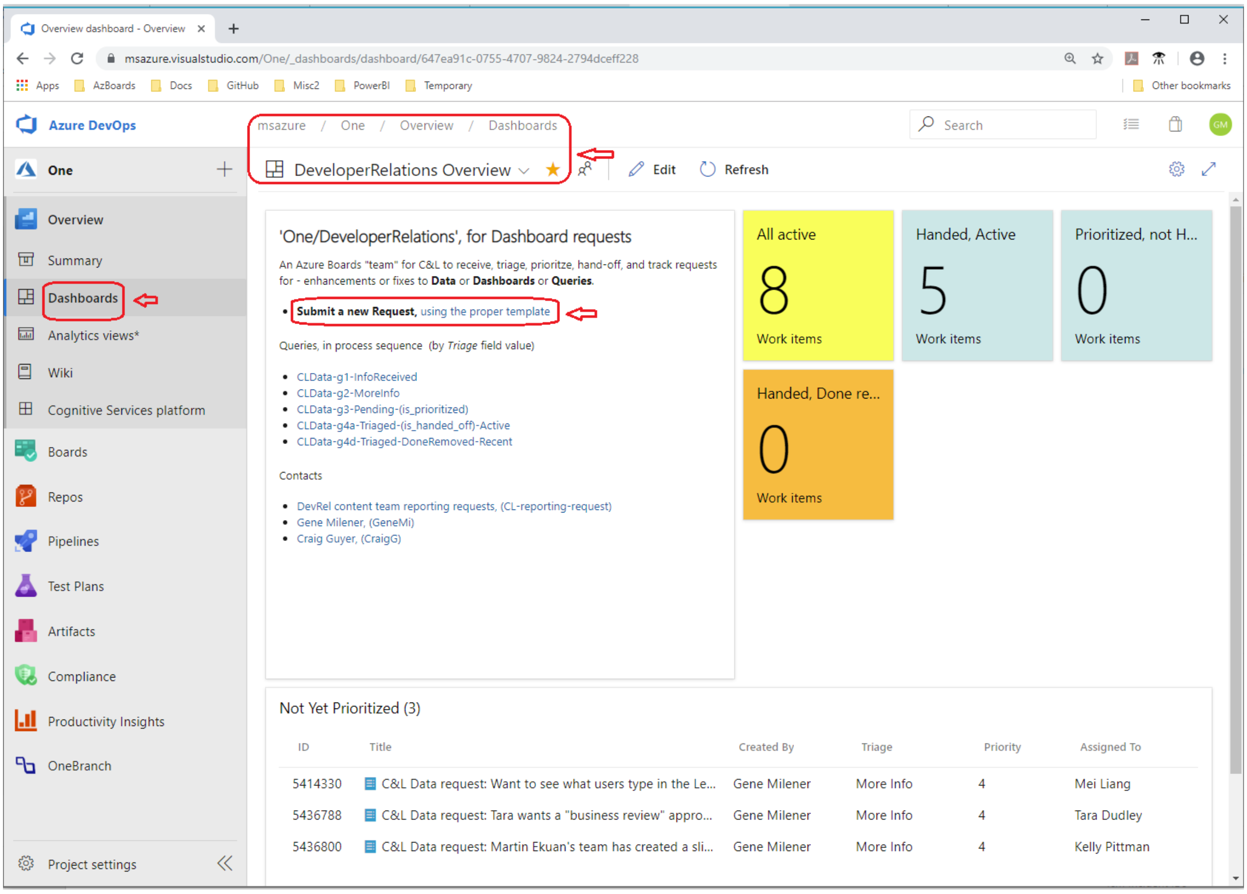
Anyone in DevRel can request enhancements or updates to the reports and dashboards that are maintained by the DevRel Analytics team. To submit a request, go to https://aka.ms/dashboard-requests. Requests can target the following items:

- **Dashboards**: only those dashboards maintained by the Analytics team or CGA.
- **Kusto queries**: to populate any dashboard.
- **Data**: the data available to Kusto queries.

The Content Data Advocacy Group (CDAG) triages the requests. Approved requests are implemented by the DevRel Analytics team or CGA (Customer Growth Analytics).

Use our dashboard

To enter your idea, or to view queries for submitted requests, go to https://aka.ms/dashboard-requests. This Azure DevOps dashboard is at the hierarchy **msazure / One / Overview / Dashboards**:



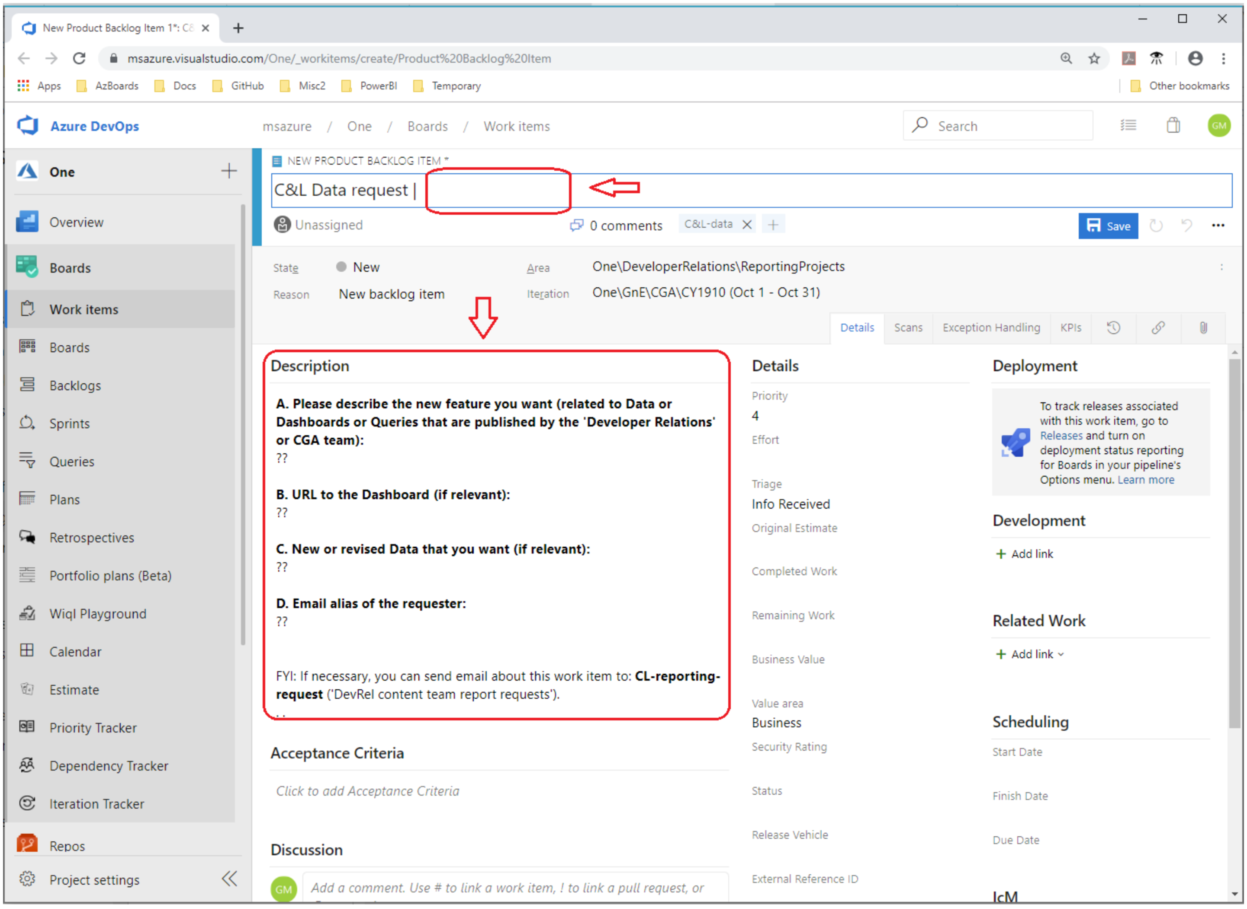
Use the `Submit a request` link

Clicking the **Submit a request** link invokes the proper template, of type **Product Backlog Item**. To complete the template, edit only the following fields:

- **Title**: Append your words to the given prefix.
- **Description**: For each prompt, replace the `??` with your information.
- Optionally, you may add an attachment, such as a screenshot.

Do not edit any other fields.

Finally, save your request. Optionally, you can click **Follow** to be notified when the item changes.



Triage

When a request is submitted, its **Triage** field is set to *Info Received* and the **Priority** is set to *4*. A request moves through the following triage states:

1. Info received (not triaged)
2. More info (not triaged)
3. Pending (not triaged)
4. Triaged

Two times per month at regular Content Data Advocacy Group (CDAG) meetings, new items are reviewed, prioritized, and the **Triage** state is updated. This meeting is open to all CDAG, including members from Content Contributor Success (CCS), Docs, Learn, Patterns and Practices (PnP), and Operational Excellence and Governance (OE&G). Occasionally, CDAG will request additional info from managers of the content teams or from Mei Liang and the Analytics team in order to triage the request.

After an item is prioritized and triaged, the Analytics team handles implementation of the request. CDAG interacts directly with the DevRel Analytics team. If an item requires work by CGA, the Analytics team handles the request and communication with CGA, and reports back to CDAG and the original requestor in the work item. CDAG and the original requestor do not need to contact CGA directly.

If a member of the Analytics team is present at the triage meeting with CDAG, then items can be handed off and assigned immediately; otherwise, CDAG will @mention Mei or other members of the Analytics team or will send email to inform them of new items that have been traiged and are awaiting assignment.

After the Analytics team completes the request, they update the **State** field to *Done* or *Removed*.

Who

As of October 2020, the following people or groups fill relevant positions to the Dashboard Request process:

POSITION	WHO	TASKS
Process coordinator	CDAG (cdag@microsoft.com)	• Maintains this article. • Reviews new requests for clarity. • Holds bi-weekly meetings where triage occurs. • Updates requests during the meetings. • Tracks handed-off requests that missed their implementation target dates. • Maintains our dashboard, and its queries. • Advertises the process to writers.
Content manager sponsor	Jill Reinauer (jillfra)	• Helps decide priorities among the new requests.
Analytics Representative	Mei Liang (mliang)	• Accepts or rejects requests during hand-off meeting. • Sees that accepted requests are implemented.

Other types of requests

- *SiteHelp*: For bugs, such as on Docs
- *Microsoft Forms*: Feature requests to Docs and Build
- *ListeningPost*: For process, policy, and other ideas
- *ListeningPost--Anonymous*: For process, policy, and other ideas
- *Doc Operations*: Cory Fowler (as of October 2019)

Getting Data on API Documentation Content

5/12/2021 • 2 minutes to read

We want the API documentation experience to be as data-driven as possible, and created the API documentation data dashboard. That page is your one-stop shop for everything that has to do with usage of API resources on docs.microsoft.com.

https://msit.microsoftstream.com/embed/video/d69cebfb-f32f-47de-91da-aa504552fb66?autoplay=false&showinfo=true

Getting Data for Your API Documents

We understand the data presented in the dashboard covers a more "global" view of the API documentation space, and you are interested in knowing how *your* content is performing. To do that, you can use the flexibility of Kusto and Lens Explorer. In this document, we are outlining some sample queries that you can run to get information about API docs performance.

To get started, you can use information from the dashboards we already have. See this video for a short intro guide on how to get the data for a specific documentation page:

https://msit.microsoftstream.com/embed/video/c6c853ee-dc9c-45aa-b605-8b935fda37c4?autoplay=false&showinfo=true

You can download Kusto Explorer yourself and try many of the possible queries to dissect the data. The query given in the video is listed here - try it!

```
PageView
| where Site == "docs.microsoft.com"
| where StartDateTime >= startofday(ago(9999d))
| where Url contains "https://docs.microsoft.com/en-us/dotnet/api/system.string.format"
| where enrich_url_pg_query_string !contains "traffic-source=cats"
| extend event_time=tostring(startofmonth(StartDateTime))
| extend event_time_reduced = todatetime(extract("\\d{4}-\\d{2}-\\d{2}",0,event_time,typeof(string)))
| summarize pageviews=dcount(PageViewId,2), visitors=dcount(VisitorId) by event_time_reduced
| sort by event_time_reduced asc
| render timechart
```

If you want an example description of how to read one of the many existing dashboards that we have in the API documentation data dashboard, take a look at our guide: How To: Interpret API Browser Data.

Issues or Missing Data?

Contact the Docs Support team - we'll be happy to address any issues, and route them appropriately.

How to perform customer research

4/16/2021 • 4 minutes to read

This article outlines how we prioritize and complete customer research on the Developer Relations Content and Learning team. It explains:

- Why customer research is important
- How to know if you need to do research
- How to complete customer research

Why is customer research important?

"You can't build a product without understanding the problem you're solving and the people for whom you're solving it. Various forms of research are the best way of understanding people who aren't you. It's really as simple as that."- Laura Klein, Author of *UX for Lean Startups*. Customer research keeps us customer focused and helps us uncover our customers' needs. More importantly, research helps us focus the *right* problems.

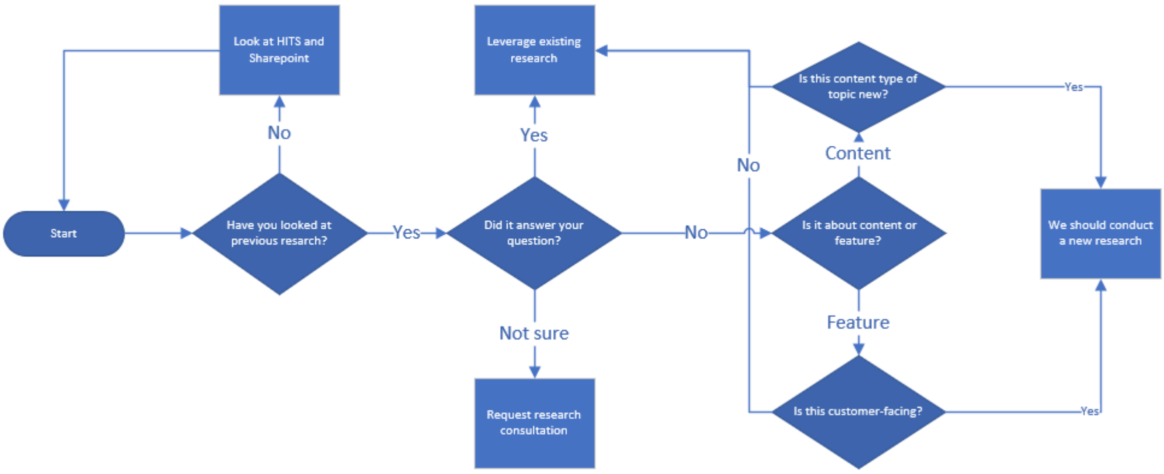
Types of Research

There are three main types of research we might conduct to get feedback on content and features on docs.microsoft.com:

- **Generative research:** Research that helps inform site-wide changes or broad strategic initiatives. Generally, this research tells us who are users are and what goals and behaviors they have.
  - Examples: content model research, ecosystem research, behavioral archetypes project
- **Feature research:** Research on adding new features to the site.
  - Examples: Learn personalized homepage, GitHub Issues research; notifications on Docs, tooltips
- **Content research:** Research that a content developer or docset owner does to help them better organize their content. This research helps us understand what customers are looking for and determines if our content is easy to find.
  - Examples: Topic landing page study, Windows Dev Environment Overview, Architecture Content Customer Research

How do I know if I need to conduct research?

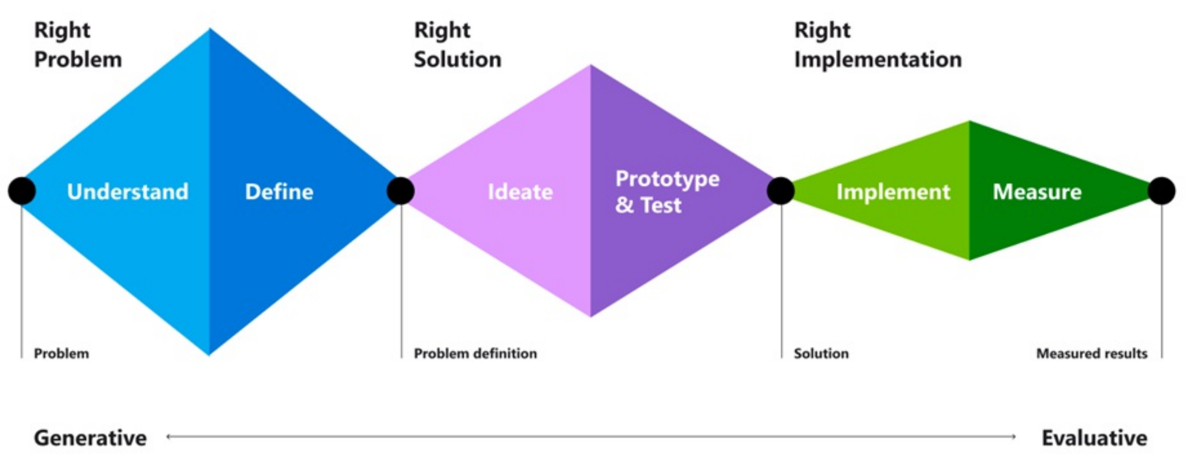
If you're not sure whether or not your feature or content requires research, see the decision tree below.



When should you conduct research?

Typically, there are three times when you might conduct research: in the problem discovery phase, in the solution phase or in the measurement phase.

- In the problem discovery phase, you'll focus on identifying the problems your users are facing.
- Then, in the solution phase, you'll validate whether your solution is usable and addresses the problem. Usually, you'll iterate a few times on it.
- Lastly, once your solution is live, you'll evaluate it and assess how people are using it, whether it meets people's expectations, and how it's performing from a KPI perspective.



Conducting your research

Below is your step-by-step guide to customer research and experimentation.

Before you begin:

1. **Figure out what you're trying to learn.** State the objective of your study. The objective is what you want to get out of the research. Here are a few tactics to further define your study.

- *Hypothesis*: (example) We believe 80% of customers are not happy with X feature because of Y.
- *Research questions* are specific areas that you want to study/gather information. (example) What are customers' impressions of the site?

2. Review research that's already been done.
   - Check out HITS (internal database full of research from other teams) for examples of studies that have

been performed.
- Look through what is posted on the Customer Research SharePoint, which has a list of research performed by the Content and Learning team.

3. **Gather baseline metrics about how things are currently performing.** It's important to know the current state so that you can track any changes that happen. If you don't have baseline metrics, you can't show progress.
   - Content Performance Dashboards:
     - Content and Learning
     - Dynamics
     - Microsoft365
   - Kusto
   - Microsoft Docs Metrics (MDM)

File an Azure Boards (previously called VSTS) work item to start the process

- What problem you are trying to solve? What are you hoping to learn?
- Who is your target audience?
- List a date you'd like to see this completed by.

Please note, the research process requires at least two weeks lead time. We will not be able to help you with a page that is launching tomorrow

Once your Azure Boards item has been triaged, someone from the research team will contact you to set up a meeting.

Below is a brief description of what to expect in each meeting.

1. **Introduction and define problem.** Review objectives, hypothesis, and/or research questions. Go over analysis of existing data to get a baseline if available.
2. **Discuss research methods.** Determine which method to use, how to recruit customers, screening guidelines, and create a research plan.
   - *Next steps: create research artifacts*
3. **Review prepared research materials** (survey outline, wireframe, prototype, etc.) We will review your research materials together.
   - *Next Steps: Conduct research*
4. **Analyze results and share knowledge** Write-up results in HITS format and post to SharePoint. Decide whether or not more research needs to be done. (Implementation experimentation, update contributor guide, share results).
   - *Next Steps: Write up research and post on SharePoint. Host a Share & Learn session.*

Expectations

- **Partnership** Please keep in mind that this is a partnership. We will work with you to provide guidance and assistance to help you complete your own research.
- **Come prepared.** At the end of each meeting, you'll be assigned action items to complete before the next meeting. Come to each meeting prepared so that we can make the best use of our time.
- **Track your work in Azure Boards.** This helps hold people accountable, allows the team to load balance, and provides a visible way to document what research is going on. Additionally, it allows others to learn from the work being done and shares knowledge across the team.
- **Have a growth mindset.** We know this might be out of your comfort zone or you might not have any experience in this area, but that's what we're here for. Reach out if you have any questions. This is a learning

process for all.

- **Share your results.** Good research makes everyone better. Share your findings with the team to show how customer research translates into actionable items.

Next Steps

If you're ready to get started, file an Azure Boards work item! We look forward to hearing from you!

Choose a customer research methodology

4/16/2021 • 4 minutes to read

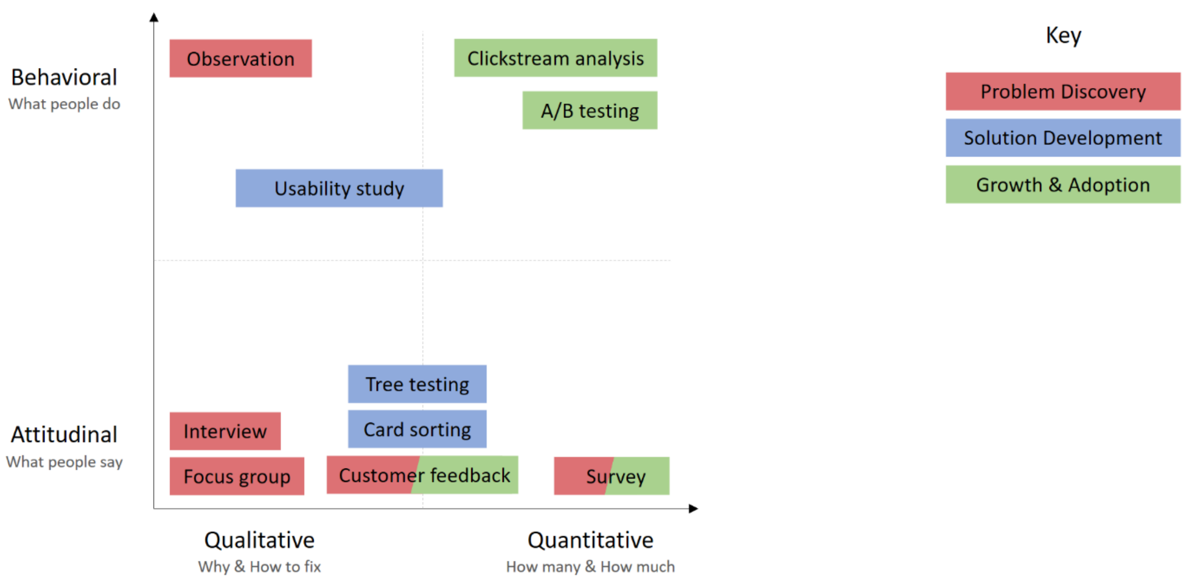
There are many different types of customer research. To decide which method to use, it's important to determine what you're trying to learn and which phase you're in as outlined in the following table:

Research methods and tools by product phase

	Problem discovery	Solution development	Growth & adoption
Questions	What problems do customers have? What's already been done and being used?	Are we building the right thing? Can users find what they're looking for?	Did we build the right thing? How are we doing?
Approach	Qualitative and quantitative	Mainly qualitative	Mainly quantitative
Research Methods (Need customers)	Observation, interview, focus group, survey, customer feedback	Card sorting, tree testing, usability study	A/B test, Click stream, survey, customer feedback
Other tools (We can do on our own)	Competitive analysis, task analysis, stakeholder interview, persona	Literature review, storyboard, journey map, guideline review, heuristic evaluation, cognitive walkthrough	

Another way to think about different types of research is whether they're attitudinal or behavioral as well as qualitative or quantitative. This chart depicts where research methods fall on these two axes.

Research method by data type



Research method by inquiry

If you're not sure where to start, try focusing on what information you're looking to learn.

Who are our customers? What problems do they face?

Interviews

Interviews can be used for lots different scenarios. For example, you might use an interview to ask participants about how they complete certain tasks. Your customers may remember some specific cases where the product worked well or poorly and can often provide more vivid details about these incidents. You can use this tactic to get an idea of the strengths and weaknesses of your content in helping users accomplish their tasks. You can also use interviews to help you identify questions to ask in a broader survey. It's important to note that in customer conversations we're focusing on participants' past experiences - not asking them to speculate about the future. Customer interviews may seem daunting at first, but it's helpful to keep in mind that it's simply a conversation between two human beings.

Surveys and questionnaires

Surveys provide you with answers similar to what you'd get from user interviews. The downside is that you can't dive deeper into those answers as there's no direct interaction with the customer. On the plus side, they allow you to get a larger volume of responses, which can open up the opportunity for more quantitative analysis. It's important to ask carefully worded questions that are free from bias.

Can people easily find information on your website?

Card sorting

This technique helps generate or validate a structure, like navigation, a TOC, or content organized into categories. To do a card sort, make a list of the concepts you need to organize, write them on cards, and ask users to organize them into categories. To generate new structures, do an open sort, where participants put the cards into groups and then name the groups themselves. To validate a structure you've already developed, do a closed sort, where you provide categories to sort the cards into. The results of a card sorting study can help you to decide the structure of your article, how to label your table of contents and/or how to group your content.

Tree testing

Tree testing helps assess the findability of content; it can be used for any kind of navigational structure but is particularly well-suited to tables of contents. In a tree test, you create a tree structure (without any UI) and define items for participants to find. Participants then try to find the items you've identified by navigating through the tree and noting when they think they've found it. Because they separate the actual navigation structure from the UI, tree tests are useful to see how understandable the labels you've chosen are and whether the structure you've defined lets people find things they're looking for. At the end of the study, you'll have metrics including task success rate, time to complete the task, and an analysis of routes users took through the site tree before selecting an answer (correctly or incorrectly).

What do people think?

Focus groups

A focus group is when you bring together a group of participants to ask them about a certain topic. For example, you might host a focus group to ask participants how they like to learn or what their initial reaction to a product is. Having a group of users together can also help them to jog memories and ideas in each other that they may not have otherwise remembered. Beware of group think as it can influence the way participants respond in these kinds of settings. Generally, our team hires vendors to complete these types of studies.

What do customers do?

A/B testing

For information about A/B testing please see the experimentation wiki

Observations

Observations involve studying people as they go about their everyday lives or tasks. This helps understand how people complete a task or use a work-around for a known problem. For example, you might use an observation study to see how developers find and use technical documentation. This is usually part of an interview and is brief. Observations are exploratory whereas usability studies generally have a goal or task to be completed.

Search engine optimization (SEO)

SEO is important because 60-70% of our traffic comes from search. By using tools such as google analytics, we can understand what customers are searching for and tag our pages with the most effective key words. More information on SEO can be found in the SEO and analytics presentation video.

Content reporting

Content reports can be used to gather performance on a page. These reports include engagement metrics such as bounce rate, CSAT, views, and traffic data. For more information, see Report and interpret content engagement metrics.

Usability testing

In usability testing, you come up with a list of tasks and observe how correctly or incorrectly users accomplish those tasks. For example, you might perform a usability test to understand how many users successfully register for a MS Learn account. The data you gather might include how long it takes users to complete the task, how many errors they make, where they deviate from our ideal and how many users complete the task.

UserTesting

3/5/2021 • 2 minutes to read

What tools do we have available to conduct studies?

Our main user research tool is UserTesting.com. UserTesting is a platform for getting rapid customer feedback on almost any customer experience you can imagine, including websites, mobile apps, prototypes, and real world experiences. You'll receive audio and video recordings of real people speaking their thoughts as they complete tasks you specify. With UserTesting, you can run moderated and unmoderated tests to get connected with your customers on a global scale. You can use this resource to deep dive in a problem space and get feedback on concepts and prototypes. To learn more about UserTesting, see their FAQs

How do I get a UserTesting license?

To take advantage of this resource, you'll need to submit a request below. You'll receive instructions to set up your UT seat and then be asked to complete some prerequisite e-learning courses before receiving your license.

What is the process?

1. Complete e-learning courses
2. Submit a request for a license
3. Receive an account from the research team
4. Build study plan with the help of the research team (there are templates available here)
5. Receive approval to launch study

Note, we require at least three business days' notice to launch a UserTesting study

What training is required?

In order to get access to UserTesting.com, you must complete the three online eLearnings listed below:

- Getting Comfortable with Customer Conversations - In this training you'll learn best practices for having customer conversations without bias. This includes preparing for conversations, moderating conversations, and documenting insights.
- Problem-Focused Thinking- In this training you'll learn the difference between problem-focused and solution-focused thinking and how to be more problem-focused in your work.
- Ethics and Privacy in Customer Interactions - In this training you'll learn how to act ethically and compliantly when interacting with customers.

After you've completed these trainings, submit a request by filing this work item

FAQs

What type of UserTesting account will I receive?

You will receive a license that gives you two months of access. During that time, you will be able to run up to three studies (moderated and/or unmoderated) with a maximum sample size of 6 per study. We currently have 10 seats available for the entire DevRel team. If after two months, you would like to run another study, feel free to reach back out. Note, you will not have to take the required training again.

Can I get quantitative as well as qualitative feedback?

Yes, you can collect both quantitative and qualitative feedback with this UserTesting trial license.

How quickly will I receive results?

Moderated studies on UserTesting depend on your calendar availability. Typically sessions are scheduled within 24 hours of launching the test. Unmoderated studies are faster and can be picked up by qualifying participants shortly after launching. Timing varies depending on the complexity of the customer profile you're looking for, however most studies are completed within 2 hours.

When should I conduct research?

Reference this guidance in the Contributors' Guide.

What resources are available?

Once you have a UserTesting account, there are many helpful trainings and resources available. Below are a few to help get you started:

Meet Insight Core

Course 1: The Basics

Course 4: Setting up your Test in Insight Core

Use Cases in the UserTesting Platform

There is also a UserTesting + Microsoft Resource Guide which features:

- Upcoming webinars and events
- Recordings of past webinars
- Helpful resources and links

Customer interviews

4/16/2021 • 3 minutes to read

What is it?

Interviews are a great way to get customer data. At first, customer interviews may seem daunting, but in reality it's just a conversation between two people.

When do I use it?

Interviews are used for lots of different things. You can use interviews to ask participants about how they complete certain tasks. They may remember times when something worked well or poorly and can provide more details about these events. You can use interviews to get an idea of the strengths and weaknesses of your content and understand how useful it is at helping customers accomplish tasks. You can also use interviews to help you gathers questions to ask in a broader questionnaire or survey. It's important to note that we are focusing on participants' past experiences - not asking them to speculate about the future.

How do I do it?

One of the most important parts of the interview is the preparation. To prepare for a customer interview, you need a hypothesis, a screener, and a discussion guide.

- A *hypothesis* is an informed guess that changes over time. It's created based on existing customer data and your assumptions about the customer, the problems they face, and the tasks they need to perform (also known as jobs-to-be-done). The hypothesis is the basis of your study. The information you learn through your research will help you to refine your hypothesis. To formulate your hypothesis, try one of the examples:
  - We believe that [customers] are most frustrated about [job-to-be-done] because [problem]. We believe addressing this [problem] will increase [metric].
  - We believe this [solution] will increase/decrease [End-User Metric].
- A *screener* is a set of specific questions that come before the interview and determine whether or not a person qualifies for your study. Tips for making a screener:
  - Keep the screener short. Target between 5 and 10 questions.
  - Avoid questions that only require "yes" or "no" answers.
  - Don't make the "correct" answer obvious. People will try to answer in a way that will qualify them for your study.
  - Always provide a neutral answer such as "none of the above" or "I don't know."
  - Aim to be diverse (varied genders, ages, ethnicity, location) and inclusive (varied computer confidence and those with different abilities)
- A *discussion guide* is a script or outline for the conversation. It's important to make sure that you're asking the right questions to get the data you need. Customer interviews are most valuable when they provide actionable feedback and confirm or disprove our assumptions. Here are some tips to writing a discussion guide:
  - Always have more questions than you think you need.
  - Ask open-ended questions. Open-ended questions leave space for the customer to share their experience and help you remove bias from the discussion. For example: Tell me about the last time you used technical documentation. What made this experience good or bad?

    - Add quantitative questions to your interview. Ask customers to rank things on a scale from one to five based on how useful something is or how frustrated they are.
    - Quick pulse calls are a great way to ensure that you're heading down the right path before you invest further into your research.

How many customers do I need?

For small changes, you need between five and seven interviews. For a large change or an entirely new approach, you want to hear from as many customers as possible. The best indicator that you've done enough research is that you stop hearing new information from customers, also known as theoretical saturation.

What tools are available?

These tools available to connect directly with customers:

- Moneyball : The Moneyball concierge team finds and schedules participants using their own database.
- UserTesting : UserTesting is an online platform for moderated and unmoderated interviews. They have their own panel of over 2 million global users.
- Azure Advisors : Azure Advisors is a community of customers who have opted in to provide feedback on Microsoft products.

Customer interview basics

- Asking The Right Questions On User Research Interviews : advice on how to perform a customer interview
- Formulate a Discussion Guide : a step-by-step guide to create a discussion guide
- Screener Library : examples of pre-built screeners
- Screener Guide : how to make a screener
- Problem-focused Interview Workbook: a step-by-step guide to writing a hypothesis, screener, and discussion guide
- Interviewing Developers : an example discussion guide by Mimi Gentz
- Reducing Bias : e-learning course from the Signal team
- Moneyball Online Training: e-learning courses

Become an interview expert

- Interview Tips
- Avoid Leading Questions to Get Better Insights from Participants
- Understanding Cognitive Biases

Surveys/Questionnaires

4/16/2021 • 2 minutes to read

What is it?

A survey is a list of questions intended to gather information about a group of people. Surveys are quick to administer and allow you to gather large sample size, but can be tricky to do well. It is important to have well-crafted and thoughtful questions to make sure that you're getting the correct information.

When do I use it?

Surveys can be a good way to follow up on qualitative data, such as interviews, to go deeper, and get more quantitative data to substantiate your hypothesis. They can also be a good way to surface areas from a large group of people to go deeper in on follow-up customer conversations.

How do I do it?

Making a concise and effective survey can be tricky. First, start with listing out what you want to learn from your survey. Begin by writing down the answers you are hoping to get. Once you know the answers you are seeking, you can turn these into the questions that you'll ask. This will help you gain actionable feedback and avoid asking unnecessary questions. Below are a few tips to writing effective survey questions,

- Make sure to use simple language and break down big ideas into multiple questions.
- Avoid leading questions and offer balanced answers (make sure to provide an equal amount of positive and negative choices.
- Always include a "none of these" or "I don't know" option.
- Avoid survey fatigue by asking 5-10 questions. If possible, show a progress bar so participants know how far along they are.
- Make sure to pre-test your survey before putting it out in the wild. Send your survey to a few colleagues for feedback and to see if any questions were unclear

How many customers do I need?

Here is one article from SurveyMonkey and one article from Qualtrics explaining population, margin of error and confidence level.

What tools are available?

Microsoft Forms

Writing good questions

It's important to write clear, unbiased questions. If you don't know where to start, check out the following articles.

- Surveys 101: A Simple Guide to Asking Effective Questions
- Survey Questions 101: Do You Make These 7 Mistakes?
- Guidance for Crafting Survey Questions
- Example Survey that Mimi Gentz Created for Cosmos DB Research

- Survey Question Best Practices and Recommended Questions

Become a survey expert

- LinkedIn Learning – How to Create a Survey
- LinkedIn Learning – What Makes a Good Survey
- LinkedIn Learning – How to Analyze Survey Results

Customer research and experimentation FAQs

3/5/2021 • 2 minutes to read

Which process answers which types of questions?

Customer research provides qualitative data and answers questions like:

- "why?"
- "what do users say?"
- "what do users think?"

Experimentation provides quantitative data and answers questions like:

- "how many?"
- "what do users do?"

What do you do first? Does one come before the other?

In general, we recommend doing research first. Once you have an idea of the direction you'd like to go, you can use experimentation to validate. In other words, if you don't know where to start, do some research. If you have a well-formed hypothesis, do an experiment.

A few things to note: with A/B testing, anyone can land on your page. With research, you can test with a specific audience. Lastly, if there is significant risk, we should always test with experimentation.

Where can I find information about everything that is offered?

If you want to learn more about the research program, see how to perform customer research. To read more about the experimentation program,see the experimentation wiki

Azure Pulse customer feedback program overview

5/10/2021 • 2 minutes to read

The Azure CXP runs a program called Azure Pulse that collects customer feedback through Technical Account Managers and Cloud Solution Architects. The customers in this program are selected for strategic importance, risk of churn, and edge case scenarios. The feedback items are periodically routed to a Content & Learning PM for triage to the appropriate content owner in the Content & Learning organization for a response.

Handling Azure Pulse feedback items quickly is high priority and required activity for content teams. Progress and completion are tracked by CXP and shared monthly in Azure Fundamentals.

Process overview

1. A Microsoft field employee such as a Technical Account Manager or a Cloud Solution Architect creates an Azure Pulse work-item based on customer feedback.
2. Azure CXP classifies the feedback items by product and documentation feedback is labeled as "Content Services".
3. A week before Fundamentals, the Pulse team sends a status/reminder mail to encourage teams to make progress before to Fundamentals.
4. The Azure Pulse Lead for Content & Learning, Mangesh Sangapu (msangapu), is responsible for triaging and assigning new items to appropriate content developers and/or team managers for the affected Azure service. The Pulse Lead sends an e-mail notification to the affected developers and/or team managers indicating a new Pulse work-item has been assigned to them.
5. The manager or delegate must review the feedback, investigate as needed, and prepare a "customer ready message" (CRM) as a response the field employee can provide to the customer. The response should address feedback provided and give next steps or the plan of action for addressing the feedback.

Query

To query for new or existing items, visit: https://vstfrd/Azure/RD/_workitems?id=e404e8af-db6c-4ea3-bcc0-91fa062fc813&_a=query

Program goals

1. White Glove / Customer Rescue
2. Drive improvements into Azure services and experiences
3. Build engineering empathy
4. Committed customer pool for design reviews and private previews

Service owner guide

See the Pulse Service Owner Guide for complete details. SLAs are documented on the last page.

Azure CPE PSAT Program overview

10/3/2019 • 2 minutes to read

The Azure PSAT survey program tracks & provides Azure all-up customer experience measurement & insights with quantitative data and compete benchmarking. The Azure PSAT team partners with the C+AI engineering teams and the CMR (Customer & Market Research) team on survey design, survey result analysis, action plans, and improvement tracking.

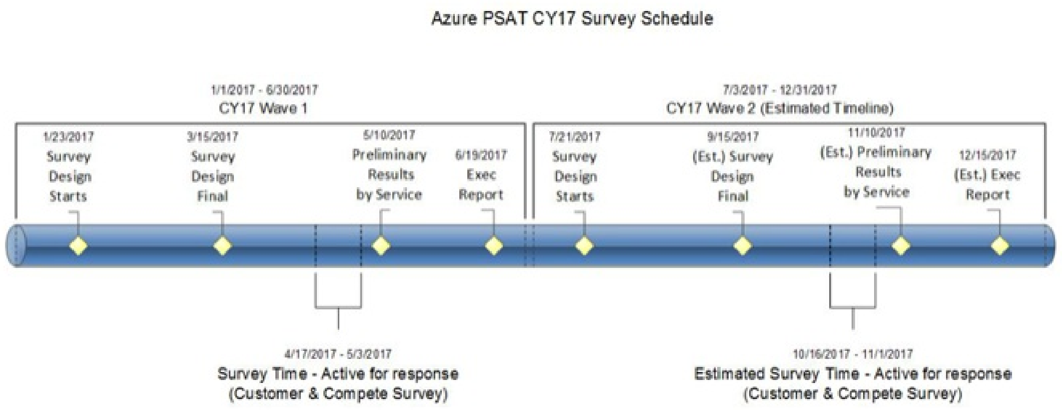
Survey Coverage

The Azure PSAT survey was reestablished by Azure CXP team in Nov 2015 and has been running twice a year since then. The survey program now runs the Azure customer survey and Azure compete survey in parallel and integrates with Azure customer telemetry data.

- Azure Customer Survey: Microsoft-identified email survey to Azure customers. World-Wide English Survey covering all existing Microsoft Azure customers with billing revenue >$0, and opt-in for marketing promotion communications (that is, customer feedback surveys).
- Compete Survey: Microsoft not revealed as survey sponsor. The US English Email Survey to panel samples is provided by a third-party survey vendor focusing on US AWS and US GCP.

Program Goals

1. Understand customer satisfaction and related drivers
2. Drive improvements into Azure services and experiences
3. Compare Azure customer feedback with the feedback of top competitors



Timeline

Resources

- For more info, go to the Azure PSAT Portal
- Executive Reports: Azure PSAT Report Archive
- For Newsletter Join Azure PSAT Distribution Group
- Azure CXP Program Lead: Amira Youssef

GitHub issue and public pull request reports

4/16/2021 • 8 minutes to read

There are several types of reports you can run for GitHub issues:

- You can sign up for scheduled reports that can be emailed to your inbox, as described later in this article Azure GitHub reports.
- Download and run GitReporter tool.
- Run reports at the GitHub Issues dashboard https://aka.ms/githubissues.

Getting started with using Azure GitHub reports

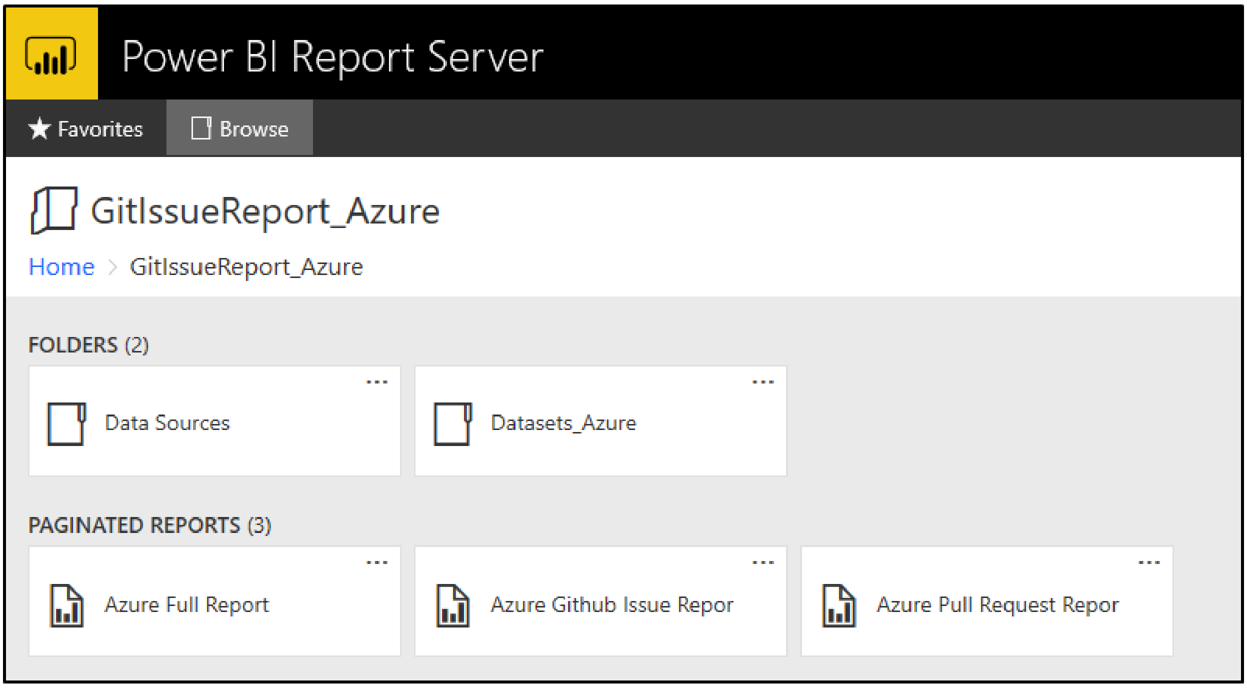
GitHub reports combine PRs and GitHub issues into a report for the Azure repo. You can receive these reports automatically and filter on them for details such as service, author, and manager. Here's how to configure and use these reports.

> **NOTE**
> Contact Van (vanto@microsoft.com) with any feedback or issues.

There are three reports created:

- **Azure Full Report** (PRs + issues) - `http://roguereporting.westus2.cloudapp.azure.com/reports/report/GitIssueReport_Azure/Azure%20Full%20Report`
- **Azure GitHub Issue Report** - `http://roguereporting.westus2.cloudapp.azure.com/reports/report/GitIssueReport_Azure/Azure%20Github%20Issue%20Report`
- **Azure Pull Request Report** - `http://roguereporting.westus2.cloudapp.azure.com/reports/report/GitIssueReport_Azure/Azure%20Pull%20Request%20Report`

> **NOTE**
> Please copy and paste the above links into your browser. The SSRS Service is not using HTTPS.



Location of these reports:

`http://roguereporting.westus2.cloudapp.azure.com/reports/browse/GitIssueReport_Azure`

IMPORTANT

If you are unable to access the Power BI Report Server, and get an unauthorized message, please contact Van (vanto@microsoft.com) for access.

The color codes in the age column correspond with the SLA for pull requests:

- 0 - 7 days: **Green**
- 8 - 14 days: **Yellow**
- 15 days or more: **Red**

NOTE

The data source behind these reports is refreshed on an hourly basis between **6:30AM** and **6:30PM** (Pacific Time) daily. This means that if you update a PR or issue outside of these hours, you will not see the data refreshed in the reports until the next refresh window.

There are two ways to consume these reports:

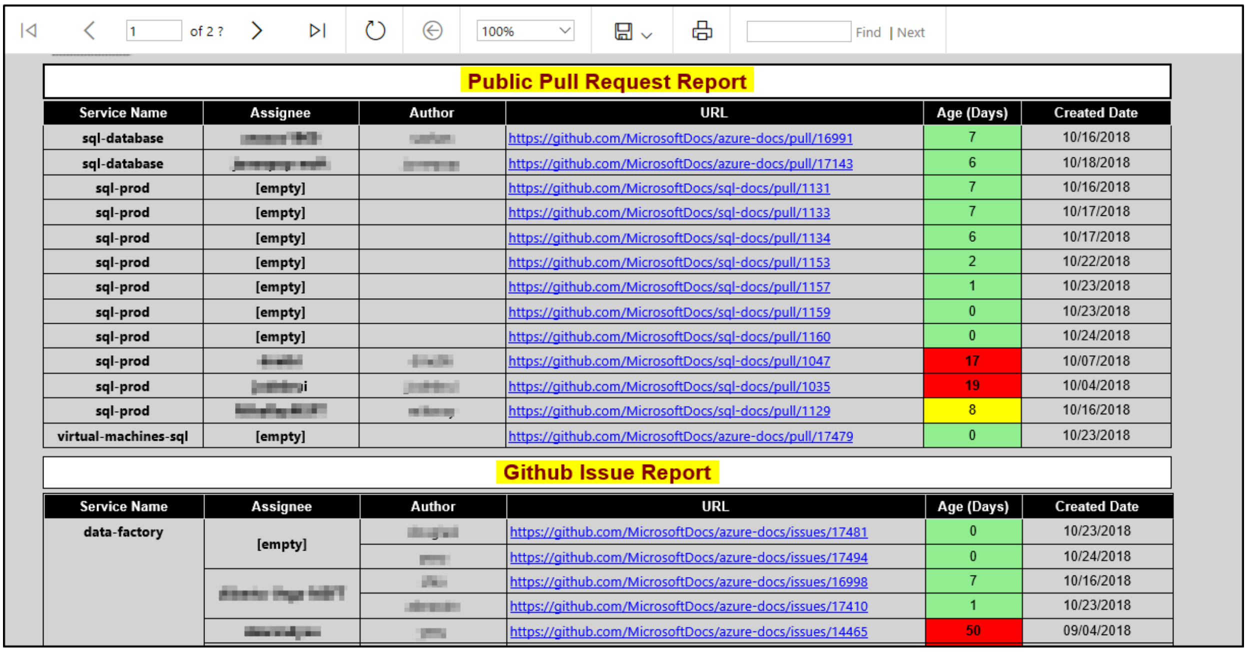
- Interactive reporting - view and configure the report in the web GUI of the Report Server.
- Subscribe to reports - configure the report to be sent via email.

Interactive reports

Interactive reporting allows the user to filter the results based on preference of services and assignees. The user can also export the results to their desired format, such as Excel or PDF.

The **Azure Full Report** incorporates both PRs and GitHub Issues into one report. **Azure GitHub Issue Report** only has Issues, and **Azure Pull Request Report** only has PRs.

Azure Full Report



Public Pull Request Report					
Service Name	Assignee	Author	URL	Age (Days)	Created Date
sql-database	[redacted]	[redacted]	https://github.com/MicrosoftDocs/azure-docs/pull/16991	7	10/16/2018
sql-database	[redacted]	[redacted]	https://github.com/MicrosoftDocs/azure-docs/pull/17143	6	10/18/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1131	7	10/16/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1133	7	10/17/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1134	6	10/17/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1153	2	10/22/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1157	1	10/23/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1159	0	10/23/2018
sql-prod	[empty]		https://github.com/MicrosoftDocs/sql-docs/pull/1160	0	10/24/2018
sql-prod	[redacted]	[redacted]	https://github.com/MicrosoftDocs/sql-docs/pull/1047	17	10/07/2018
sql-prod	[redacted]	[redacted]	https://github.com/MicrosoftDocs/sql-docs/pull/1035	19	10/04/2018
sql-prod	[redacted]	[redacted]	https://github.com/MicrosoftDocs/sql-docs/pull/1129	8	10/16/2018
virtual-machines-sql	[empty]		https://github.com/MicrosoftDocs/azure-docs/pull/17479	0	10/23/2018

Github Issue Report					
Service Name	Assignee	Author	URL	Age (Days)	Created Date
data-factory	[empty]	[redacted]	https://github.com/MicrosoftDocs/azure-docs/issues/17481	0	10/23/2018
		[redacted]	https://github.com/MicrosoftDocs/azure-docs/issues/17494	0	10/24/2018
	[redacted]	[redacted]	https://github.com/MicrosoftDocs/azure-docs/issues/16998	7	10/16/2018
		[redacted]	https://github.com/MicrosoftDocs/azure-docs/issues/17410	1	10/23/2018
	[redacted]	[redacted]	https://github.com/MicrosoftDocs/azure-docs/issues/14465	50	09/04/2018

There are five parameters available for filtering in the **Azure Full Report**.

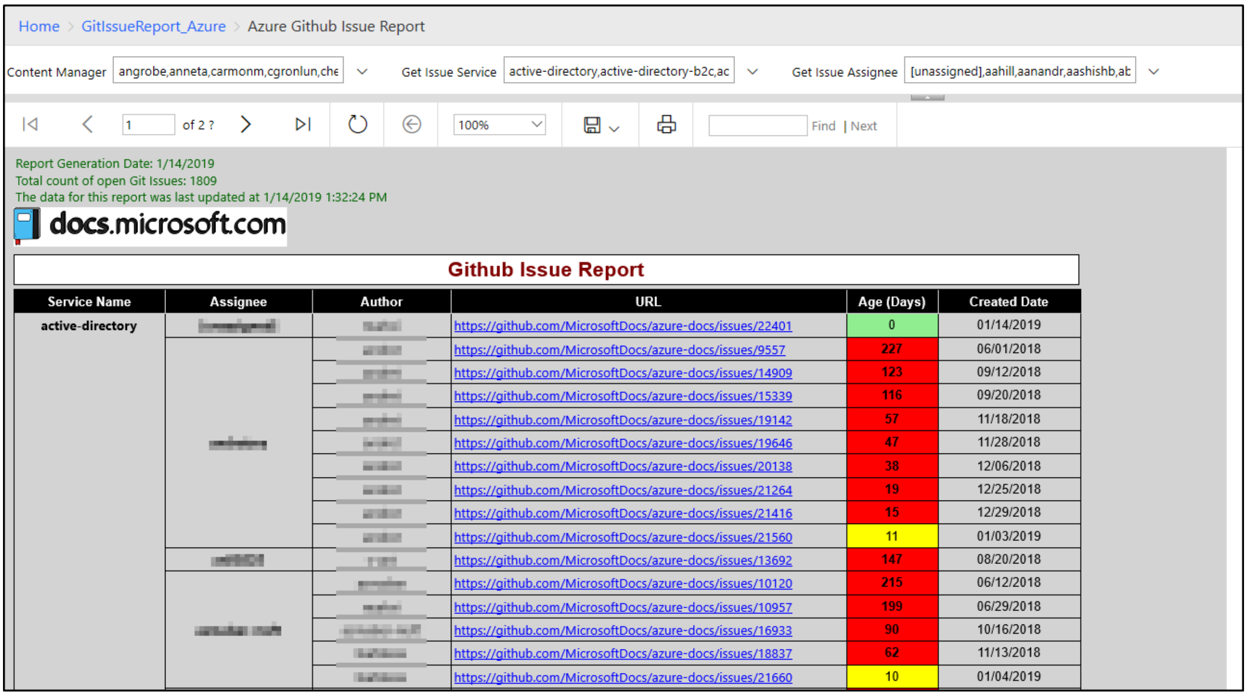
- **Content Manager**: Filter based on the Content Manager alias, which will limit the result set to services owned by the Content Manager(s)
- **Git Issue Service**: GitHub Issue Report filter based only on the selected service(s)
- **Git Issue Assignee**: GitHub Issue Report filter based only on the selected assignee(s)
- **Git PR Service**: Pull Request filter based only on the selected service(s)
- **Git PR Assignee**: Pull Request filter based only on the selected assignee(s)

The top-level parameter is Content Manager. Based on this selection, the services that are available will be narrowed down to the services owned by the selected Content Manager(s). The next level parameter is Service. Based on this selection, the Assignee will be narrowed down to the Assignee(s) that has Issues or PRs opened in that service.

> **NOTE**
>
> If you select a Content Manager and/or a Service that doesn't have GitHub Issues currently open, there will be a null value in the parameter selection for Assignee. The full report must have at least 1 selection in each parameter, or the report will fail to render.

Once the user is done with their parameter selection, they can select the **View Report** button located on the upper right-hand corner to render the report.

Azure GitHub Issue Report



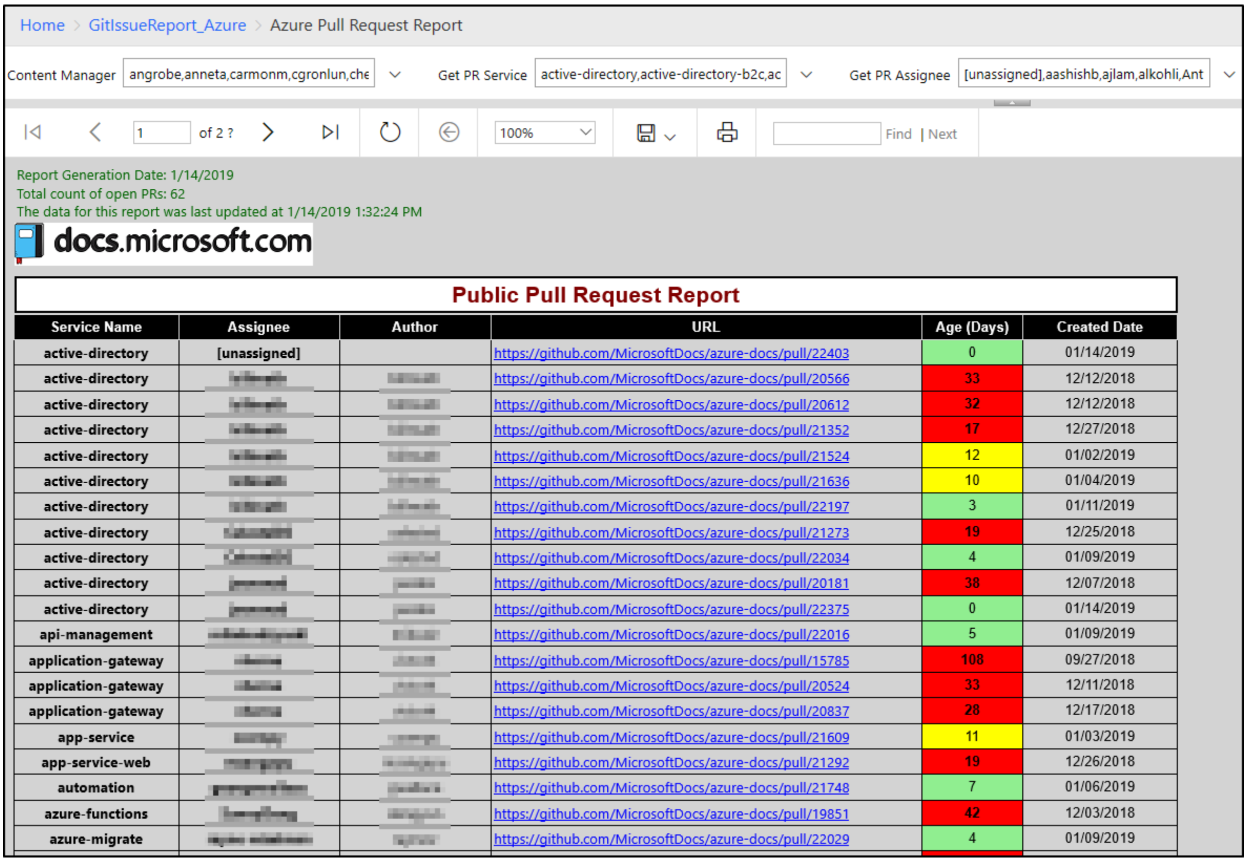
There are three parameters available for filtering in the **Azure GitHub Issue Report**.

- Content Manager, Git Issue Service, and Git Issue Assignee

The following table describes the columns and definitions of the GitHub Issue Report.

COLUMN	DEFINITION
Service Name	Refers to the Service listed in the GitHub Issue. The value is obtained from the labels assigned from GitHub.
Assignee	Refers to the current designated user that is under Assignees in the GitHub Issue. This username is the GitHub username.
Author	Refers to the author of the article from the GitHub Issue when it was opened. The value is obtained from the **Document Details** of the GitHub Issue, and is the Microsoft alias of the author. This value is also populated from the ms.author metadata for the article when the Issue was first opened.
URL	Refers to the URL of the GitHub Issue.
Age (Days)	Refers to the number of days that has passed since the GitHub Issue was opened and when the report execution.
Created Date	Refers to the date that the GitHub Issue was created.

Azure GitHub Pull Request Report



Service Name	Assignee	Author	URL	Age (Days)	Created Date
active-directory	[unassigned]		https://github.com/MicrosoftDocs/azure-docs/pull/22403	0	01/14/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20566	33	12/12/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20612	32	12/12/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21352	17	12/27/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21524	12	01/02/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21636	10	01/04/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22197	3	01/11/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21273	19	12/25/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22034	4	01/09/2019
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20181	38	12/07/2018
active-directory	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22375	0	01/14/2019
api-management	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22016	5	01/09/2019
application-gateway	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/15785	108	09/27/2018
application-gateway	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20524	33	12/11/2018
application-gateway	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/20837	28	12/17/2018
app-service	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21609	11	01/03/2019
app-service-web	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21292	19	12/26/2018
automation	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/21748	7	01/06/2019
azure-functions	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/19851	42	12/03/2018
azure-migrate	[illegible]	[illegible]	https://github.com/MicrosoftDocs/azure-docs/pull/22029	4	01/09/2019

There are three parameters available for filtering in the **Azure Pull Request Report**.

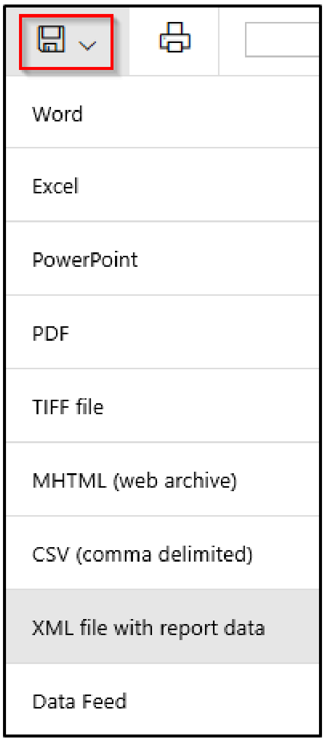
- Content Manager, Git PR Service, and Git PR Assignee.

The following table describes the columns and definitions of the GitHub Pull Request Report.

COLUMN	DEFINITION
Service Name	Refers to the Service listed in the GitHub Public Pull Request. The value is obtained from the labels assigned from GitHub.
Assignee	Refers to the current designated user that is under Assignees in the GitHub Issue. This username is the GitHub username.
Author	Refers to the Microsoft alias that corresponds to the user in the Assignee column.
URL	Refers to the URL of the GitHub PR.
Age (Days)	Refers to the number of days that has passed since the GitHub PR was opened and when the report was executed.
Created Date	Refers to the date that the GitHub PR was created.

Export the results from the report

You can export the results from the report after choosing your desired parameters and rendering the report by selecting the **save** icon in the report controls. Choose your desired export format:



To print the report results, select the **print** icon next to the **save** icon.

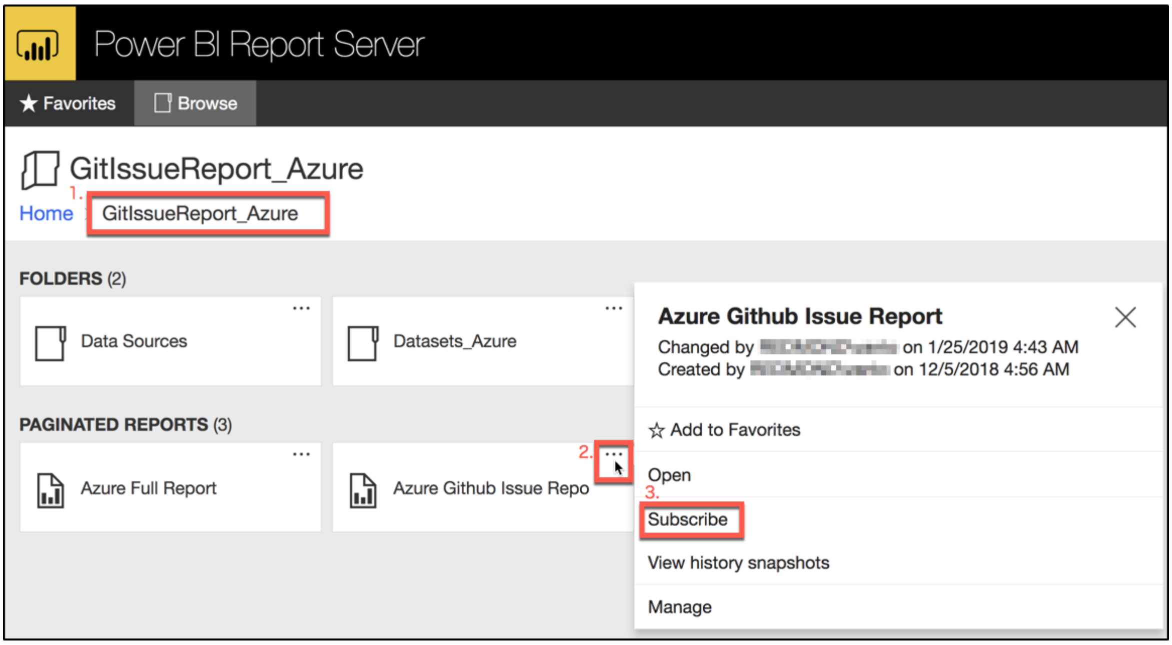
Subscribe to reports

Subscriptions are the ideal way to consume these reports, as they allow for reports to be automatically generated and sent via email, in any format of your choosing. The simplest format is MHTML, which is readable in email. You can set up subscriptions for yourself, your team(s), or your engineering partners.

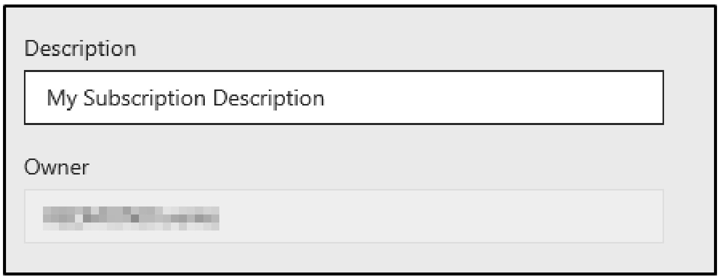
Setting up subscriptions

1. Go to the **GitIssueReport_Azure** folder in the Power BI Report Server, and select the **"..."** button next to the report that you want to subscribe to. Select **Subscribe**.

   `http://roguereporting.westus2.cloudapp.azure.com/reports/browse/GitIssueReport_Azure`

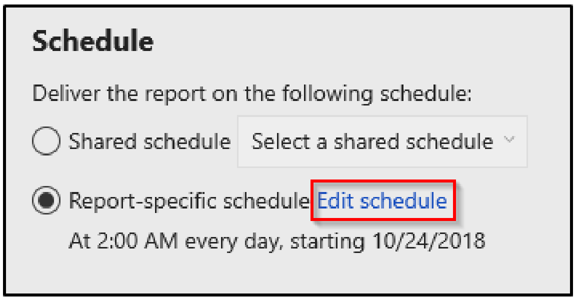
   

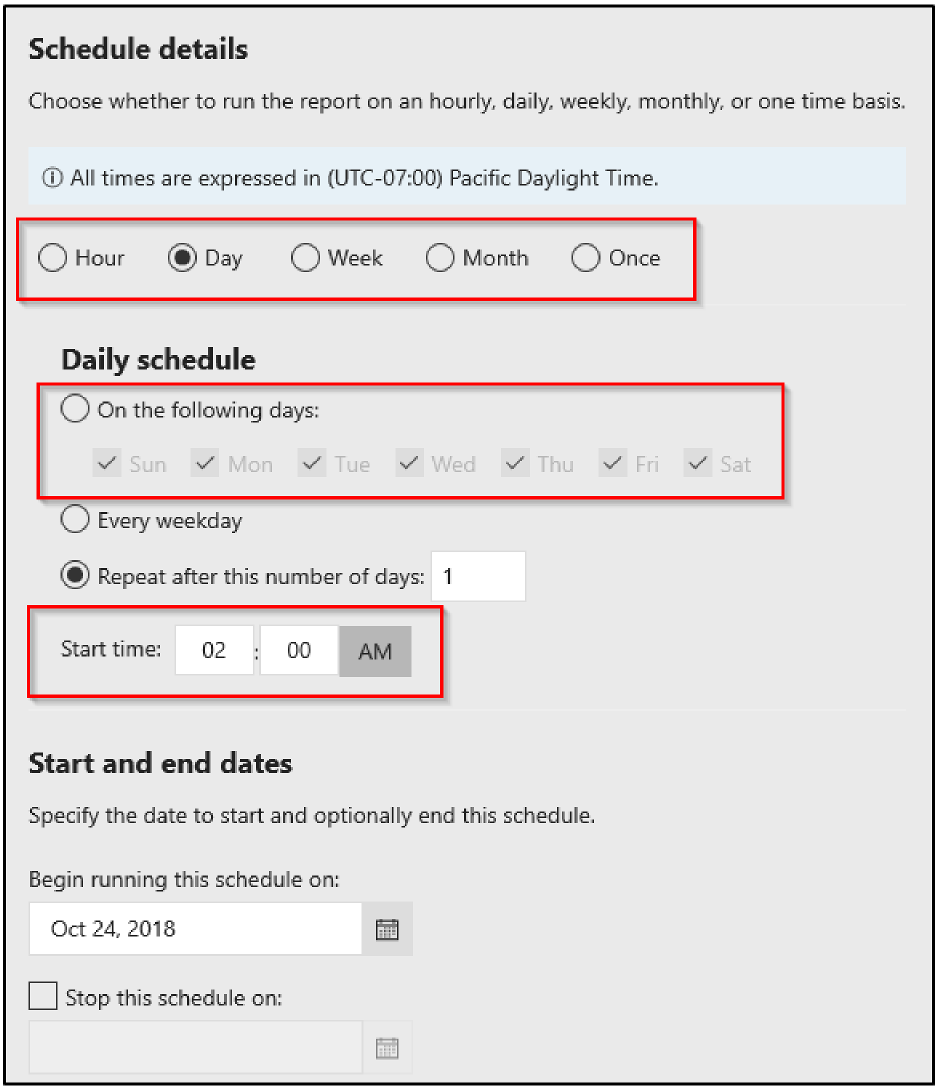
2. Enter a description for this subscription.



3. Type of subscription: **Standard subscription** (default)

4. In the **Schedule** portion, select **Edit schedule** and configure the **Schedule details**.



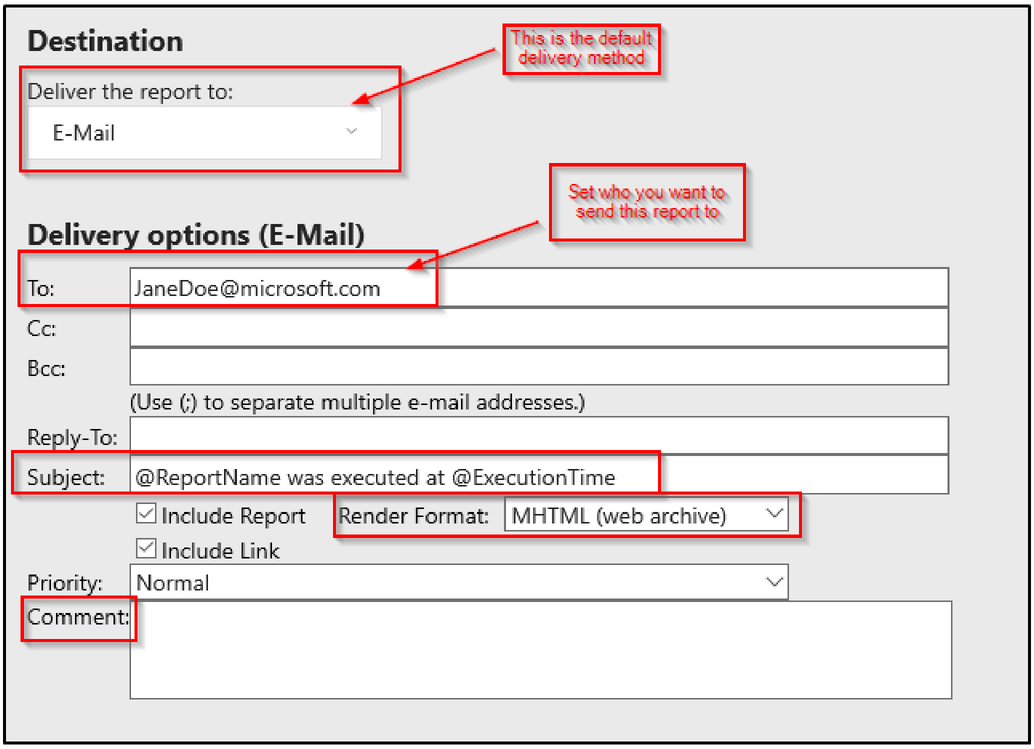


5. Once you're done editing the schedule, select **Apply** at the bottom of the page.

6. The next section of configuring your subscription is to set the delivery method to **E-mail**.

7. Add an email address you want to send this report to, and then *optionally* configure the **Delivery options** for email: **Subject** and **Render Format**. If you want to send the report to multiple email

addresses, separate the email addresses using a semicolon (;).



IMPORTANT

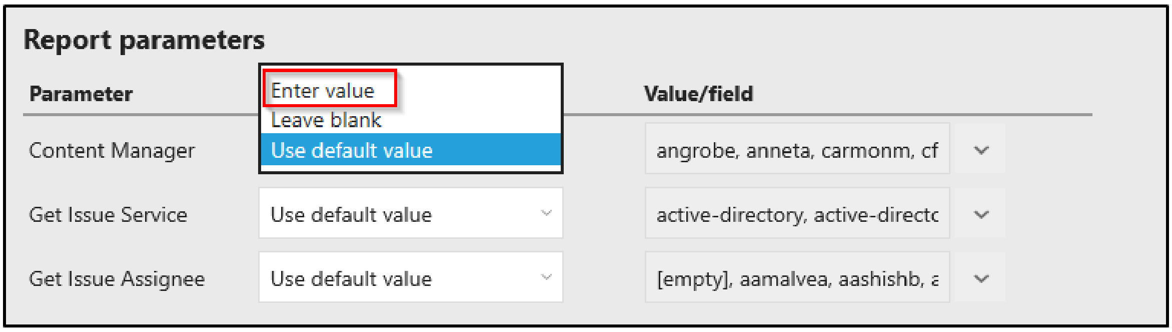
Use fully qualified domain addresses for the email address. For example, JaneDoe**@microsoft.com**. You will get an error if you only add an alias.

NOTE

Normal users will not see a section for wider distribution such as adding emails to the **Cc** or **Bcc** line. Add additional emails on the **To** line. If there is a need for greater permissions, contact Van (vanto@microsoft.com).

8. Set the **Report parameters**. Each parameter can be set by selecting the drop-down box next to the Parameter name and choosing **Enter value**. You will then be able to provide a selection of available values to set.

- Choosing a Content Manager will filter the other Parameters, Service, and Assignee. Choosing a Service will also narrow the Assignee Parameter to assignees that have an open Issue or PR under that Service.

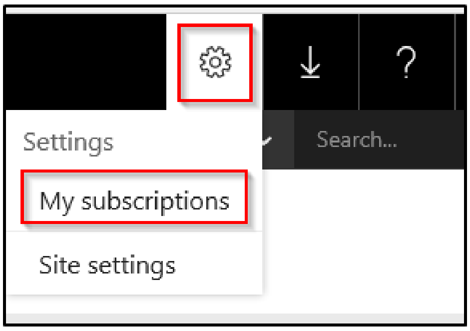


> **WARNING**
>
> If any of the report parameters have null values, the whole subscription will fail to run. An example of this is choosing to subscribe to the full report, and specifying a Content Manager or Service that does not have any open GitHub Issues for the execution of the report. Consider creating a subscription for the GitHub issue report separately. If the subscription does fail, it will mean that there were no GitHub Issues for that particular service when the subscription ran. This only applies to GitHub Issues, as PR reports will still run with null values.

9. Select the **Apply** button at the bottom of the new subscription to create your subscription.

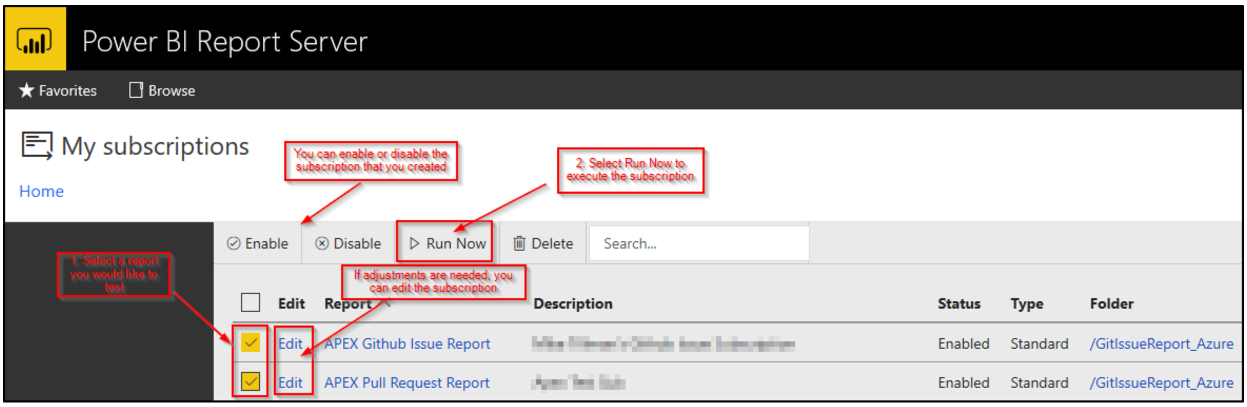
Managed and test your subscription

1. To manage your subscription, open the **Settings** (gear wheel) icon on the upper right-hand corner of the Report Portal, and select **My subscriptions**.



2. On the **My subscriptions** page, you can:

- Select your report, and run it immediately.
- Edit your subscriptions.
- Enable, disable, or delete your subscription.
- Check the status of the subscription, the last run time, and the results.



Once you are done with the settings and testing, that should be it! You should now get automated emails to the report with little maintenance.

Known issues

- If you try to render any of the reports that do not have GitHub Issues associated with the report, there will be no values picked up in the Parameter's data source. The report will fail to render, as the Multi-value Parameter must not be null. For example, if you try to render the Azure Full Report for a particular Content Manager or Service, and there are no Assignees Parameter available (because there are no open Issues under that Service), the report will not render correctly.
  - You can get around this issue by running only the report for GitHub Issue, or GitHub Pull Request.
- If you are running into an issue with your browser caching your session, and returning blank parameters,

you can get around this issue by checking all parameters and making sure that they reset by selecting all values.

How to host a hack-a-doc

3/5/2021 • 17 minutes to read

This article describes two different approaches to organizing a hack-a-doc. We coin the term hack-a-doc as a collaborative, focused event to create, review, and edit Docs to achieve immediate impact on a stated business goal.

The first case study is a hack-a-doc designed to reduce support pain by improving documentation for the Power BI service. The second case study is a hack-a-doc designed to improve outdated documentation (freshness) for the Azure HDInsight service.

There is an attached PowerPoint that describes best practices for planning and hosting an event.

> **NOTE**
>
> Include the Global Experiences team in your Hack-a-doc planning so they can properly forecast their localization efforts. See Localization contacts to find the PM owner for your content area.

Case Study 1 - Power BI support hack-a-doc

The Power BI content team hosted a five-hour hack-a-doc and closed 79 (out of 108) doc bugs. This story summarizes the experience and guidance to run your own hack-a-doc.

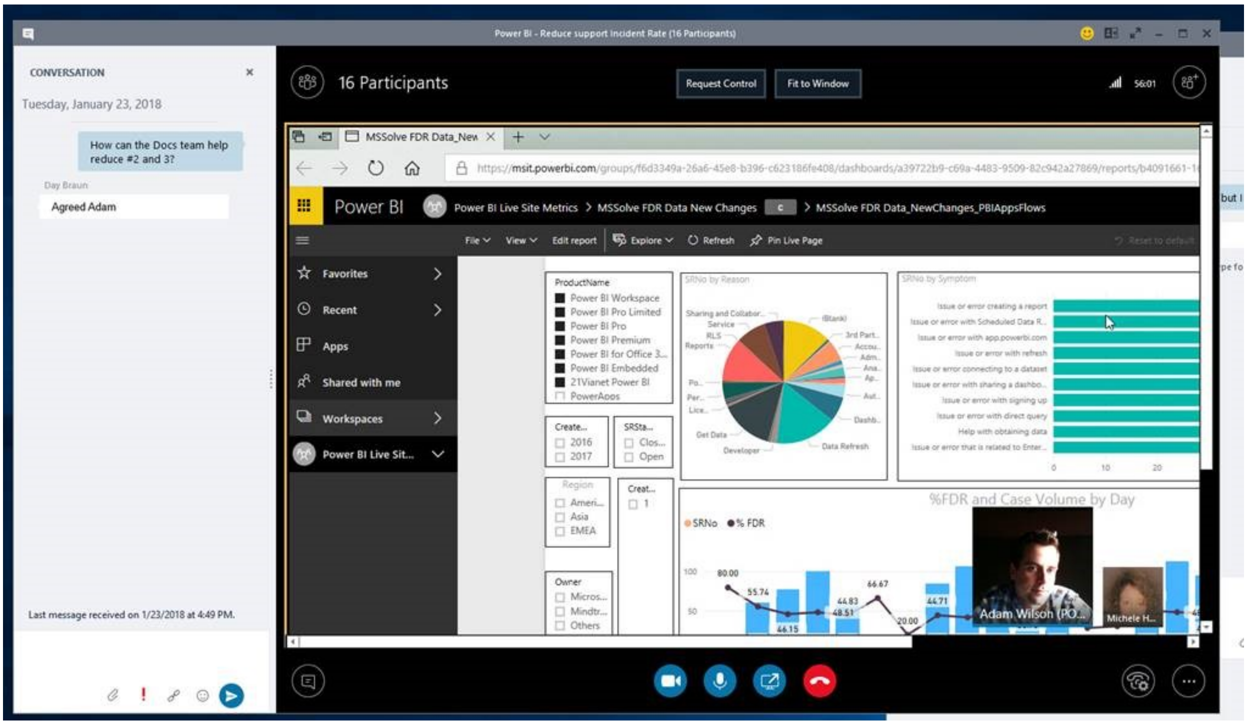
Background

The Power BI hack-a-doc (HaD) focused on deflecting support volume, but hack-a-docs can be used in many other situations: closing bugs, improving freshness, getting consensus on mockups, prioritizing work, and more. The beauty of a hack-a-doc is that it happens in real time and coordinates input from a large group of people across multiple organizations.

The Power BI hack-a-doc Support teams (CSS, Supportability, Serviceability) and Dev team (PMs, developers, Marketing) were faced with an overwhelming amount of support-related work. In many cases, more hours of “support” ticket work than work hours in a week. There were also an incredibly high volume of doc comments, Support site activity, Disqus comments, doc feedback, and Community Forums questions. The Support overhead was crippling the organization.

The Dev team undertook a study to look at the metrics for the high volume. The research number that got our attention was 23% of the volume could be reduced with documentation and user education. (later research showed volume as high as 40%).

Row Labels	Count of ServiceRequestNumber	Count of ServiceRequestNumber2
Gateway built-in Diags	135	16.11%
DOCUMENTATION	125	14.92%
USER EDUCATION	91	10.86%
BETTER ERROR	57	6.80%
External	49	5.85%
Customer Abandoned	45	5.37%
Still active	44	5.25%
External -Authentication	27	3.22%
advisory	23	2.74%
(blank)	22	2.63%
MFA Enablement	19	2.27%
Customer Resolved	17	2.03%
TESTING	17	2.03%
DCR	17	2.03%
COMMS	11	1.31%
DISCOVERABILITY OF DOC	11	1.31%
BUG	11	1.31%
Self service - Whitelisting	9	1.07%
PBIT	9	1.07%
Unknown	8	0.95%
User Error	8	0.95%
Clear token/cache options	7	0.84%
Content Pack Reliability	6	0.72%
Intermittent issue - Self Resolved	6	0.72%
User resolved	6	0.72%
External - Connectivity	5	0.60%
Content Pack Authentication	4	0.48%



Prep work

The writing team met with members of the Support and Dev orgs and discussed plans and strategies. One of these plans was to take the 23% tickets out of the proprietary tool, anonymize them, and create a spreadsheet. Few Microsoft employees have access to ticket information because of compliance and security restrictions, the help of the Support org was critical.

73	1248953	117100416444949	websites.net/?type	Audit Log Access Delay	Y	Explicitly note that it can take up to 48 hours for audit log			DISCOVERABILITY OF DOC
74	1219274	117100816460245	websites.net/?type	Orphaned Content Packs cannot	N	Update article to note that if the app		/service-organizat	DOCUMENTATION
75	1235153	117101016471680	websites.net/?type	Resource Not Allowed - SPO cus	N	Document error message" Correlation id: undefined, http			DOCUMENTATION
76	1249444	117101016473107	websites.net/?type	Advisory - Filtering import query		Unable to quickly find documentation on use of paramete			USER EDUCATION
77	1249556	117101091305201	websites.net/?type	Data Import - From Folder					USER EDUCATION
78	1251155	117101116474400	websites.net/?type	Gateway upgrade					USER EDUCATION
79	1219127	117101216485120	websites.net/?type	Timeout on refresh	Y	N/A	User Effort	esh-troubleshootir	DOCUMENTATION
80	1229056	117101216486385	websites.net/?type	Advisory - Design	N	How To: Options for storing and displaying images in Pow			DOCUMENTATION

Create Azure Boards bugs

The support items in the spreadsheet were converted into Azure Boards doc bugs by one person from Supportability and one person from Content and assigned to writers. To streamline, we used an existing chart that identified writers by their content areas.

	PM	Area	Notes	Writer	CSS	Marketi	Dev	UX	EE	K
15	Chris F	Data refresh		David	Didier/Bal	Jacob Grir	Doina B	mattlo	Brian Schumer's team	
16	Chris F	Email subscriptions		Michele	Didier/Ra	Jacob Grimm			Brian Schumer's team	
17	Chris F	Email subscriptions		Michele	Bala	Jacob Grir	parianoo	mattlo	Brian Schumer's team	
18	Chris F	HOME onboarding		Michele	Didier/Ra	Jacob Grimm			Brian Schumer's team	
19	Chris F	Pro trials		Mark?	Didier	Jacob Grir	parianoo	mattlo	Brian Schumer's team	
20	Chris F	query strings		Michele	Didier	Jacob Grir	parianoo	mattlo	Brian Schumer's team	
21	Chris F	real-time streaming datasets		David?	Tushar	Jacob Grir	parianoo	mattlo	Brian Schumer's team	
22	Chris F	Usage metrics		Michele	Tushar	Jacob Grir	parianoo	mattlo	Brian Schumer's team	

and a page we created in the CSS Power BI wiki that had links to Azure Boards bug templates by author and/or content area.

So a single-click to open a **pre-populated** bug template.

Create a VSTS documentation ticket [edit]

Use the following templates to create a **new documentation article** request or a **documentation bug**. Change the **Iteration** field to reflect the due date a team -- for example, preface the title with **CSS**-<Bug Title>.

Area	Create	Default owner
Analysis Services - Azure, SSAS, PBIPrem	New article or Doc bug	Owen Duncan
DAX Reference	New article or Doc Bug	Owen Duncan
Power Query/M Reference	New article or Doc Bug	Owen Duncan
Power BI Desktop	New article Doc Bug	David Iseminger
Power BI service	New article Doc Bug	Michele Hart
Power BI Premium, Admin, gateways	New article or Doc Bug	Michael Blythe
Power BI Dev (including PBIE and CV)	New article Doc Bug	Mark Ghanayem
Power BI Report Server/Reporting Services	New article or Doc bug	Maggie Sparkman, Mark Ghanayem
Power BI mobile	New article Doc Bug	Maggie Sparkman
Power BI Miscellaneous	Doc bug or request for new article	Power BI content team

These items will be assigned to a doc owner to work on. When an item is created, or updated, the assigned owner will be notified by an email notification.

There were some bugs that were hard to assign, so we used our weekly PM:UE Sync meeting to fill in the blanks. Additionally, we had one writer act as triage. If a PM or writer had an issue with an assigned bug (such as wrong owner or not enough info), the triage person was responsible for finding the right owner or working with Support to dig into the ticket for more info.

Bugs:

- Created in Technical Content database and we ensured all parties had access
- Used already-created doc bug templates that had as many fields pre-populated as possible, including writer, iteration, Problem description, and problem solution.
- Created tags that we could use in queries and metrics: “Support Volume” assigned to all bugs, “Support Solution” assigned to lower-pri bugs that already has a solution as part of the bug.
- Worked with PM team managers to come up with a list of content area tags (remember, the PMs and Devs are the specialists so these tags will help them the most) and added these additional tags (for example gateway, refresh, connectors, DAX) to help participants pick bugs to work on
- Updated spreadsheet with Azure Boards IDs
- Created Azure Boards queries for “Support Volume” and “Support Solution.”

In total, the spreadsheet recorded 840 Support tickets, and 140 were identified as being solvable with docs and UE, and 108 were able to be converted to Azure Boards doc bugs. Not all, because many support tickets just didn’t have enough information to create an actionable doc bug.

Hack-a-doc idea envisioned vs reality

There was team-wide urgency and pressure to get these bugs closed quickly. Erin suggested a Hack-a-doc.

We started with assumptions and plans that we quickly realized would not work. In the final week, we completely redesigned the HaD.

Ideas we ended up dropping

1. Logistics – set up triads of people (CSS + PM + writer) in advance, and pre-assign bugs. Dropped because attendees would be coming and going, many PM "owners" were away at conferences, too much up-front work. We wanted this experience to be *frictionless*.

2. For each HaD bug, assign the "owner" PM and writer. Dropped because not all PMs could attend and/or assign replacements. Didn't fit our goal of *frictionless*.

3. Do the edits in real time using **our** tools and process (such as Git), which meant sending installation/setup instructions ahead of time. Rejected because using our tools proved to be a barrier to participation for everyone but writers.

In my time at Microsoft I've been involved in at least two other hack-a-docs that I can recall, even spent time creating and prioritizing lists of articles for these hack-a-docs, and neither was successful....no one showed up. So we were nervous.

Sample email invitation

> Hello, thank you for participating in our hack-a-doc. We will be there at 7 a.m. to work with people in advanced time zones, but those members attending in person are welcome to start at 8am (late arrivals are still welcome).
>
> **The high number of support calls and posts and tickets is putting a large strain on the Power BI organization. This hack-a-doc is intended to help reduce that strain by adding oft-requested information to the Power BI docset.**
>
> This Azure Boards query returns the first set of bugs that we'll tackle at the hack-a-doc. These bugs were collated by the Serviceability team and categorized as issues that could have been solved with documentation. For the hack-a-doc, we'll create ad-hoc teams as people arrive in the room or online. Each two-person team will consist of one writer and one non-writer who will work together to resolve bugs, update existing Power BI docs, and write new docs if necessary. RSVP as soon as you can.
>
> If you are in the Redmond area, we'd prefer that you attend in person....THERE WILL BE FOOD!
>
> To prepare for the hack-a-doc, you may want to do the following tasks. This is optional.
>
> 1. Sign up for GitHub, it's quick and free.
> 2. Go to https://repos.opensource.microsoft.com and join MicrosoftDocs (if you haven't already joined).
> 3. Open the Azure Boards query.
> 4. Review the list of bugs and pick out some that fall into your area of expertise. For example, look at the bug title and tags and if those descriptors don't help, open the bug itself. Come to the hack-a-doc with a list of bugs you'd feel qualified to work on.
>
> For remote employees, we'll still put you on teams and create Skype meetings, but won't be able to provide food.
>
> I'll send out more information over the next eight days. Please forward the meeting invite to anyone I may have missed.
>
> Thank you for helping to reduce support volume, Michele (remote, no food for me), and Michael (on-site Hack-a-doc facilitator)

Identify participants

Start with your PM, Dev, Support, Marketing, and other partners and network out from there. Contact managers to help identify invitees. Continually ask invitees to suggest others who would be an asset to the effort. If someone can't make it, ask them to recommend a replacement. In spite of this effort, you'll inevitably leave people out and miss people...we did. Apologize and ask if you can follow up with them after.

Hack-a-doc details

As early as possible (and repeat), communicate the goals of the Hack-a-doc and how you expect to achieve them. Entice people with food and prizes. Make the experience as easy as possible - limit pre-work, requirements, processes. Communicate often and using different tools and strategies, such as messaging or calling individuals directly, meeting with managers so they're onboard and a partner in achieving the goal, and mentioning in other meetings.

Schedule to accommodate different time zones and allow attendees to come and go (or participate even if they only have one hour available).

For our team and audience, we quickly realized that since we weren't going to have participants in the same room for the entire five hours, creating working teams ahead of time would not work. We switched to a more free-flowing ad-hoc model and communicated far and wide.

Our reasoning was that only the PMs are Power BI “specialists” but the Support folks have a breadth of knowledge and could work on almost any bug. Same for writers, writers can write about anything as long as they have a knowledgeable partner. So we moved from pre-assigned teams of three to ad-hoc duos. And we added a facilitator in the room who assigned duos as people arrived in person or on Skype. When anyone was free, they posted on Skype and let the facilitator know. We left the Skype meeting open for conversations and messaging.

We did little doc editing in real time because it slowed the process of collecting solutions. Instead, we focused on getting the experts to the table and writing down the ticket solutions any way possible. In some cases, we edited the docs directly in VS Code (Fantastic). In some cases, we edited the docs in the browser (Great). And, in most cases, we took notes in the Azure Boards ticket, in Word, in email, and in OneNote (perfectly wonderful). For the latter method, the writer in the duo was responsible for claiming ownership of the bug in Azure Boards, adding the notes, and then reassigning the bug back to the doc owner. Any notes outside of Azure Boards were added to the Azure Boards bug before assigning back to the doc owner.

After the Hack-a-doc event

Once the actual hack-a-doc was done, writers concentrated on applying the solutions to the docs, closing the bugs, and pushing out PRs. The hack-a-doc was on a Wednesday, and we had a goal to close the bugs we worked on within two business days (if they weren't already closed). Because the doc bugs for the hack-a-doc were created by only two team members, it was their responsibility to set the bugs to closed once the doc owner marked the bug as complete. Work continues on the remaining few.

The prize was awarded using the participant list and a randomly generated number. A follow-up email was sent to all participants thanking them for their help before, during, and after the hack-a-doc. A post-mortem was held by the writing team and, less-formally, with Support and Product teams.

What next?

Ensure that you have processes in place to measure the success of your hack-a-doc.

1. track tickets - volume, category
2. track "tree" solutions - how many involved docs? How many involved a doc with a "Support" tag?
3. Track volume of Comments for each updated doc
4. Redo initial investigation of support tickets categorized by solution. Are docs and user education still 23% or more?
5. From Support site - track keywords, paths, doc hits (especially docs updated during hack-a-thon)
6. In many cases, tickets were identified as having a doc solution but user did not even search. Or, user searched but results didn't surface correct docs. This information should guide our next effort - SEO and redesign of Support site (already partially completed) so customers are unable to submit a ticket without at least searching, seeing a list of docs that might answer the question.

Key takeaways

1. Freeform ad-hoc worked better than structured
2. Having a facilitator

3. Letting participants know that their attendance for the full 5 hours is not required – people will come and go
4. Limit the requirements - we started with many (for example sign-up for our repo, install Git and markdown tools, review all bugs and add your name in Azure Boards) and ended up removing them ALL. They were barriers to participation.
5. Only the PMs and Devs are "specialists" so structure the hack-a-doc around them
6. Don't have someone else set up the meeting invite. We were not able to make changes to the logistics and to attendees getting invited at different times and receiving different invitation details.
7. Communication: email, call, IM over and over again. As much 1:1 as possible, no matter the time zone. Explain the win: win benefit, entice with food and prizes, emphasize that managers and up are supporting this event and even attending, ask for substitutes if someone is unable to attend, follow-up, remind. Repeat and repeat.
8. Expect pushback and be ready with the value prop. Win: win. "Yes, you're overloaded, that's one of the reasons we're doing this event. Just commit to an hour or two of your time and together we can make this better." *Hi so-and-so, Just confirming that you will be attending the hack-a-doc on May 30. Sounds like you'll be a key member of the project. Are there other PMs and Devs you recommend who could join us and help with these types of doc updates? Is there a particular CSS Engineer you work with?*

Wins (in addition to closing 79 bugs)

The Hack-a-doc was a wonderfully collaborative experience with members of CSS, PMs, Devs, Supportability/Serviceability, Marketing, and writers working in ad-hoc teams to tackle a problem. In addition to closing bugs, we forged new relationships across teams and, I believe, developed more appreciation for the role we each play in making Power BI successful.

- Improved Support tickets with "trees" being used again
- Improved Support tickets with new categories created to align better with PBI components (another level of dropdowns)
- Increase in PMs and Support contributing/writing docs and submitting doc bugs
- Invited to Support monthly presentations to teach engineers how to update docs.
- Complete engagement between Content and CSS with CSS committing to not only improved bug details BUT to submitting doc bugs, cutting out the middleman (Devs, PMs). CSS is giving a $50 monthly prize to the engineer who opens the most doc bugs in Azure Boards or who makes the most doc updates using the browser version of GitHub. Basically, one writer tracks CSS contributions to docs and picks a winner, communicates that to Support Manager (Day Braun) who awards the gift card.

Case Study 2 - HDInsight docs freshness hack-a-doc

Several content developers planned a one-day hack-a-doc event for our partner product group PM team. After several months with no assigned resources on a technology area, Azure HDInsight, we found 89 of the top 100 HDInsight docs were outdated, that is abandoned or stale. We needed to identify a way to get help from experts that know the product fairly well and can help update the docs with the latest revisions to freshen them up.

In the one day event, we made a huge dent on our HDInsight doc freshness by reviewing 69 docs, editing 65 of those docs during the session via pull requests. We reached 75% of the high priority stale doc pages in our to-do list.

Best practices we followed:

1. **Set a theme that is focused and general enough to be inclusive** – our stated goal was to "Improve doc freshness" with a secondary goal of "shift PM culture to contribute to docs more often"

2. **Divide and conquer the planning effort**, so that one person doesn't have to handle all the stressful planning. Someone handle logistics and food purchasing, someone else handle planning and communication with stakeholders, someone else handle the instructional printouts and work assignment,

someone steer the group and offer feedback, etc.

3. **Get the buy-in from the management** on your team and the partner teams. Need to be sure that there is enough common interest to make the event worthwhile. Ask for time budgeting to make sure enough people can attend to make critical-mass, and financial budgeting to get money for food and swag.

4. **Ask the partner team's leadership to evangelize** the event ahead of time, and **request that they attend the event**. Having the team's GPM and their Partner PM director attend our event really kept people motivated to attend and contribute.

5. **Keep to-do lists and notes** in a simple, bite-sized way. A shared OneNote hosted in a Teams share worked well since everyone could type notes simultaneously and see the state of our to-do lists in near-real-time. Its light enough that we didn't fight with permissions or too many fields in DevOps items.

6. **Plan lists of priority docs to work on**. Use metrics to plan priorities (page views, customer satisfaction, bounce rate, etc.) Having some predefined categories of work sorted top-down in priority order in different OneNote tabs helped us match the subject-matter expertise of the attendees. Everyone had something to do without overlapping with each other.

7. **Book the right size conference room and the right location** that is near the team you are targeting. Easy location allows the team flexibility to come and go for high priority meetings. We had Teams phone call open for any remote participants. One of our content developers flew in for the week to help facilitate the event in person.

8. **During the event, do as much live editing of docs as possible.** Get the changes in Pull Requests to avoid creating an even longer backlog of work. Edit in the browser for light edits, using the private repo to get Acrolinx and preview builds. When the work was too much or too complex to finish, we used OneNote to making clear notes on what follow-up is needed, and track the follow-up work in Azure DevOps after the event was over.

9. **Celebrate often** – Every time someone did a pull request, we wrote their name on a raffle ticket into a hat, for a prize drawing at the end of the day. This activity became a competitive energy building tool. High energy and focus are important for events like this. We took a few photos to capture the memory as well.

10. **Measure results and communicate** - We counted our accomplishments (pull request numbers are easy to quantify in our OneNote lists), and will trend our CSAT and freshness metrics to show a clear before and after. We sent out summary emails to the team and their leadership to explain the effort and our results.

Case Study 3 – Cosmos DB docs content refactor hack-a-doc.

Azure Cosmos DB team had a hack-a-doc to refactor the API/sub-service specific content. It was a three-day event with audience from Content, PM, Dev & CSS teams. We were successful in splitting the content by API and fill in some known gaps. We are 60% complete on the items we tracked for this event. This hack-a-doc was remote only. In addition to the tips from other case studies, the following are some considerations if you are running a hack-a-doc over teams or in a distributed environment:

- Make sure to sync with the teams ahead to get the inventory of work items to tackle during the event. Assign owners and SMEs for different items ahead.

- In addition to heavy/medium weighted tasks, track some work items with minor edits. These tasks are helpful for first-time contributors to get started without much effort.

- If the hack-a-thon is for the whole day or a set of days, have a clear agenda for each day. Explicitly callout if you plan to hold a short session on how to contribute to docs and the time it's time slot. New contributors can make time for it.

- Offer the flexibility for team members to leave the call and go back to their desks to work on the content. They can also have offline meetings with a smaller group of SMEs while making the content updates.
- Keep a track of the Teams conversation to address any questions. It's best to use the same Teams link if your hack-a-thon is multiple days as the conversation history is saved.
- Have progress check-ins twice a day to see how things are going, and if there are any blockers.
- Use 30 min at the end of the event for show and tell, where all the contributors can show their content updates and PRs.
- After the event, measure the success and celebrate. Get the metrics on number of files added, updated in the release branch, and items completed in the spreadsheet. Send a summary email with the highlights, and learnings.
- You can offer swag to the top contributors. Work with the DevRel admin team to arrange the swag, get the shipping labels printed, and ship the swag from one of the Microsoft buildings where shipping facility is available.

Using the Docs experimentation platform to improve my articles

6/16/2021 • 2 minutes to read

WARNING

The content on this page has been deprecated. Contact Sara Lerner for information about how to do user research, including A/B testing, using UserTesting.com.

Docs experimentation platform

Docs has a built-in experimentation platform allowing authors to test in-page article changes with a subset of their audience. All you need is to do is:

- Fill out an experiment form on the CGA site.
- Create one more .md file with the changes and save it in the same directory in GitHub as the original one.
- Add some metadata to both articles.

Once you start the experiment, based on the configuration you provided, a specific percent of users will see the alternative article until the system records the configured number of page views. For example, 1000 page views. The experiment automatically stops and all users only see the original article.

During and after the experiment, you can see all the stats:

- Verbatim comments.
- LiveFyre comments.
- Ratings on the Microsoft Docs Metrics tool (MDM).

You can use the metrics reported in MDM to determine if you want to go mainstream with those changes and stick with the original.

Finish instructions on learning when and how to run an experiment on the Docs platform are available in the following resources.

PowerPoint Deck:

- A/B testing for everyone, in 3 steps

If you have questions, email CGA Support.

What is an experiment?

What makes an experiment different than other types of data analysis are three parts: making a *controlled change* to the user experience, making a *prediction* about the effect (the hypothesis), and intentionally structuring *data capture* to measure the actual effect. Experimentation may sound intimidating, but is actually an approachable and intuitive way for non-data scientists to become more data-driven. Why? Because a well-designed experiment yields results that are easy to interpret. Simply comparing the predicted effect to the actual effect is where the learning happens. We invite you to give it a try!

The CSI data science team will partner with writers and PMs to enable the following tests:

- A/B tests
- Before vs. After tests
- Email-driven experiments
- Usability tests
- Surveys
- Click-pathing experiments

Docs Build validation overview

3/5/2021 • 4 minutes to read

Whenever you submit a pull request to contribute to docs.microsoft.com, Docs Build runs a set of validation rules to check the quality of the contribution. Some validations are built into the Docs Build engine. Some are "custom" validations that are configured in an external service and then integrated into Docs Build. The main difference is that custom validations can be configured into rulesets. Different rulesets can be applied to different Docs repos. Once applied to a repo, the user experience for custom validations is identical to the experience for other Docs Build validations: Results appear the same way in PR conversations and build reports.

For example, here's a PR that contains both Docs Build validation results (`Author cannot be recognized in GitHub` and `Invalid file link`) and custom validation results (`Missing attribute` and `Invalid value for ms.date`):

⚠ Validation status: warnings

File	Status	Preview URL	Details
articles/active-directory-b2c/active-directory-b2c-access-tokens.md	⚠ Warning	View	Details

articles/active-directory-b2c/active-directory-b2c-access-tokens.md

- [Warning] `Missing attribute: either ms.prod or ms.service is required. Use ms.prod for on-premise products, or ms.service for cloud services.`
- [Warning] `Invalid value for ms.date: '04/160/2019'. Must be a date in format MM/DD/YYYY, no more than 30 days from today.`
- [Warning] `Author mmmacy cannot be recognized in GitHub.`
- Line 18: [Warning] `Invalid file link:(~/articles/active-directory-b2c/active-directory-b2z-reference-tokens.md).`

For more details, please refer to the build report.

If you click the **build report** link from the GitHub comment, you'll see your PR results and any outstanding results in the repo. Many rules include error codes linked to detailed documentation. To provide doc feedback, click the speech bubble next to the error code to file a GitHub issue in the Docs Feedback repo.

Severity	Type	Code (click link for details)	File	Message
Warning	User	ms-prod-or-service-missing	articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Missing attribute: either ms.prod or ms.service is required. Use ms.prod for on-premise products, or ms.service for cloud services.
Warning	User	ms-date-invalid	articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Invalid value for ms.date: '04/160/2019'. Must be a date in format MM/DD/YYYY, no more than 30 days from today.
Warning	User	GitInfoNotFound	articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Author mmmacy cannot be recognized in GitHub.
Warning	User	InvalidFileLink	articles/active-directory-b2c/active-directory-b2c-access-tokens.md	Invalid file link:(~/articles/active-directory-b2c/active-directory-b2z-reference-tokens.md).
Warning	U: Click to provide feedback for Code or Message	ms-author-invalid	articles/cosmos-db/how-to-use-…ns-gremlin.md	Invalid value for ms.author: 'olignat' is not a valid Microsoft alias.
Suggestion	User	h1-missing	articles/cloud-shell/persisting-shell-storage.md	H1 is required. Use a single hash (#) followed by a space to create your top-level heading.

Currently, custom validation is supported for English conceptual Markdown articles in Docs Build v2 and Docs Build v3. A subset of custom validation is available for Learn included Markdown files in Docs Build v2, but not yet for Learn YAML files. Custom validation for reference is expected to be available in FY21 after reference migrates to Docs Build v3. Custom validation isn't available for localized repos because content issues should be fixed in the English source.

A new severity, "Suggestion", has been implemented to help roll out new validations. New validations are initially released as Suggestions, which don't impact build or block workflow automation such as PRMerger. After eight weeks they'll be escalated to Warnings, which don't block builds but do block PRMerger. If a new validation has unusually high impact and more time is required for initial cleanup, we might extend this period.

Custom validation rulesets include *metadata rules* and *content rules*.

Metadata rules

Metadata rules check for the presence and validity of various attributes, mostly used for content reporting. Different rulesets have been defined for different content areas. For example, some metadata requirements are global across the Docs platform, while others only apply to repos that use certain content reporting features.

Metadata can be set in multiple ways: specified directly in the Markdown file, specified at a folder level in docfx.json, or specified globally in docfx.json. Validation respects this hierarchy and returns results appropriately.

Metadata rules aren't yet available for Learn included Markdown files.

Content rules

Content rules check for common Markdown mistakes that cause Docs site problems, including broken content, accessibility bugs, and localization problems. Most content rules are global across the Docs platform.

Custom rulesets

There are two main custom rulesets in use for conceptual Markdown content:

- The Docs Platform ruleset, which is required for all repos unless an explicit exception is granted.
- The Developer Relations/CGA reporting ruleset, which is a superset recommended for all conceptual repos and required for repos that use CGA reporting.

The Docs team can create alternative rulesets for teams outside of Developer Relations only if the teams have different reporting requirements from Developer Relations. These exceptions must be approved by the Docs Governance Committee.

Docs Platform ruleset

The Docs Platform ruleset checks that an accountable owner is available for each article on Docs, and that basic content standards are met. It's applied to all English conceptual repos by default.

The Docs Platform ruleset contains the following rules:

- author-missing
  - Additionally, native Docs Build checks that the value of `author` is a valid GitHub ID.
- h1-empty
- h1-in-moniker
- h1-missing
- h1-no-space
- h1-not-first
- hard-coded-locale
- insecure-link
- ms-date-invalid

- multiple-h1s
- title-missing

Developer Relations/CGA Reporting ruleset

The Developer Relations/CGA Reporting ruleset contains the following rules:

- author-missing
  - Additionally, native Docs Build checks that the value of `author` is a valid GitHub ID.
- deprecated-attribute
- h1-empty
- h1-in-moniker
- h1-missing
- h1-no-space
- h1-not-first
- hard-coded-locale
- insecure-link
- ms-author-invalid
- ms-author-missing
- ms-date-invalid
- ms-date-missing
- multiple-h1s
- ms-prod-and-service
- ms-prod-missing
- ms-prod-or-service-missing
- ms-prod-technology-invalid
- ms-service-missing
- ms-service-subservice-invalid
- ms-topic-invalid
- ms-topic-missing
- title-missing

To request onboarding to the full Developer Relations/CGA Reporting ruleset, file a work item via this template.

Learn ruleset

The Learn ruleset only runs on included Learn Markdown files. Metadata validation isn't yet supported for Learn, so only content rules that apply to included files are included.

- hard-coded-locale
- insecure-link

Other custom rulesets

Alternative rulesets have been enabled for the Office Developer, Marvel, and Protocols teams. There's also an "approved-exceptions" rulesets for non-conceptual repos that had unexpected results with the Docs Platform ruleset.

To request onboarding to one of these rulesets, file a work item via this template. Be sure to specify which ruleset is appropriate for your team.

For more information about alternative rulesets, contact mbradley.

allowlists

Some validations, including metadata validations for `ms.prod`, `ms.service`, and `ms.topic`, check values against an allowlist. You can view allowlists on on the Taxonomy lists page.

To request new allowlist values, follow the process described in Metadata changes.

PR only validation rules

11/2/2020 • 2 minutes to read

A challenge for rolling out automated validation rules has been that Docs repos contain many thousands of articles, many of which aren't actively maintained. To turn on a new validation rule, we used to have to fix the issue in all existing files to avoid too much noise in the repos. This meant we often missed catching issues in new and updated content because the impact of reporting on old content was too high.

We now have the ability to turn on rules as "PR only". This means results only appear in validation reports if the issue was found in files in a PR. Older files aren't checked until they're touched. This allows content contributors to focus on the content they're working on now, without having to worry about dormant content.

For rules enabled as PR only, such as description-missing, you'll see the following validation behavior in PR comments and build reports:

- In the PR comment, you'll see the PR only result like any other validation:

File	Status	Preview URL	Details
help-content/contribute/validation-ref/alt-text-bad-length.md	Suggestion	View	Details

  help-content/contribute/validation-ref/alt-text-bad-length.md

  - Line 2, Column 1: [Suggestion-description-missing] `Missing required attribute: 'description'. NOTE: This Suggestion will become a Warning in late October, 2020.`

  For more details, please refer to the build report.

- If you click the **build report** link from the PR comment, you'll see the full build report for the repo. In the **Validated Files** table, you'll see the files in the PR and any validation results, including PR only results:

  Validated Files

File	Status	Preview URL	Details
help-content/contribute/validation-ref/alt-text-bad-length.md	Suggestion	View	• **Line 2: [Suggestion]** Missing required attribute: 'description'. NOTE: This Suggestion will become a Warning in late October, 2020.

  The **Details** table in the build report shows all outstanding Errors, Warnings, and Suggestions in the repo, *not* including PR only rules.

- If you run a full branch build, for example by forcing a build via Docs Portal, the build report won't contain any PR only results.

If you'd like to fix all issues in your content including PR only rules, you can request a test report by contacting mbradley.

alt-text-bad-length

3/5/2021 • 2 minutes to read

Applies to: All content types

This validation catches alt text that's outside the recommended range of 40 - 150 characters (these values are configurable). Because this range is currently a guideline and not a hard requirement, this validation hasn't been enabled on a production ruleset for Docs or Learn at this time. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Alt text '{text}' is out of range at {length} characters. Alt text should be between {min} and {max}
characters.
```

The min and max values are configurable, but 40 - 150 (inclusive) is recommended. Alt text of fewer than 40 characters is generally not descriptive enough to meet Accessibility guidelines, and alt text longer than 150 characters is clumsy for screen readers. Although these aren't hard requirements, running this rule on your repo can give you an overall picture of how good your alt text is, and help identify areas for improvement.

For more information about accessibility requirements for images, see Accessibility guidelines for multimedia.

Resolution

The following examples show how to add alt text for images in standard Markdown and in the Docs `:::image:::` Markdown syntax with `type='content'` (for standard conceptual images):

```
Standard Markdown image:

![Screenshot showing how to force a build from the Docs Portal by clicking Manual Publish, specifying a
branch, then clicking Publish.](media/force-build.png)

Docs Markdown content image:

:::image type="content" source="media/force-build.png" alt-text="Screenshot showing how to force a build
from the Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.":::
```

If your image is too complex to meaningfully describe within 150 characters, you should use the Docs `:::image:::` Markdown syntax with `type='complex'`. For complex images, the alt text should be a high-level description of the image. The image content should describe in detail what the image conveys:

```
:::image type="complex" source="../media/page-views.png" alt-text="Graph showing page view trends.":::
   <Detailed description of the page view data shown in the graph goes here...>
:::image-end:::
```

If your image is purely decorative, such as a product icon, alt text isn't required, but you must use the Docs `:::image:::` Markdown syntax with `type='icon'` to indicate that the image is decorative:

```
:::image type="icon" source="../media/docs-icon.svg":::
```

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

alt-text-bad-value

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

`Alt text for an image can't be the same as the image file name.`

Image alt text must meaningfully describe the content and intent of the image. Using the name of the image as alt text provides no additional information and isn't allowed.

Resolution

Alt text must accurately describe the purpose of the image in at least 10 and no more than 250 characters. Between 50 and 150 characters is recommended. The following examples show how to add alt text for images in standard Markdown and in the Docs `:::image:::` Markdown syntax with `type='content'` (for standard conceptual images):

```
Standard Markdown image:

![Illustrates how to force a build from the Docs Portal by clicking Manual Publish, specifying a branch,
then clicking Publish.](media/force-build.png)

Docs Markdown content image:

:::image type="content" source="media/force-build.png" alt-text="Illustrates how to force a build from the
Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.":::
```

If your image is too complex to meaningfully describe within 250 characters, you should use the Docs `:::image:::` Markdown syntax with `type='complex'`. For complex images, the alt text should be a high-level description of the image. The image content should describe in detail what the image conveys:

```
:::image type="complex" source="../media/page-views.png" alt-text="Graph showing page view trends.":::
   <Detailed description of the page view data shown in the graph goes here...>
:::image-end:::
```

If your image is purely decorative, such as a product icon, alt text isn't required, but you must use the Docs `:::image:::` Markdown syntax with `type='icon'` to indicate that the image is decorative:

```
:::image type="icon" source="../media/docs-icon.svg":::
```

For more information about accessibility requirements for images, see Accessibility guidelines for multimedia.

> **TIP**
>
> To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

alt-text-missing

3/5/2021 • 2 minutes to read

Applies to: All content types

Suggestion

`Missing alternate text for image '{image path}'. Alt text is required for accessibility.`

Screen readers rely on alternate text so that users with visual impairments can hear a meaningful description of an image.

Resolution

If your image is intended to convey meaning, you must provide alt text. The alt text must accurately describe the purpose of the image in at least 10 and no more than 250 characters. Between 50 and 150 characters is recommended. The following examples show how to add alt text for images in standard Markdown and in the Docs `:::image:::` Markdown syntax with `type='content'` (for standard conceptual images):

```
Standard Markdown image:

![Illustrates how to force a build from the Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.](media/force-build.png)

Docs Markdown content image:

:::image type="content" source="media/force-build.png" alt-text="Illustrates how to force a build from the Docs Portal by clicking Manual Publish, specifying a branch, then clicking Publish.":::
```

If your image is too complex to meaningfully describe within 250 characters, you should use the Docs `:::image:::` Markdown syntax with `type='complex'`. For complex images, the alt text should be a high-level description of the image. The image content should describe in detail what the image conveys:

```
:::image type="complex" source="../media/page-views.png" alt-text="Graph showing page view trends.":::
   <Detailed description of the page view data shown in the graph goes here...>
:::image-end:::
```

If your image is purely decorative, such as a product icon, alt text isn't required, but you must use the Docs `:::image:::` Markdown syntax with `type='icon'` to indicate that the image is decorative:

```
:::image type="icon" source="../media/docs-icon.svg":::
```

For more information about accessibility requirements for images, see Accessibility guidelines for multimedia.

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

author-missing

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Missing attribute: author. Add the current author's GitHub ID.`

The `author` attribute identifies the author of the article by GitHub ID.

Resolution

Add the current author's GitHub ID to the YML header:

```
---
author: meganbradley
ms.author: mbradley
---
```

The value should be the *current* owner of the article, not the original author if ownership has changed.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

author-not-found

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

`Invalid value for author: '{author}' is not a valid GitHub ID.`

The value specified for author must be a valid GitHub ID.

Resolution

Update the `author` value with the current author's valid GitHub Id. We recommend that the author is a full-time employee, rather than a short-term vendor.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

bookmark-not-found

11/2/2020 • 2 minutes to read

Applies to: All content types

Warning

`Cannot find bookmark '{bookmark-id}' in '{parent-file}'.`

You're trying to link to a heading in the current file or another file that doesn't exist.

Resolution

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Verify the heading you want to link to and update the link.

To link to a section in the current article, use a hash symbol, followed by the words of the heading. Remove punctuation from the heading and replace spaces with dashes.

```
[Managed Disks](#managed-disks)
```

To link to a heading in another file, use a relative link to that file, followed by a hash symbol and the words of the heading. Remove punctuation from the heading and replace spaces with dashes.

```
See [the Resolution section in h1-empty](h1-empty.md#resolution).
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

code-block-indented

4/27/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Indented code blocks aren't allowed. Use a Markdown code block surrounded by triple backticks.`

Code blocks created by indentation aren't allowed on Docs, because you can't specify a code language for correct colorization in indented code. Indentation also isn't allowed as a way to create other block text, because indented text renders as code and isn't localizable.

Resolution

If the indented text is intended to be a code block, use a Markdown codeblock with a valid devlang instead:

````
Before:

        public static void Log(string message)
        {
            _logger.LogInformation(message);
        }

After:

    ```csharp
    public static void Log(string message)
    {
        _logger.LogInformation(message);
    }
    ```
````

Supported devlang values are listed in the devlang allowlist.

For short lines of code, you can use inline code with single backticks:

```
Before:

        I want this to look like code.

After:

`I want this to look like code.`
```

If you indented a paragraph for emphasis, use an appropriate localizable style instead, such as a note or a quoted plain text paragraph:

```
> [!NOTE]
> Use an alert such as a note when appropriate.

"Use quotes if appropriate... Lorem ipsum dolor sit amet,
consectetur adipiscing elit. Etiam efficitur id orci ac euismod.
Cras risus justo, molestie ac nulla eget, porttitor aliquam
lectus. Aliquam eu ipsum vestibulum, tempus ante at, porta dui."
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

code-block-unclosed

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

````
Unclosed code block. Code blocks must begin and end with triple backticks (```).
````

Resolution

Close your code block with triple backticks:

````
```md
Here's a properly closed Markdown block.
```
````

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

column-header-missing

5/6/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Column headers are required for tables. Add appropriate header text, or remove any extra table cells that
cause an extra column.
```

Column headers are required for screen readers. Without headers, users who can't visually scan the whole table before reading won't know what to expect from each column.

Resolution

Add a header to each column that meaningfully describes the content of the column. The Markdown for a basic table with headers is as follows:

```
|header 1 |header 2 |header 3 |
|---------|---------|---------|
|content 1|content 2|content 3|
|and......|so.......|on.......|
```

In some cases, this message is caused by extra table cells, which create an extra column without a header. This case can be difficult to spot in Markdown. For example, this table looks pretty good at a glance:

```
|header 1 |header 2 |header 3  |
|---------|---------|----------|
|content 1|content 2||content 3|
|and......|so.......|on........|
```

But there's an empty cell in the first row, which creates a fourth and header-less column on the published page:

header 1	header 2	header 3	
content 1	content 2		content 3
and......	so.......	on........	

If you can't immediately see what's wrong with your Markdown table, check the file preview on review.docs.microsoft.com.

Data matrix tables

A header might not be appropriate for the left column of your table if the table is a data matrix - that is, a table with both a top header and a header column.

To create a data matrix table, leave the first column header blank and bold the text in the rest of the cells in the first column:

```
|                     |top header 1 |top header 2 |top header 3|
|---------------------|-------------|-------------|------------|
|**column header 1** |content 1a   |content 2a   |content 3a  |
|**column header 2** |content 1b   |content 2b   |content 3b  |
|**column header 3** |content 1c   |content 2c   |content 3c  |
```

TIP

The Docs Authoring Pack for VS Code includes a function to convert a regular Markdown table into a data matrix table. Just select the table, right-click, and select `Convert to data matrix table`.

What about custom layouts?

Note that the `:::columns:::` extension is no longer recommended for creating custom layouts. Many custom column layouts have been found to have accessibility issues or otherwise violate Docs style guidelines. You shouldn't create custom layouts. Use standard Docs features.

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

dependent-repo-branch-mismatch

3/5/2021 • 2 minutes to read

Applies to: All content

Suggestion

```
The referenced branch '{branch}' in dependent repo '{repo}' can't be found. Update the branch value of the
dependent repo in the .openpublishing.publish.config.json file.
```

Cross repository references can be used to pull content and other assets into the current repo from another repo. These references are configured in the .openpublishing.publish.config.json configuration file. This Suggestion means that the entry for the specified dependent repo specifies a branch that doesn't exist.

Resolution

Go to the dependent repo to confirm the branch name (usually the default branch, such as `main`). Then update the `branch` property for the repo under `dependent_repositories`:

```
"dependent_repositories": [
   {
     "path_to_root": "repo-name",
     "url": "https://github.com/MicrosoftDocs/repo-name",
     "branch": "main"
     }
   }
 ]
```

For more information, see Cross Repository Reference.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

deprecated-attribute

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`Deprecated attribute: ms.component. Use ms.subservice instead.`

Use `ms.service` to specify cloud services. To specify more detailed information about `ms.service`, you can optionally specify `ms.subservice`. Don't use `ms.component`; it's deprecated for this content.

Resolution

Confirm that your `ms.service` value is correct for your article. Then choose a valid `ms.subservice` value.

Valid values can be found in the ms.service allowlist.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

description-bad-length

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

This validation catches description values that are outside the recommended range of 100 - 160 characters. However, because this range is a guideline and not a hard requirement, this validation hasn't been enabled on a production ruleset for Docs or Learn at this time. If you would like this rule run on your repo, contact mbradley.

> **TIP**
>
> The recommended way to verify that your description will provide meaningful information to users when returned in search results is to use the Search results Preview available in the Docs Preview VS Code extension, which is part of theDocs Authoring Pack for VS Code.

Suggestion

```
Description '{text}' is out of range at {length} characters. Descriptions should be between 100 and 160
characters.
```

Resolution

Write a meaningful description between 100 and 160 characters.

For more information about how to write descriptions, SEO: How to write good meta descriptions.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

description-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Missing attribute: description. Add a meaningful description to be returned in search results. Descriptions
should be between 100 and 160 characters.
```

The `description` attribute is required both for SEO and for site search.

This rule has been enabled for PR only.

Resolution

Write a meaningful description between 100 and 160 characters.

For more information about how to write descriptions, SEO: How to write good meta descriptions.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

dev-language-invalid

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn Markdown

This validation is not yet enabled on production repos because the devlang taxonomy is still in progress. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Dev language '{language}' is not allowed. Specify a valid language slug from the devlang taxonomy. Language
values are case-sensitive.
```

Resolution

Supported devlang values are listed in the devlang allowlist. Add a valid language slug. To request new devlang values, email DevRelIA@service.microsoft.com.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

dev-language-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn Markdown

This validation is not yet enabled on production repos because the devlang taxonomy is still in progress. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Dev language missing. You must specify a valid language slug from the devlang taxonomy. Values are case-sensitive
```

Resolution

Supported devlang values are listed in the devlang allowlist. Add a valid language slug. To request new devlang values, email DevRelIA@service.microsoft.com.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

disallowed-heading

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Heading '{heading-text}' matches the text of a disallowed Docs template heading.
```

Docs generates some article headings, such as "Feedback" and "Is this page helpful?", automatically so that authors don't have to add them manually and users get a consistent experience across the site. If you manually add a heading that matches one of these auto-generated headings, you end up with duplicate headings - an accessibility violation. Screen readers rely on headings to navigate the page for users with visual impairments, so duplication causes confusion.

Resolution

Remove the duplicate heading. If the content of the heading is redundant with autogenerated content, remove the whole section. If the content is unique and required, rename the heading as appropriate.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

docs-link-absolute

4/28/2021 • 2 minutes to read

Applies to: All topic types

Suggestion

```
Absolute link '{URL}' will be broken in isolated environments. Replace with a relative link.
```

Absolute links to other Docs pages will be broken in environments that aren't connected to the internet. They also aren't checked in Docs Build.

Resolution

Use a relative link (for Docs pages in the same docset) or site-relative (for pages in different documents) instead.

For more information, see Add links to articles.

> **TIP**
>
> The Docs Authoring Pack for VS Code includes functionality to insert relative links, including links to articles in the same repo and site-relative links based on a Docs URL.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-alt-text

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Alt text '{text}' is duplicated. Within an article, alt text must be unique.
```

Users of screen readers rely on meaningful alternate text to understand the purpose of an image. If images with different file names have the same alt text, the alt text doesn't convey enough meaning to distinguish between the images. Duplicate alt text is also a problem for SEO: If multiple images have the same alt text, search engines don't know which image to rank higher in image search.

This rule has been enabled for PR only.

Resolution

Add unique, meaningful alt text to each image. For more information about how to write good alt text, see Accessibility guidelines. For more information about SEO for images, see Improve search rank with diagrams: Image alt text and filename.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-descriptions

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Attribute 'description' with value '{description-text}' is duplicated in '{file-path-1}', ...'{file-path-n}'.
Descriptions must be unique within a docset.
```

For SEO, descriptions must be unique within the current docset, and ideally should be unique across the site.

This rule has been enabled for PR only.

Resolution

Edit one or more `description` so all descriptions in your docset are unique.

For more information about how to write descriptions, SEO: How to write good meta descriptions.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-h1s

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
H1 '{h1-text}' is duplicated with other articles: '{file-path-1}', ...'{file-path-n}'. First level headings
must be unique within a docset.
```

For SEO, H1 headings (`#`) must be unique within a docset, and ideally should be unique across the site.

Resolution

Edit one or more first level heading so all H1s in your docset are unique.

For more information about how to write H1s, see SEO: Tips for writing a heading 1 (H1).

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-h2s

4/28/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Duplicate H2: '{heading text}'. Each H2 in an article must be unique.
```

Duplicated headings on a web page are an accessibility violation. Screen readers rely on headings to navigate the page for users with visual impairments, so duplication causes confusion. This validation checks for duplicate authored H2 headings within an article.

Resolution

Rename H2 headings within your article as necessary so they're all unique.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-pivot-groups

3/5/2021 • 2 minutes to read

Warning

```
Pivot group id '{id}' is duplicated in '{definition-file}'. Only the first group with this ID will be used.
Delete or rename the subsequent group(s).
```

If you define more than one pivot group with the same ID in a pivot definition file, only the first group will be used. Users attempting to reference subsequent groups with the duplicate ID will end up with broken or incorrect pivots.

Resolution

First, try to find all files in your docset that are using the duplicated pivot group ID, and determine which version they are meant to use. Then delete or rename the duplicate(s) and update files as appropriate.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-pivot-ids

3/5/2021 • 2 minutes to read

Warning

```
Pivot id '{pivot-id}' is duplicated in pivot group '{group-id}' in '{definition-file}'. Only the first pivot
with this ID will be used. Delete or rename the subsequent pivot.
```

The specified pivot group contains two or more pivots with the same ID. Only the first one will be used.

Resolution

Remove duplicate pivots or update their IDs.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-titles

4/28/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Attribute 'title' with value '{title-text}' is duplicated in '{file-path-1}', ...'{file-path-n}'. Titles must be unique within a docset.
```

For SEO, the value of the `title` attribute plus the `titleSuffix` attribute must be unique within the current docset, and ideally should be unique across the site.

Resolution

Edit one or more `title` so all titles in your docset are unique, and/or add `titleSuffix` metadata to add different branding to different folders within the docset.

For more information about how to write titles, SEO: Tips for writing titles.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-toc-entries

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
TOC entry '{file}' is duplicated within a TOC node.
```

Resolution

Only link to each file once in a TOC node.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

duplicate-uids

3/5/2021 • 2 minutes to read

Applies to: Learn

Error

`UID '{uid}' is duplicated in '{file-list}'.`

Learn UIDs must be globally unique.

Resolution

Verify that each learning path, module, and unit has a unique UID.

Learning path UIDs should follow this format:

```
<prefix>.<learning-path-folder-name>

Such as:
learn.architect-great-solutions-in-azure
```

For more information, see Create a learning path in the Learn Contributor Guide.

Module UIDs should follow this format:

```
<prefix>.<module-folder-name>

Such as:
learn.align-requirements-in-azure
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Unit UIDs should follow this format:

```
<prefix>.<module-folder-name>.<unit-name>

Such as:
learn.add-and-size-disks-in-azure-virtual-machines.introduction
```

Don't include file order prefixes in the unit name, such as "1-introduction". The unit *file name* will include a prefix to indicate its position in the module. The UID shouldn't in case the units get reordered in the future.

For more information, see Create a unit in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

file-not-found

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

`Invalid file link: '{file-path}'.`

The referenced file can't be found.

Resolution

Check the file name and path and update the link. For information about how to use links, see Links.

> **TIP**
>
> To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

> **IMPORTANT**
>
> Build validation checks relative links within a repo, but for reasons of scale doesn't currently check links to external sites. Also, links can break after content is published. To find all links currently broken in your content, see the broken link report for your repo on Docs Portal. For information about how to use the broken link report, see Broken link report in Docs Portal.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

forbidden-content

3/5/2021 • 2 minutes to read

Applies to: All content

Error

```
Link contains forbidden content. You can't link to air-gapped content. Don't fix this PR - close it
permanently and contact dacoulte@microsoft.com for further instructions.
```

Linking to air-gapped content is a high severity security issue. For more information, see Publishing for the air-gapped cloud (AGC).

Resolution

Close the PR permanently. Don't fix the issue in the same PR. Contact dacoulte@microsoft.com for further instructions to make sure the issue isn't reflected in GitHub commit history.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

forbidden-link

3/5/2021 • 2 minutes to read

Applies to: All content

Error

```
Link '{0}' is forbidden. You can't link to air-gapped content.
```

Linking to air-gapped content is a high severity security issue. For more information, see Publishing for the air-gapped cloud (AGC).

Resolution

Remove the link.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-empty

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`H1 is required. Add content to your top-level heading.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font.

To create an H1, begin a line with a single hash (#) followed by a space, then the heading text.

Resolution

To fix this issue, make sure your H1 includes content, not just a hash.

Bad:

```
---
author: meganbradley
ms.author: mbradley
---
#
This is not an H1
```

Good:

```
---
author: meganbradley
ms.author: mbradley
---
# This is an H1
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-in-include

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn

Suggestion

```
H1 headings aren't allowed in included files. Every parent file should have an H1.
```

Every article must have one and only one H1. The H1 must be in the parent file, not the included file.

Resolution

Remove the H1 from your included file and make sure its parent contains an H1.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-in-moniker

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`H1s are not allowed in moniker sections. Each article should have one and only one H1.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font. An H1 is created by beginning a line with a single hash (#) followed by a space, then the heading text. You can only have one H1 in each file. H1s aren't allowed in moniker sections. The reason is that they can easily cause duplicate or missing H1s depending on how versioning is configured.

You might also get this message if a moniker section contains a line of equals signs making a double underline, like this: `=======`. A double underline is an alternative Markdown syntax for an H1. It's also commonly seen in merge conflicts:

```
<<<<<<< HEAD
...
=======
...
>>>>>>> 1d82c7efe18f86136247fb366df5030843199c19
```

Resolution

To fix this issue, remove H1s from all moniker sections. Make sure the H1 at the top of the article is appropriate for all moniker sections.

If an H1 in a moniker section is actually a double underline (`=======`), remove it or replace it with a hashtag heading, such as `##`, as appropriate. If the double underline is part of a merge conflict, make sure to also remove the merge conflict beginning and ending markers and the obsolete text.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-missing

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`H1 is required. Use a single hash (#) followed by a space to create your top-level heading.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font.

To create an H1, begin a line with a single hash (#) followed by a space, then the heading text.

Resolution

To fix this issue, add an H1 as the first content after the YAML metadata block in your file:

```
---
author: meganbradley
ms.author: mbradley
---
# This is an H1
```

> **NOTE**
>
> This rule does not apply to included files. If you get H1 results on included files, you most likely need to move your included files into an `includes` folder. The `includes` folder can be at any level in the file path. Based on the path, Docs build will recognize the file as an included file and H1 validations won't run.
>
> A common cause of missing H1s in parent files is misuse of included files: the H1 is in the included file, not in the parent file. This isn't allowed, because using an H1 in an included file either means there are duplicate H1s in parent files or the included file is used only once. H1s should be unique within a content set and included files should only be used to share content among multiple files. If you get `h1-missing` results because the H1 is in an included file, the solution is to move the H1 - and all the included content if the included file is used only once - into the parent file. For more information about included files in Docs, see the Microsoft-internal article Include reusable content in articles.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-no-space

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`A space is required after the hash (#) in H1.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font.

To create an H1, begin a line with a single hash (#) followed by a space, then the heading text.

Without the space after the hash, Docs won't recognize an H1.

Resolution

To fix this issue, add a space after the hash in your H1.

```
---
author: meganbradley
ms.author: mbradley
---
# This is an H1
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

h1-not-first

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`Markdown content is not allowed before H1.`

Only the YAML metadata header can come before the H1 in a Markdown file. For example, you can't put a note before the H1:

```
---
# This is the YAML metadata header
author: meganbradley
---
> [!NOTE]
> You can't do this.

# This is the H1
```

Instead, put the note after the H1:

```
---
# This is the YAML metadata header
author: meganbradley
---
# This is the H1

> [!NOTE]
> This is OK.
```

Another common cause of this issue is byte order marks (BOMs) before the YAML header. BOMs are sometimes introduced by encoding issues when you commit content to a repo. They result in bad rendering in GitHub and should be removed.

Resolution

Remove any content other than the YAML metadata header from before the H1.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

hard-coded-locale

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

```
Link '{URL}' contains locale code '{code}'. For localizability, remove '{code}' from links to most Microsoft
sites.
```

Locale codes, such as `en-us`, shouldn't be included in links to certain Microsoft sites. If you include a locale code in a link in English content, it will also be included in localized links, which leads to a bad localized experience. For example, if a link in German localized content includes `en-us`, German customers will find themselves linking to the English article, even if a German version is available.

The following sites are in scope for this validation:

- azure.microsoft.com
- docs.microsoft.com
- msdn.microsoft.com (excluding social.msdn.com, which needs locale to ensure the correct forum is linked to)
- partner.microsoft.com
- technet.microsoft.com

Resolution

Remove locale codes from links to Microsoft sites, as follows:

Before:

```
https://docs.microsoft.com/en-us/vsts/load-test/app-service-web-app-performance-test
```

After:

```
https://docs.microsoft.com/vsts/load-test/app-service-web-app-performance-test
```

TIP

The Docs Markdown extension for VS Code includes a cleanup script for Microsoft links. The script checks all links to Microsoft sites in a repo to ensure that they begin with `https` instead of `http` and don't contain locale codes, such as `en-us`. To run the script:

1. Install the Docs Markdown extension for VS Code.
2. Click alt + M to open the Markdown menu.
3. Select **Cleanup**, then **Microsoft links**.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

heading-empty

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn units

Suggestion

`Empty {heading-level}. Headings must have text.`

Headings within an article are designated by hashes. Make sure each H1 (`##`) - H6 (`#######`) contains heading text.

Resolution

To fix this issue, make sure your heading includes text, not just hashes.

Bad:

```
---
author: meganbradley
ms.author: mbradley
---
##

This is an empty H2.
```

Good:

```
---
author: meganbradley
ms.author: mbradley
---
# This H2 has heading text
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

heading-no-content

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn units

Suggestion

`{heading} contains no content. Add content or remove the unused heading.`

Each heading in an article must contain content, including subheadings. For example, you can have an H2 immediately followed by an H3. You can't have an H2 followed by another H2 with no intervening content, an H3 followed by an H2 with no intervening content, or any heading without content at the end of an article.

Bad:

```
## H2

## Another H2 with no intervening content
```

```
### H3

## H2 with no intervening content
```

```
## H2
<file end>
```

OK:

```
## H2

### H3 with no intervening content
```

Resolution

Ensure that all your headings contain content. Included files, zones, and monikers might affect heading content and order.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

heading-skipped

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn units

Suggestion

```
Skipped heading. {0} is followed by {1}. For accessibility, headings must increment with no skipped levels.
```

Screen readers for users with visual impairments depend on headings being in order to properly convey the structure of a document.

Resolution

Review the current article and any included files and restructure headings so that the published article has no skipped levels. Note that zones and monikers might affect heading level.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

heading-with-underline

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Bad {heading} format. Use hashes to create H1 (#) or H2 (##)`

Markdown supports creating H1s and H2s with underlining, like this:

```
This is an H1
=============

This is an H2
-------------
```

These headings aren't allowed on Docs because they can be created accidentally and can't be validated for all potential issues.

Resolution

Replace headings created with underlines with standard hash headings:

```
# This is an H1

## This is an H2
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

html-link

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

This validation is not enabled on any production ruleset at this time, because HTML links aren't recommended but aren't prohibited. If you'd like to run this rule on your repo to find HTML links, contact mbradley.

Suggestion

`HTML link '{0}' is not allowed. Use a Markdown link.`

To ensure that all links in Docs content are accessible and secure, you must use valid Markdown links instead of HTML links.

Resolution

Use a valid Markdown link. The following are examples:

```
Relative link to another article on Docs:

[hard-coded-locale](hard-coded-locale.md)

Bookmark link to a heading in another article:

[hard-coded-local Resolution](hard-coded-locale.md#resolution)

URL to a website:

[microsoft.com](https://www.microsoft.com)
```

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

image-invalid

3/10/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Error

`Image reference '{image path}' is invalid per the schema. Unexpected attribute: '(attribute}'.`

The schema for the Docs `:::image:::` Markdown extension is as follows:

ELEMENT	DESCRIPTION	EXAMPLES
`:::image:::`	Identifies an image. Required.	`:::image type="content" source="../media/git-and-github/fork.png" alt-text="Illustrates the process flow for forking a repo." loc-scope="other":::`
image content	The content of the `:::image:::`, used for long descriptions of complex images. Required if `type="complex"`, otherwise not allowed. Should contain plain text only - no formatting.	`:::image type="complex" source="../media/git-and-github/fork.png" alt-text="Illustrates the process flow for forking a repo." loc-scope="other":::` `Long description of complex image goes here.` `:::image-end:::`
`:::image-end:::`	Identifies the end of an image. Required only if `type="complex"`.	`:::image ... :::` `...` `:::image-end:::`
`alt-text`	Provides alternative text for screen readers. Required if `type="content"` or `type="complex"`. Not allowed if `type="icon"`. Should contain plain text only - no formatting. For detailed alt text guidelines, see Accessibility guidelines for multimedia.	`alt-text="Illustrates the process flow for forking a repo."`
`border`	Specifies whether to automatically add a gray border to images. Default is `true`. To override the default, add `border="false"`.	`:::image type="icon" source="../media/git-and-github/fork.png" border="false":::`
`light-box`	Specifies an expandable image. Optional. The value is the path to the expanded image.	`:::imagetype="content"source="../media/inline-image.png"alt-text="howtoeditanarticleonDocs"lightbox="../media/e image.png":::`
`loc-scope`	Specifies the localization scope for the image, which may be different than for the parent article or module. The `loc-scope` attribute is optional per the image schema.	`loc-scope="vs-code"`
`source`	The path to the image. Required. The file name extension must be a supported image type: .png, .svg, .jpg, .jpeg, .gif	`../media/git-and-github/fork.png`

ELEMENT	DESCRIPTION	EXAMPLES
`type`	The type of image. Required. Valid values are `content` (default), `complex` (for images requiring a long description), and `icon` (for purely decorative images, such as product logos).	`type="content"`

Resolution

Review the schema requirements above and update your `:::image:::` to comply with them.

> **TIP**
>
> To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

include-not-found

3/5/2021 • 2 minutes to read

Error

```
Included file '{file-path}' can't be found in current repository.
```

Resolution

Verify that the included file exists in the repo and that the INCLUDE path is correct.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

insecure-link

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`Link '{URL}' is insecure. Links to Microsoft sites must use 'https' instead of 'http'.`

Links to Microsoft web sites, such as `microsoft.com`, must be secure.

This message means you either used an explicit Markdown link with `http` or you used a raw URL, such as `www.microsoft.com`. Raw URLs are converted to insecure links when published to Docs, and shouldn't be used. URLs that use `http` but aren't meant to be clickable should be styled as inline code rather than live links.

Resolution

Change clickable links to Microsoft sites to use `https` instead of `http`.

If your link is a raw URL, change it to an explicit Markdown link beginning with `https`:

```
[www.microsoft.com](https://www.microsoft.com)
```

For web addresses that aren't meant to be clickable, such as fully qualified domain names and namespace references, use inline code:

```
`www.microsoft.com:90`
```

> **TIP**
>
> The Docs Markdown extension for VS Code includes a cleanup script for Microsoft links. The script checks all links to Microsoft sites in a repo to ensure that they begin with `https` instead of `http` and don't contain locale codes, such as `en-us`. To run the script, first style any links that shouldn't be clickable as inline code, then follow these steps:
>
> 1. Install the Docs Markdown extension for VS Code.
> 2. Click alt + M to open the Markdown menu.
> 3. Select **Cleanup**, then **Microsoft links**.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

invalid-value

4/28/2021 • 2 minutes to read

Applies to: YAML files

Error

```
Invalid value for '{YAML field}': '{value}'.
```

The YAML field contains a value that's invalid per the relevant taxonomy.

Resolution

Choose a value from the appropriate taxonomy.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

languages-missing

4/28/2021 • 2 minutes to read

Applies to: Learning paths, Learn modules

Suggestion

`Missing required attribute: 'languages'. Add at least one language from the Dev Lang taxonomy.`

Learning paths and Learn modules must specify at least one programming language.

Resolution

Add one or more programming languages as appropriate from the Dev Lang taxonomy to a languages array in index.yml:

```
languages:
- al-language
- brainscript
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

link-out-of-scope

3/10/2021 • 2 minutes to read

Applies to: All content types

Warning

```
File '{file-name}' referenced by link '{link-value}' will not be built because it is not included in build scope.
```

The file has been excluded from the build scope via the `exclude` array in the docfx.json configuration file.

Resolution

Resource files such as included files and readmes can be excluded from the build in docfx.json. If you try to reference an excluded file by a file link or image reference, you'll get this Warning.

To fix the issue, first find the `content` group in docfx.json that includes the out-of-scope file. For example, the `articles` folder of azure-docs-pr:

```
"content": [
  {
    "src": "articles",
    "dest": ".",
    "files": [
      "**/*.md",
      "**/*.yml"
    ],
    "exclude": [
      "iot-edge/**",
      "**/_themes/**",
      "**/includes/**",
      "**/obj/**"
    ]
  }
```

Review the `exclude` array. You should find the path of out-of-scope file there. In this example, if you try to link to anything under `iot-edge/**` you'll get this Warning.

Then either update docfx.json so the file isn't excluded, or move the file to a path that isn't excluded. If you're not sure what to do, ask your repo owner for help.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

link-redirected

3/5/2021 • 2 minutes to read

Applies to: All content types

Suggestion

```
Redirected link '{0}' will be broken in isolated environments. Replace with a relative link or direct URL.
```

Link redirection services such as aka.ms and fwlink aren't recommended. These links will be broken in isolated (offline) environments.

Resolution

For links to other pages on docs.microsoft.com, use a direct relative link. These links will still work in isolated environments.

For links to other web sites, use a direct URL. Although these will still be broken in isolated environments, users will be able to see the URL and can view the content when outside the isolated environment.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

link-text-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Missing link text for link '{link path}'. Add appropriate text.`

Link text is required for links to be visible and clickable, and for screen readers to indicate the link target.

Resolution

Add descriptive link text, such as the title of the target page, to your link.

```
Bad:
[](../contribute-accessibility-multimedia.md)

Good:
[Accessibility guidelines for multimedia](../contribute-accessibility-multimedia.md)
```

TIP

To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

invalid-tab-group

11/2/2020 • 2 minutes to read

Warning

`Tab group with different tab id set.`

You can only have one tab group per article. Tab groups are defined implicitly based on the tabs used in your article. For example, suppose you define the following three tabs by using them in your article:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

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

PowerShell content...
```

This creates a tab group, which you must use throughout the article. That is, if you use the C# tab again, you must also use Python and PowerShell, and you can't add additional tabs unless you add them to the initial grouping. You also must maintain the tab order in subsequent usage.

Good:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

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

PowerShell content...

# [C#](#tab/csharp)

More C# content...

# [Python](#tab/python)

More Python content...

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

More PowerShell content...
```

Bad:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

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

PowerShell content...

# [C#](#tab/csharp)

More C# content, but no Python tab!

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

More PowerShell content...
```

Bad:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

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

PowerShell content...

# [C#](#tab/csharp)

More C# content...

# [Python](#tab/python)

More Python content...

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

More PowerShell content...

# [JavaScript](#tab/javascript)

JavaScript tab wasn't defined in initial tab group!
```

Bad:

```
# [C#](#tab/csharp)

C# content...

# [Python](#tab/python)

Python content...

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

PowerShell content...

# [C#](#tab/csharp)

More C# content...

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

PowerShell content out of order!

# [Python](#tab/python)

More Python content...
```

Resolution

Make sure all required tabs are defined in your initial tab group, and use the same tabs in the same order in all subsequent groups.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

manager-invalid

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

This is a custom validation that isn't implemented on most repos. If you'd like this rule run on your repo, contact mbradley.

Suggestion

`Invalid value for manager: '{value}' is not a valid Microsoft alias.`

Some content groups require the `manager` attribute to identify the author's manager.

Resolution

The value of `manager` must be a valid alias for an individual Microsoft employee. Verify the author's manager's alias and update the value.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

manager-missing

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

This is a custom validation that isn't implemented on most repos. If you'd like this rule run on your repo, contact mbradley.

Suggestion

`Missing attribute: manager. Add a valid Microsoft alias for the author's manager.`

Some content groups require the `manager` attribute to identify the author's manager.

Resolution

Add a valid Microsoft alias for `manager`:

```
---
ms.author: mbradley
manager: jemash
---
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

merge-conflict

6/15/2021 • 2 minutes to read

Applies to: All content types

Warning

`File contains merge conflict markers.`

Git merge conflict markers remain in your article, like this:

```
<<<<<<< HEAD
...
=======
...
>>>>>>> 1d82c7efe18f86136247fb366df5030843199c19
```

You get this message when you mark merge conflicts resolved for a pull request without deleting all the merge conflict markers. Merge conflict markers add meaningless content to the published page.

Resolution

Follow the steps in Resolve the merge conflict article to fix the conflict.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-achievement-not-badge

3/5/2021 • 2 minutes to read

Warning

```
Achievement UID '{uid}' doesn't refer to a badge. Modules must have badge achievements.
```

Resolution

Replace the specified achievement with a valid badge UID.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-achievement-not-found

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

```
Achievement UID '{uid}' is invalid. Specify a valid badge UID.
```

Resolution

Make sure the achievement specified for your module is a badge with a valid UID.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-badge-missing

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

`Missing attribute: badge. A valid badge is required for each module.`

Resolution

Add a valid badge UID, as follows:

```
badge:
  uid: <module-uid>.badge
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-child-not-found

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

`Child UID(s): '{uid-list}' can't be found.`

One or more of the UIDs specified in the `units` property of the module is invalid.

Resolution

Verify that all UIDs specified in the `units` property are valid unit UIDs. Here's an example:

```
units:
- learn.align-requirements-in-azure.introduction
- learn.align-requirements-in-azure.public-private-hybrid
- learn.align-requirements-in-azure.service-models
- learn.align-requirements-in-azure.iaas
- learn.align-requirements-in-azure.paas
- learn.align-requirements-in-azure.saas
- learn.align-requirements-in-azure.quiz
- learn.align-requirements-in-azure.summary
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-child-not-unit

3/5/2021 • 2 minutes to read

Error

`Invalid child UID(s): '{uid-list}'. A module can only have units as children.`

Resolution

Verify that all UIDs specified in the `units` property of the module are valid unit UIDs. Here's an example:

```
units:
- learn.align-requirements-in-azure.introduction
- learn.align-requirements-in-azure.public-private-hybrid
- learn.align-requirements-in-azure.service-models
- learn.align-requirements-in-azure.iaas
- learn.align-requirements-in-azure.paas
- learn.align-requirements-in-azure.saas
- learn.align-requirements-in-azure.quiz
- learn.align-requirements-in-azure.summary
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-uid-missing

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

`Missing attribute: uid. A valid Uid is required for each module.`

Resolution

Add a unique module UID in this format:

```
<prefix>.<module-folder-name>

Such as:
learn.align-requirements-in-azure
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

module-units-missing

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

`Missing attribute: units. At least one valid unit is required for each module.`

Resolution

Add one or more unit UIDs to the `units` property of the module. Here's an example:

```
units:
- learn.align-requirements-in-azure.introduction
- learn.align-requirements-in-azure.public-private-hybrid
- learn.align-requirements-in-azure.service-models
- learn.align-requirements-in-azure.iaas
- learn.align-requirements-in-azure.paas
- learn.align-requirements-in-azure.saas
- learn.align-requirements-in-azure.quiz
- learn.align-requirements-in-azure.summary
```

For more information, see Scaffold a module manually in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-author-missing

5/27/2021 • 2 minutes to read

This article applies to Conceptual (Markdown), .NET, REST, and PowerShell content.

Warning

`Missing attribute: ms.author. Add the current author's Microsoft alias.`

Resolution

Add the current author's Microsoft alias for the `ms.author` metadata attribute. The value should be the *current* owner of the article, not the original author if ownership has changed. We recommend that the author is a full-time employee or team distribution list (DL), rather than a short-term vendor.

If the alias is a DL, it must also be on the `ms.author` allowlist.

Valid values for `ms.author` DLs can be found on the ms.author allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-author-invalid

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), Learn, .NET, REST, PowerShell

Suggestion

```
Invalid value for ms.author: '{value}' is not a valid Microsoft alias, or is not an allowed distribution
list.
```

Resolution

Update the `ms.author` value with the current author's valid Microsoft alias. We recommend that the author is a full-time employee or team distribution list (DL), rather than a short-term vendor.

If the alias is a DL, it must also be on the `ms.author` allowlist.

See the ms.author allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-date-invalid

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

For Docs content:

`Invalid value for ms.date: '{value}'. Must be a date in format MM/DD/YYYY, no more than 30 days from today.`

For Learn content:

`Invalid value for ms.date: '{value}'. Must be a date in format MM/DD/YYYY, no later than today.`

Invalid `ms.date` values provide incorrect information about how up to date an article is, and can break some site and reporting functionality. For Docs content, dates can be up to 30 days in the future to allow predating for large releases such as Build and Ignite. This isn't currently allowed for Learn because a technical limitation related to the Learn Catalog API can cause partner tools to break if the `ms.date` is in the future.

Resolution

Confirm that the article is up to date with no broken content, then add a valid date in the format MM/DD/YYYY:

```
---
ms.date: 02/19/2019
---
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-date-missing

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

```
Missing attribute: ms.date. A freshness date is required for this content. Add a date in format MM/DD/YYYY.
```

Some content groups require `ms.date` to indicate "freshness" - that is, when the article was last reviewed for relevance, accuracy, correct screenshots, and working links. Freshness isn't the same as the last date the article was *published*, which will show on the page if `ms.date` isn't explicitly specified.

Resolution

Confirm that the article is up to date with no broken content, then add a valid date in the format MM/DD/YYYY:

```
---
ms.date: 02/19/2019
---
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-prod-and-service

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

```
Both ms.prod and ms.service can't be specified. Use ms.prod for on-premise products, or ms.service for cloud services.
```

Resolution

Either `ms.prod` or `ms.service` is required, and they can't both be present: `ms.prod` is used for on-premises products; `ms.service` is used for cloud services. Determine which is appropriate for your article, verify that the value is correct, and remove the other field.

For valid `ms.prod` and `ms.service` values, see Taxonomy lists. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-prod-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Missing attribute: ms.prod. If you specify ms.technology, you must also specify ms.prod.`

Use `ms.prod` to specify on-premises products. To specify more detailed information about `ms.prod`, you can optionally specify `ms.technology`, but if you specify `ms.technology`, you must also specify `ms.prod`. The values for `ms.prod` and `ms.technology` must be a valid pair.

Resolution

Confirm that the `ms.technology` value you've specified is correct for your article. Then add the appropriate `ms.prod` value that is a valid parent for the `ms.technology`.

For valid `ms.prod` and `ms.technology` pairs, see the ms.prod allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-prod-or-service-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

```
Missing attribute: either ms.prod or ms.service is required. Use ms.prod for on-premise products, or
ms.service for cloud services.
```

Resolution

Either `ms.prod` or `ms.service` is required, and they can't both be present: `ms.prod` is used for on-premises products; `ms.service` is used for cloud services. Determine which is appropriate for your article, verify that the value is correct, and remove the other field.

For valid `ms.prod` and `ms.service` values, see the ms.prod allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-prod-technology-invalid

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Invalid value for ms.prod: '{value}'.`

`Invalid value for ms.technology: '{value}' is not valid with ms.prod value '{value}'.`

Use `ms.prod` to specify on-premises products. To specify more detailed information about `ms.prod`, you can optionally specify `ms.technology`. The values for `ms.prod` and `ms.technology` must be a valid pair. Either your `ms.prod` value is invalid, or your `ms.technology` value isn't a valid pair with your `ms.prod`.

Resolution

Confirm that your `ms.prod` value is correct for your article. Then choose a valid `ms.technology` value.

For valid `ms.prod` and `ms.technology` pairs, see the ms.prod allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-service-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Missing attribute: ms.service. If you specify ms.subservice, you must also specify ms.service.`

Use `ms.service` to specify cloud services. To specify more detailed information about `ms.service`, you can optionally specify `ms.subservice`, but if you specify `ms.subservice`, you must also specify `ms.service`. The values for `ms.service` and `ms.subservice` must be a valid pair.

Resolution

Confirm that the `ms.subservice` value you've specified is correct for your article. Then add the appropriate `ms.service` value that is a valid parent for the `ms.subservice`.

For valid `ms.service` and `ms.subservice` pairs, see the ms.service allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-service-subservice-invalid

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Invalid value for ms.service: '{value}'.`

`Invalid value for ms.subservice: '{value}' is not valid with ms.service value '{value}'.`

Use `ms.service` to specify cloud services. To specify more detailed information about `ms.service`, you can optionally specify `ms.subservice`. The values for `ms.service` and `ms.subservice` must be a valid pair. Either your `ms.service` value is invalid, or your `ms.subservice` value isn't a valid pair with your `ms.service`.

Resolution

Confirm that your `ms.service` value is correct for your article. Then choose a valid `ms.subservice` value.

For valid `ms.service` and `ms.subservice` pairs, see the ms.service allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-topic-invalid

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

`Invalid value for ms.topic: '{value}'.`

Resolution

Add a valid value for `ms.topic` that indicates the article type.

For valid `ms.topic` values, see the ms.topic allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

ms-topic-missing

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

```
Missing attribute: ms.topic. Add a valid value to indicate the article type.
```

Resolution

Add a valid value for `ms.topic` that indicates the article type.

For valid `ms.topic` values, see the ms.topic allowlist. To request new values, see Metadata changes.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

multiple-h1s

3/5/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Multiple H1s are not allowed. You can only have one top-level heading.`

H1 refers to the first heading in a Markdown file. When published to docs.microsoft.com, the H1 shows at the top of the page in a large font. An H1 is created by beginning a line with a single hash (#) followed by a space, then the heading text. You can only have one H1 in each file.

You might also get this message if your article contains a line of equals signs making a double underline, like this: `=======`. A double underline is an alternative Markdown syntax for an H1. It's also commonly seen in merge conflicts:

```
<<<<<<< HEAD
...
=======
...
>>>>>>> 1d82c7efe18f86136247fb366df5030843199c19
```

Resolution

To fix this issue, change the heading level of later H1s to H2 (`##`), or otherwise reorganize your file.

```
---
author: meganbradley
ms.author: mbradley
---
# This is an H1

Some content...

## This is an H2
```

If an additional H1 is actually a double underline (`=======`), remove it or replace it with a hashtag heading, such as `##`, as appropriate. If the double underline is part of a merge conflict, make sure to also remove the merge conflict beginning and ending markers and the obsolete text.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

multiple-values

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown), .NET, REST, PowerShell

Warning

`Single-valued attribute '{attribute}' has multiple values. Remove additional values.`

You specified more than one value for an attribute that is only allowed one value.

`Single-valued attribute '{attribute}' is in multi-valued array format. Change to scalar format.`

Attributes that aren't allowed to have more than one value must be specified in the single-valued (scalar) YAML format.

Attributes in scope

The following attributes are required to be single-valued:

- `author`
- `ms.author`
- `ms.date`
- `ms.prod`
- `ms.service`
- `ms.subservice`
- `ms.technology`
- `ms.topic`
- `title`

Resolution

You get this Suggestion anytime a multi-valued array is used for a single-valued attribute, either with multiple values or a single value in the array.

YAML supports the following array formats for multi-valued attributes, such as `ms.custom`:

```
---
# comma-separated bracketed list:
ms.custom: [WIP, generated-via-CI]

# each value on its own line:
ms.custom:
  - WIP
  - generated-via-CI
---
```

These formats aren't valid for single-valued attributes, such as `author`.

If you specified multiple values, remove incorrect values and specify the single value on the same line as the attribute with no brackets:

```
---
author: meganbradley
---
```

If you have a single-valued array, change it to the scalar format above.

> **TIP**
>
> The Docs Markdown extension for VS Code includes a cleanup script for single-valued metadata in array format. The script checks all YAML metadata headers in the repo and converts badly formatted single-valued metadata to scalar format. To run the script, follow these steps:
>
> 1. Install the Docs Markdown extension for VS Code.
> 2. Click alt + M to open the Markdown menu.
> 3. Select **Cleanup**, then **Single-valued metadata**.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

notebook-dependent-schema-failed

6/16/2021 • 2 minutes to read

Applies to: Learn units

Error

```
DependentSchemas validation failure for attribute: 'notebook'. Make sure 'sandbox' is present and set to
'true'. The 'tasks' attribute must not be present when notebook is present.
```

The `notebook` attribute can only be specified for a Learn module unit if the `sandbox` attribute is also present and set to `true`. Only one of `notebook` or `tasks` can be specified for a unit.

Resolution

To specify a Jupyter notebook, make sure `sandbox` is set to `true` and `tasks` is not present in the unit YAML file.

```
### YamlMime:ModuleUnit
uid: learn-test-sandbox.test-orphan2-module.unit-1
title: Unit with Jupyter notebook
metadata:
  title: Unit with Jupyter notebook
  description: Specify a Jupyter notebook in a Learn modulde unit
durationInMinutes: 10
sandbox: true
notebook: http://raw.githubusercontent.com/test.ipybk
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-achievement-not-found

3/5/2021 • 2 minutes to read

Applies to: Learn modules

Error

```
Achievement UID '{uid}' is invalid. Specify a valid trophy UID.
```

Resolution

Make sure the achievement specified for your learning path is a trophy with a valid UID.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-achievement-not-trophy

3/5/2021 • 2 minutes to read

Error

```
Achievement UID '{uid}' doesn't refer to a trophy. Learning paths must have trophy achievements.
```

Resolution

Replace the specified achievement with a valid trophy UID.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-child-not-found

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

`Child UID(s): '{uid-list}' can't be found.`

One or more of the UIDs specified in the `modules` property of the learning path is invalid.

Resolution

Verify that all UIDs specified in the `modules` property are valid module UIDs. Here's an example:

```
modules:
- learn.pillars-of-a-great-azure-architecture
- learn.design-for-security
- learn.design-for-performance-and-scalability
- learn.design-for-efficiency-and-operations
- learn.design-for-availability-and-recoverability
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-child-not-module

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

`Invalid child UID(s): '{uid-list}'. A learning path can only have modules as children.`

Resolution

Verify that all UIDs specified in the `modules` property for the learning path are valid module UIDs. Here's an example:

```
modules:
- learn.pillars-of-a-great-azure-architecture
- learn.design-for-security
- learn.design-for-performance-and-scalability
- learn.design-for-efficiency-and-operations
- learn.design-for-availability-and-recoverability
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-duplication

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

This validation checks that published URL paths follow best practices for SEO. Because existing paths can't be easily changed and would result in many unactionable results in some build reports, this validation hasn't been enabled on a production ruleset for Docs at this time. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Article URL '{URL}' contains duplicated text '{segment-text}'. For SEO, article URLs should not contain duplication.
```

Duplicate path segments, such as `https://docs.microsoft.com/.../admin/get-help-with-domains/get-help-with-domains`, are bad for SEO and a bad user experience.

Resolution

Update your file paths so that segments aren't duplicated. If your content is already live, you'll have to redirect the old paths.

For more information, see File name and path guidelines.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-modules-missing

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

`Missing attribute: modules. At least one valid module is required for each learning path.`

Resolution

Add one or more module UID to the `modules` property of the learning path. Here's an example:

```
modules:
- learn.pillars-of-a-great-azure-architecture
- learn.design-for-security
- learn.design-for-performance-and-scalability
- learn.design-for-efficiency-and-operations
- learn.design-for-availability-and-recoverability
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-too-deep

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

```
Article URL '{URL}' is '{level}' levels deep. For SEO, URLs should be 3 levels or fewer.
```

Deep file paths are bad for SEO.

Resolution

Update your file paths so that they're no more than 3 levels deep.

For more information, see File name and path guidelines.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-trophy-missing

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

```
Missing attribute: trophy. A valid trophy is required for each learning path.
```

Resolution

Add a valid trophy UID, as follows:

```
trophy:
  uid: <path-uid>.trophy
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

path-uid-missing

3/5/2021 • 2 minutes to read

Applies to: Learning paths

Error

`Missing attribute: uid. A valid uid is required for each learning path.`

Resolution

Add a unique learning path UID in this format:

```
<prefix>.<learning-path-folder-name>

Such as:
learn.architect-great-solutions-in-azure
```

For more information, see Create a learning path in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-definition-conflict

3/5/2021 • 2 minutes to read

Warning

`Multiple source files are present for '{definition-file}'.`

This means that more than one zone pivot definition file is present in the repo that will be published to the same output path. This usually happens in versioned content when more than one `monikerRange` includes a zone pivot definition file.

Resolution

First, find the conflict. For example, suppose you have two versions of content, each stored in their own directory and associated with a different `monikerRange`. If each directory contains a definition file at the same location with the same name, it will be published to the same output path. Docs uses the output path to resolve zone pivots, so even if the configuration seems correct for each version, the result will be a conflict.

Once you've identified the conflict, you fix it in one of these ways:

- Use a single definition file for all versions. If the versions can all use the same pivot groups, you can use one file to apply to all versions. By default this should be zone-pivot-groups.yml at the docset root.
- Rename and/or move versioned definition files so each outputs to a unique path. Use `zone_pivot_group_filename` metadata to point to the correct output path for each version that doesn't use the default.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-definition-not-found

3/5/2021 • 2 minutes to read

Warning

```
No source file is present for '{definition-file}'. To use zone pivots, you must first define the zone pivot
groups in zone-pivot-groups.yml or specify a different zone pivot definition file in the
'zone_pivot_group_filename' metadata. Cross docset references to zone pivot groups definition files aren't
supported by Docs validation.
```

The system can't find the expected zone pivot definition source file. This is either zone-pivot-groups.yml at the docset root (the default) or a custom definition file specified in the `zone_pivot_group_filename` metadata.

Resolution

First you must determine what zone pivot definition file is expected for your article. This can be tricky because it depends on how your repo is configured. By default, the expected definition file is zone-pivot-groups.yml at the root of your docset. For example, in the azure-docs-pr repo, the `articles` folder is the root of the docset, meaning that everything under `articles` is published to `https://docs.microsoft.com/en-us/azure`. Therefore, by default the Docs system expects a source file called zone-pivot-groups.yml as a direct child of `articles` in the azure-docs-pr repo. In sql-docs-pr, the root is the `docs` folder, which publishes to `https://docs.microsoft.com/en-us/sql`. Therefore the default definition file for sql-docs-pr is zone-pivot-groups.yml as a direct child of `docs`. Each repo is different, so if you don't know which folder is your docset root you can ask for help as described in "Need help?" below.

Alternatively, you can specify a different definition file in the `zone_pivot_group_filename` metadata. You can save this definition file anywhere in the docset with any name, and reference it by its relative path to the root in the `zone_pivot_group_filename` metadata. For example, if you add a custom file called data-explorer-zone-pivot-groups.yml to the `articles/data-explorer` folder in azure-docs-pr, you'd add metadata as follows:

```
zone_pivot_group_filename: data-explorer/data-explorer-zone-pivot-groups.json
```

> **IMPORTANT**
> The extension is .json, not .yml, because you're technically referencing the published definition file, as opposed to the .yml source file.

To resolve this Warning, make sure the expected file is at the expected path, or update the `zone_pivot_group_filename` metadata to the correct path.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-group-missing

3/5/2021 • 2 minutes to read

Warning

```
Missing metadata attribute: 'zone_pivot_groups'. To use zone pivots in your file, you must specify the valid
pivot group that contains the pivot IDs you want to use.
```

The `zone_pivot_groups` metadata field specifies which zone pivot groups defined in a zone pivot definition file can be used in an article or included Markdown file. This metadata can be assigned:

- At the global or folder level in docfx.json.
- In the YAML header of a Docs Markdown article.
- In the `metadata` section of a Learn YAML file.

Each value of `zone_pivot_groups` must be a pivot group ID defined in the zone pivot definition file, such as zone-pivot-groups.yml.

Resolution

First, ensure that the pivot group you want to use is defined in zone-pivot-groups.yml (or in the custom zone pivot definition file if your docset doesn't use zone-pivot-groups.yml). Then add the pivot group ID as the value of `zone_pivot_groups`. If you're using a definition file other than zone-pivot-groups.yml, make sure the `zone_pivot_group_filename` metadata is also set to the path of the definition file.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-group-not-found

3/5/2021 • 2 minutes to read

Warning

```
Pivot group '{metadata-value}' isn't defined in '{definition-file}'. Make sure every pivot group you
reference in your content has been defined in your zone pivot definition file.
```

A pivot group ID specified in the `zone-pivot-groups` metadata isn't defined in the zone pivot definition file for your docset, usually zone-pivot-groups.yml.

Resolution

Check your zone pivot definition file. If your group is defined, update the `zone-pivot-groups` metadata with the correct pivot group ID. If the group isn't defined, add a new group as follows:

```yaml
- id: pivot-group-id
  title: Pivot group label
  prompt: Prompt text for the group
  pivots:
    - id: pivot-1
      title: Pivot 1 label
    - id: pivot-2
      title: Pivot 2 label
    - id: pivot-3
      title: Pivot 3 label
```

Then add the `pivot-group-id` value as the value of `zone-pivot-groups` for your article or unit.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-id-not-found

3/5/2021 • 2 minutes to read

Warning

```
Pivot ID '{pivot-id}' is not defined in '{group-list}' in '{definition-file}'. You can only use pivots that
have been defined in the referenced pivot group.
```

You've specified one or more pivot group in the `zone-pivot-groups` metadata, but you've used a pivot ID that isn't included in any of the groups.

Resolution

Either add the desired pivot ID to the appropriate group, or use a different pivot ID. If you add a pivot ID to a group, make sure to add that pivot ID to every other article or unit that uses the group.

For more information, see Zone pivots for Docs articles and/or Add zone pivots to a unit for Learn units.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

pivot-id-unused

3/5/2021 • 2 minutes to read

Warning

```
Pivot ID '{id}' is defined in the '{pivot-group}' group in '{definition-file}', but not used. You must use
all pivot IDs in a pivot group to avoid blank tabs.
```

Pivot groups define a standard set of tabs. If you use a group in a file but don't use all pivots in the group, content for the pivot will be missing.

Resolution

Add the missing pivot. If it's not appropriate to use all the pivots defined in the group, use a different group or don't use zone pivots in this file. In the case that the pivot group is appropriate but some content is supplemental for one or more pivots, you can add an empty zone. For example, suppose you have an article that provides variations of instructions for Windows versus Linux. You might have a section where the information is the same except you need to add an extra note for Linux. To indicate that the note is intentionally left off for Windows, you can add an empty zone for Windows with a comment, such as:

```
:::zone pivot="windows":::
<!-- No corresponding content for Windws. -->
:::zone-end:::
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

preserve-view-not-set

4/8/2021 • 2 minutes to read

Applies to: All content types

Suggestion

`Did you mean to link to a specific version? If so, add &preserve-view=true to the URL: {URL}`

To specify a specific version for a link, you must specify both the `view` parameter, which specifies the version, and `&preserve-view=true`, which tells the system to always preserve the `view` value. Without `preserve-view` set to `true`, Docs can't tell the difference between a hard-coded `view` and a `view` that was autogenerated based on the user's selections.

For example, a content developer might add a link to the URL `https://docs.microsoft.com/sql/t-sql/functions/cast-and-convert-transact-sql?view=sql-server-ver15`, assuming that the link will always go to the SQL Server 2019 (v15) version of the target page. However, a user might get to this same URL by selecting SQL Server 2019 from the site version picker, and Docs can't tell the difference. So if you don't also add `&preserve-view=true` to your link, Docs will determine the view based on version defaults and user selections, potentially overriding the specified version.

This rule has been enabled for PR only.

Resolution

To ensure that your specified view is always preserved, add `&preserve-view=true` to the URL, such as `https://docs.microsoft.com/sql/t-sql/functions/cast-and-convert-transact-sql?view=sql-server-ver15&preserve-view=true`.

If it's not essential to link to a specific version of the target page, remove the `view` parameter from your link, such as `https://docs.microsoft.com/sql/t-sql/functions/cast-and-convert-transact-sql`.

For more information about versioned links, see Links to specific versions.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

quiz-multiple-answers

3/5/2021 • 2 minutes to read

Error

`Question must have one correct answer.`

Each question in a quiz can have only one answer.

Resolution

Review the possible answers for the question and indicate which one is correct via the `isCorrect` YAML key. Only one can be correct. The following is an example:

```
quiz:
    title: Logging in to the Azure CLI 2.0
    questions:
    - content: "Were you able to log in to the Azure CLI 2.0?"
      choices:
      - content: "Yes"
        isCorrect: true
        explanation: "Great! Proceed to the next step in the tutorial."
      - content: "No"
        isCorrect: false
        explanation: "Try to log in to your Azure account at [https://portal.azure.com]. If you can't, try
resetting your password, then logging in again."
```

For more information, see Add a knowledge check in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

quiz-no-answer

3/5/2021 • 2 minutes to read

Warning

`Question must have one correct answer.`

Each question in a quiz must have one correct answer.

Resolution

Review the possible answers for the question and indicate which one is correct via the `isCorrect` YAML key. The following is an example:

```
quiz:
    title: Logging in to the Azure CLI 2.0
    questions:
    - content: "Were you able to log in to the Azure CLI 2.0?"
      choices:
      - content: "Yes"
        isCorrect: true
        explanation: "Great! Proceed to the next step in the tutorial."
      - content: "No"
        isCorrect: false
        explanation: "Try to log in to your Azure account at [https://portal.azure.com]. If you can't, try
resetting your password, then logging in again."
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

redirect-url-invalid

11/2/2020 • 2 minutes to read

Applies to: All content types

Warning

```
Can't preserve document ID for redirected file '{file-name}' because redirect_url '{URL}' is invalid or is is
in a different docset. Specify a redirect_url in the same docset, or set redirect_document_id to false in
.openpublishing.redirection.json.
```

This message means that either the `redirect_url` is an invalid link, or you're trying to redirect the document ID of the source file to a file in another docset. Document IDs can only be redirected within the current docset.

Resolution

First, determine which case is causing this Warning.

If your `redirect_url` is invalid, replace it with a valid relative path or absolute URL. If the `redirect_url` is a relative path to a file in the same docset, and no other redirection to that path specifies `redirect_document_id` as `true`, you can preserve the document ID.

If your `redirect_url` is a valid link to an article in a different docset, set `redirect_document_id` to `false`.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

redirected-file-not-removed

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

```
Redirected file '{file-name}' is still in the repo. After adding a file to the redirection JSON file, you
must delete the original file from the repo.
```

Resolution

Delete the redirected file. For more information about redirection, see How to redirect obsolete articles.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

rule-override-invalid

11/2/2020 • 2 minutes to read

Applies to: docfx.json

Warning

```
Rule '{error-code}' can't be overridden in docfx.json."
```

Some validation rules can be overridden for a file or set of files via the docfx.json config file, but most can't. This message means docfx.json contains override syntax for a rule that isn't overridable. The rule will still run on the excluded file(s).

Resolution

Remove the override syntax from docfx.json. If you believe you have a legitimate exception, file a bug via SiteHelp and the Docs team will evaluate your request.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

sandbox-not-mapped

3/5/2021 • 2 minutes to read

Applies to: Learn units

Error

```
Module '{parent-module}' is not mapped to the Azure sandbox. To use the sandbox, you must first submit an
onboarding request.
```

You've added `azureSandbox` metadata to a unit, but haven't registered its parent module with Azure Sandbox.

Resolution

To use the Azure Sandbox in an exercise unit, you must first register the parent module in the Sandbox database by filling out an onboarding form. Remove the `azureSandbox` metadata until mapping is confirmed. For more information, see Request an Azure Sandbox for a module in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

sensitive-language

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

```
Term '{term}' is sensitive and should not be used in content or code, or should be used with caution
depending on context.
```

Resolution

For a list of problematic terms and how to handle them, see the Terminology and inclusive language article.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

Tabbed conceptual

3/22/2021 • 2 minutes to read

Tab syntax

The syntax for tabs is as follows:

Single level tab:

```
# [Tab Display Name](#tab/tab-id)
```

Optional dependent tab:

```
# [Tab Display Name](#tab/tab-id/tab-condition)
```

Example of a single-level tab section with two tabs and the tab group terminator (---):

```
# [Linux](#tab/linux)

Content for Linux...

# [Windows](#tab/windows)

Content for Windows...

---
```

Tabs can optionally contain secondary tabs, or dependency tabs. This makes tabs dependent on the selection in another set of tabs. Here's an example:

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

Azure CLI content for Linux...

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

Azure CLI content for Windows...

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

PowerShell content for Linux...

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

PowerShell content for Windows...

---
```

The following validations apply to tab syntax:

- Tab syntax must be correct.
- Dependent tabs must have been defined in a previous tab group.
- Only one level of dependency is allowed.
- No fewer than two tabs are allowed.
- No more than four tabs are allowed.

- Tabs must be approved.
- Tab/ID pairs must be valid.
- Cannot have the same tab ID multiple times in one tab group.

Approved tabs

The following tab name/tab ID pairs are approved. Tabs names and IDs aren't validated, so you can currently use any tabs you want.

TAB NAME	TAB ID
`[.NET]`	`(#tab/dotnet)`
`[.NET Core 1.x]`	`(#tab/netcore1x)`
`[.NET Core 2.x]`	`(#tab/netcore2x)`
`[.NET Core 2.0]`	`(#tab/netcore20)`
`[.NET Core 2.1]`	`(#tab/netcore21)`
`[.NET Core 2.2]`	`(#tab/netcore22)`
`[.NET Core 3.x]`	`(#tab/netcore3x)`
`[.NET Core 3.0]`	`(#tab/netcore30)`
`[.NET Core CLI]`	`(#tab/netcore-cli)`
`[Azure CLI]`	`(#tab/azure-cli)`
`[Android]`	`(#tab/android)`
`[Ansible]`	`(#tab/ansible)`
`[Browser]`	`(#tab/browser)`
`[C#]`	`(#tab/csharp)`
`[C# Script]`	`(#tab/csharp-script)`
`[CentOS]`	`(#tab/centos)`
`[Command Line]`	`(#tab/command-line)`
`[Debian]`	`(#tab/debian)`
`[Docker Hub]`	`(#tab/docker-hub)`
`[F#]`	`(#tab/fsharp)`

TAB NAME	TAB ID
`[Fedora]`	`(#tab/fedora)`
`[iOS]`	`(#tab/ios)`
`[Java]`	`(#tab/java)`
`[JavaScript]`	`(#tab/javascript)`
`[Linux]`	`(#tab/linux)`
`[macOS]`	`(#tab/macos)`
`[Managed Kubernetes]`	`(#tab/kubernetes-managed)`
`[Maven]`	`(#tab/maven)`
`[Mint]`	`(#tab/mint)`
`[Node.js]`	`(#tab/nodejs)`
`[npm]`	`(#tab/npm)`
`[NuGet]`	`(#tab/nuget)`
`[openSUSE]`	`(#tab/opensuse)`
`[Other]`	`(#tab/other)`
`[Oracle Linux]`	`(#tab/oracle-linux)`
`[Package Manager]`	`(#tab/package-manager)`
`[PEAR]`	`(#tab/pear)`
`[pip]`	`(#tab/pip)`
`[Portal]`	`(#tab/azure-portal)`
`[PowerShell]`	`(#tab/azure-powershell)`
`[Private Registry]`	`(#tab/private-registry)`
`[Python]`	`(#tab/python)`
`[Resource Manager Template]`	`(#tab/azure-resource-manager)`
`[RHEL]`	`(#tab/rhel)`

TAB NAME	TAB ID
`[RubyGems]`	`(#tab/rubygems)`
`[SQL Server]`	`(#tab/sql-server)`
`[SQLite]`	`(#tab/sqlite)`
`[Terraform]`	`(#tab/terraform)`
`[TypeScript]`	`(#tab/typescript)`
`[Visual Basic]`	`(#tab/vb)`
`[Visual Studio]`	`(#tab/visual-studio)`
`[Visual Studio 15.6 and earlier]`	`(#tab/vs156)`
`[Visual Studio 15.7 and later]`	`(#tab/vs157)`
`[Visual Studio Code]`	`(#tab/visual-studio-code)`
`[Visual Studio for Mac]`	`(#tab/visual-studio-mac)`
`[Ubuntu]`	`(#tab/ubuntu)`
`[Unmanaged Kubernetes]`	`(#tab/kubernetes-unmanaged)`
`[Windows]`	`(#tab/windows)`

Example articles

Here are examples of articles with tabbed conceptual formatting:

- Handle SameSite cookie changes in Chrome browser
- Logging in MSAL applications
- Tutorial: Use dynamic configuration in an ASP.NET Core app

table-syntax-invalid

5/13/2021 • 2 minutes to read

Applies to: Conceptual (Markdown)

Suggestion

`Table syntax is invalid. Ensure your table includes a header and is surrounded by empty lines.`

Tables must have headers and be surrounded by blank lines to render correctly.

```
This table isn't valid because it has no header:

|This | table has |no header|at all |

This table isn't valid because it isn't surrounded by blank lines:
|Tables |need |space... |
|-------|-----|---------|
... to breathe!
```

Resolution

Make sure your table has a header and is surrounded by blank lines:

```
|This is    |a simple   |table header|
|----------|-----------|------------|
|table     |data       |here        |
|it doesn't|actually   |have to line up nicely!|
```

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

task-type-invalid

3/5/2021 • 2 minutes to read

Error

```
Task has invalid Azure resource type: '{resource-type}'.
```

Resolution

Verify that the specified `resourceGroup` matches a resource group that's available for your lab.

For more information, see Add task validation to an Azure Sandbox lab in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

title-missing

11/2/2020 • 2 minutes to read

Applies to: Conceptual (Markdown)

Warning

```
Missing attribute: title. Add a title string (19 - 59 characters) to show in search engine results.
```

Resolution

Add a title string to show in search results. In general, titles should be between 19 and 59 characters, should be distinct from the H1 of the article, and should include relevant branding words. Don't include " | Microsoft Docs" in your title - it's added automatically by Docs, and is ignored if you add it explicitly. If you want to add additional branding, such as " - Azure", to all titles in a content set, set the `titleSuffix` metadata value globally or for a folder.

You can preview how your title will look in Google on https://moz.com/learn/seo/title-tag.

For more information about good titles and search engine optimization (SEO), see SEO Basics.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

title-suffix-missing

11/2/2020 • 2 minutes to read

This validation checks for the `titleSuffix` metadata, which appends branding information to Docs titles. Applying `titleSuffix` globally or at the folder level is recommended, but because manually added branding is still valid, this validation hasn't been enabled on a production ruleset for Docs at this time. If you'd like this rule run on your repo, contact mbradley.

Suggestion

```
Missing attribute: titleSuffix. Add a titleSuffix value to append branding to your title. Using titleSuffix
is the recommended way to append a branding string to titles for SEO.
```

The `titleSuffix` metadata field applies a branding string, such as " - Xamarin", to all titles in scope.

Resolution

Apply `titleSuffix` globally or at the folder level via docfx.json. For example, the following JSON will append the string `- Xamarin` to all title values in the docset:

```
"globalMetadata": {
  "titleSuffix": "Xamarin",
   ...
}
```

Using `titleSuffix`, you can apply consistent branding to all articles in a docset or folder, without having to manually add it to every `title` value.

For more information, see SEO: Tips for writing titles.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

toc-breadcrumb-link-external

11/2/2020 • 2 minutes to read

Suggestion

```
TOC or breadcrumb link '{link}' goes to an external web site. TOCs and breadcrumbs should only link to pages
on docs.microsoft.com.
```

Resolution

Remove links to external sites from TOC and breadcrumb files.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

toc-missing

11/2/2020 • 2 minutes to read

Suggestion

```
Article '{file-name}' is not present in a toc.yml file. Every article must be in a TOC.
```

Resolution

Make sure the article or an article above it in its hierarchy is referenced in toc.yml file.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

token-not-found

3/5/2021 • 2 minutes to read

Warning

```
Token '{0}' can't be found in current repository.
```

Resolution

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-duration-missing

3/5/2021 • 2 minutes to read

Error

```
Missing attribute: DurationInMinutes. DurationInMinutes is required for each unit and must be greater than 0.
```

Resolution

Add the estimated number of minutes it will take a user to finish this unit.

For more information, see Create a unit in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-multiple-parents

3/5/2021 • 2 minutes to read

Error

```
Unit '{uid}' can't have two or more parents ('{parent-list}').
```

Resolution

Remove the unit UID from all but one parent module.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-no-module-parent

3/5/2021 • 2 minutes to read

Error

```
Unit '{uid}' must belong to a valid module.
```

This means either the unit doesn't have a parent or its parent isn't a module.

Resolution

Verify that the unit is the child of a valid module.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-task-and-quiz

3/5/2021 • 2 minutes to read

Error

```
A unit can't have both a quiz and a task.
```

Resolution

Remove either the quiz or the task.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

unit-uid-missing

3/5/2021 • 2 minutes to read

Error

```
Missing attribute: uid. A valid uid is required for each unit.
```

Resolution

Add a unique unit UID in this format:

```
<prefix>.<module-folder-name>.<unit-name>

Such as:
learn.add-and-size-disks-in-azure-virtual-machines.introduction
```

Don't include file order prefixes in the unit name, such as "1-introduction". The unit *file name* will include a prefix to indicate its position in the module. The UID shouldn't in case the units get reordered in the future.

For more information, see Create a unit in the Learn Contributor Guide.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

validation-timeout

11/2/2020 • 2 minutes to read

Applies to: All content types

Warning

```
The call to the validation service timed out and validation was not completed. This happens when there's an
issue with the service and continuing to retry the call could cause build delays. You might have content
issues that were not reported. To retry validation, close and re-open your PR, or force a full build of your
branch via https://ops.microsoft.com. Note that forcing a full build requires admin permissions to the repo.
If you don’t know who your repo admin is, or if you continue to see this message after a forced build, file
an issue via https://SiteHelp.
```

You get this Warning when there are transient issues in the validation service. For example, a server might be in a bad state. After several tries, the call times out and validation is canceled to avoid build delays and clogging the build pipeline.

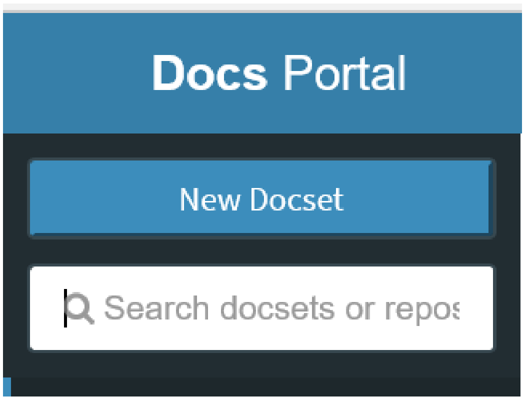
Resolution

Try closing and reopening your PR, or forcing a full build via Docs Portal. Often service issues clear themselves up after the initial retry.

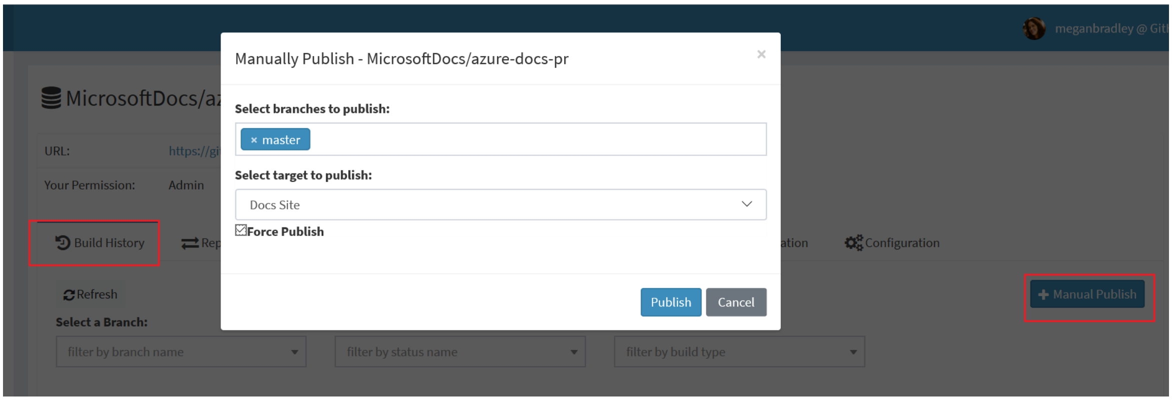
You must be a repo admin to force a build via Docs Portal. If you don't know who your repo admin is, or if you continue to get this message after a forced build, file an issue via https://SiteHelp.

If you're a repo admin, you can force a full build as follows:

1. Go to Docs Portal and sign in.
2. Find your repo by searching in the upper left corner, and select it.



3. On the **Build History** tab, click **+ Manual Publish**.
4. Select the branch that's getting the Warning, such as master.
5. For target, keep the default, **Docs Site**.
6. Check the **Force Publish** checkbox.
7. Click **Publish**.



Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

xref-not-found

3/5/2021 • 2 minutes to read

Applies to: All content types

Warning

`Cross reference not found: '{uid}'.`

The referenced UID can't be found.

Resolution

Verify the UID and that XRef links are enabled for your docset. For more information, see XRef (cross reference) links and Xref Service.

> **TIP**
>
> To easily author Docs Markdown, install the Docs Authoring Pack for VS Code. This set of extensions includes link and asset insertion with relative paths, text styles, Docs preview, Markdown linting, and a spell checker.

Need help?

If you need help with this problem, post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp!

Error documentation not available

11/2/2020 • 2 minutes to read

No documentation is currently available for this validation issue. For help with the issue and to request documentation, please post on the Docs Support Teams channel or file an issue via https://aka.ms/SiteHelp.

Docs Local Validation

4/15/2021 • 2 minutes to read

The Docs Validation extension for Visual Studio Code allows you to perform build validations against your local Docs conceptual or Learn repository. This ensures your content is free of validation issues before making a pull request.

The Docs Validation extension is included in the Docs Authoring Pack.

Install Options

- Install as standalone
- Install as part of the Docs Authoring Pack

Prerequisites

- Install git.
- Clone your Docs or Learn repo locally in VS Code.
- All files in the repo must be saved.

Perform Local Validation

1. Clone your Docs or Learn repository locally.
2. Open the local repository in VS Code with the extension installed.
3. Make content changes, and **save** them locally.

> **NOTE**
> You do **NOT** have to commit them to your local repository yet.

4. If you're a Microsoft employee, optionally sign in by clicking **Docs Validation** from the VS Code status bar. You can still use the extension without signing in, but sign in is recommended to use the most recent build information for private repos.
5. Right-click anywhere on your folder/file structure and click on **Validate this repository**.

6. The process will start, and you can monitor the status in the output window

7. Once the process is complete, you can review any Validation issues identified in the "Problems" pane

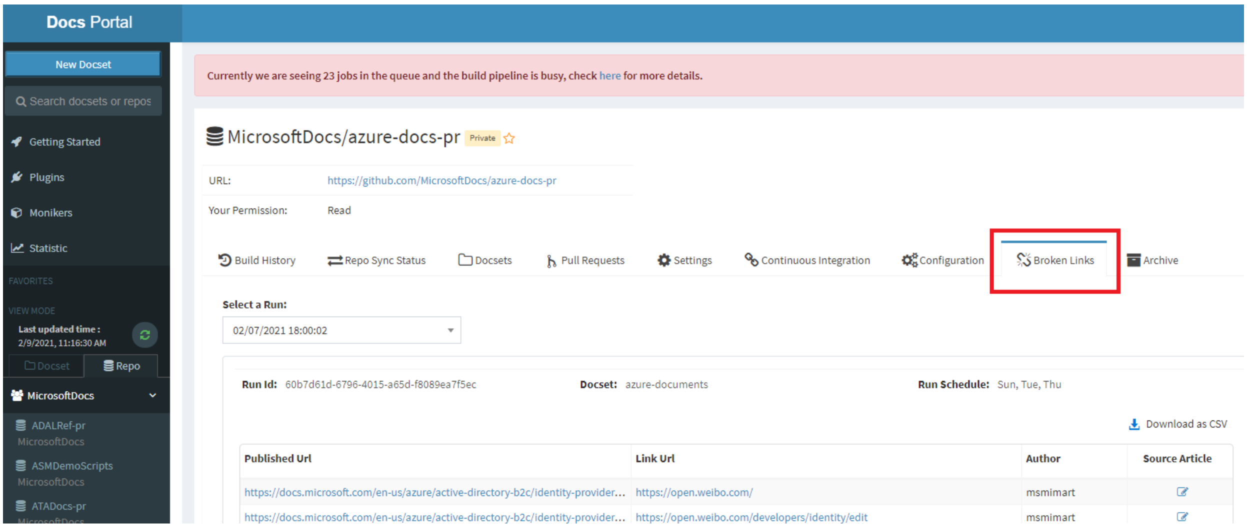
Details

From within VS Code a content creator can now perform the exact same Docs Build Validations against your locally cloned or forked repository **PRIOR** to submitting it into the hosted repository and waiting the Docs Build pipeline to evaluate your content.

Broken link report in Docs Portal

11/2/2020 • 5 minutes to read

A new broken link report is enabled on all repos that are live on Docs. The report is available through the Docs Portal. Any user with Read permissions for a repo can view the broken link report for that repo. Just find the repo in the portal and click the **Broken links** tab on the far right:



If the **Broken links** tab isn't available for your repo, it means the repo hasn't been published live since the broken link report was enabled. Publish the repo live to see broken link results and begin weekly runs.

When you click the **Broken links** tab, you'll see the results of the latest run. You can select a different run of the most recent three from the **Select a Run** drop-down box. By default, the report runs once per week. It can be run more frequently if needed; contact mbradley to request more frequent runs.

On the web view of the report, you'll see the **Published Url** of the article that contains the broken link, the **Link Url** of the broken link itself, the **Author** of the article by GitHub ID, and a **Source Article** edit button to open the article in GitHub or DevOps. Click **Download as CSV** at the top right corner of the results to download a sortable, filterable CSV with the following additional columns:

- **ms.author**: The author's Microsoft alias.
- **ms.service**: The ms.service value, if present.
- **ms.subservice**: The ms.subservice value, if present.
- **ms.prod**: The ms.prod value, if present.
- **ms.technology**: The ms.technology value, if present.
- **ms.topic**: The ms.topic value.

What types of issues are checked for?

The broken link report returns results for links that return 404 or otherwise unavailable when pinged by the broken link service. This includes links in conceptual Docs articles, reference articles, Learn modules, TOCs, and breadcrumbs. It includes links to web sites external to docs.microsoft.com, links to other docs.microsoft.com pages, images, videos, and samples. It doesn't include links to non-existent bookmarks on valid pages, because these links fall back to the beginning of the article and therefore return success.

The report also returns results for security issues.

The following are result categories:

CATEGORY	DESCRIPTION	USER ACTION
Broken	The link returns 404 or another HTTP status code that indicates that the link is permanently unavailable.	Replace the link with a valid one, or remove it altogether.
Security issue	The link has a security issue, such as an invalid certificate, and might cause a vulnerability.	We strongly recommend that you remove the link. If the link is essential, contact the site owner to fix the security issue.

Is the broken link report integrated into Docs Build?

Although the broken link results show in Docs Portal, the report isn't currently integrated into Docs Build. Some types of links - specifically Markdown links to other articles on Docs, image links, code links, and bookmarks - are checked at build time. However, many of these links can become broken after the fact if assets are moved or removed, so build checking isn't sufficient to ensure no 404s on the live site. Links to sites external to Docs aren't checked at build time because it would degrade performance. Therefore the broken link report is separate from Docs Build, although some integration is planned for the future.

What about CATS?

The Docs Portal broken link report replaces CATS, which was costly to maintain and didn't scale well to support all of Docs. The main differences between CATS and the new report are as follows:

	CATS	DOCS PORTAL
Scope	Ran only on demand or as manually scheduled via CATS.	Runs on all Docs repos that have been published live at least once since the report was enabled.
Cadence	Ran on demand or as scheduled.	Runs weekly by default. Can be run more frequently if needed.
Docs-to-Docs links	Checked.	Checked.
External site links	Checked.	Checked.
Bookmarks	Checked.	Not checked, because broken bookmarks fall back to the top of the target page and don't 404. Bookmarks are now checked as part of build validation. For more information, see bookmark-not-found.
TOC links	Checked as a separate content validation, "Broken TOC links".	Included in the broken link report.
Breadcrumb links	Checked as a separate content validation, "Broken breadcrumb links".	Included in the broken link report.
Images, videos, code samples	Not checked.	Checked, because missing assets return 404.

	CATS	DOCS PORTAL
Link allowlist	For URLs that weren't meant to be clickable links, such as domain names, namespace references, and example links like `www.contoso.com`, CATS maintained an allowlist to exclude results. This solution wasn't ideal, because the list was quite broad and sometimes cause real broken links to be missed. Links that weren't meant to be clickable also rendered as real links, which did in fact return 404 if clicked.	In the new report, we've eliminated the allowlist. To avoid broken link reports, links that aren't meant to be clickable should be styled as inline code (`like this`).

What do I do to exclude non-clickable links from results?

If your repo used the CATS link allowlist to exclude URLs from the report, you'll find results in the new report that seem to be false positives. The solution is to style URLs that aren't meant to be clickable links, or that only 404 under certain circumstances, as inline code (`like this`). In addition to excluding these URLs from the report, this will render them as non-clickable on the Docs site, so users won't be confused by the appearance of clickable links that actually return 404.

The following is an example:

```
`https://gallery.azure.com`
```

It renders like this on the Docs site:

`https://gallery.azure.com`

Because the old allowlist was quite broad, you might want to review these results to make sure they're true exceptions before you convert them to inline code. If you used the allowlist heavily and want a more efficient solution than converting each one manually, a script is available to find URLs in a repo that match the patterns specified in the allowlist and wrap them with inline code. This script will be run on the core Docs and Learn repos supported by the Developer Relations PR Review process/PRMerger as part of migration to the new report. If you want help scrubbing a repo that isn't supported by the PR Review process, send email to mbradley.

CATS deprecation

11/2/2020 • 6 minutes to read

The CATS site QA tool was developed over several years to test published Docs content for missing metadata, broken links, content rendering problems, and so on. Because CATS was supported by a large vendor team, it was often easier to get work done on CATS than within the Docs pipeline. Many content tests were implemented as site QA tests when implementation as build validation would have been more enforceable and more consistent with other validation experiences.

Now that Docs is a mature product, we've deprecated CATS and replaced it with functionality within the core Docs authoring platform. The main replacement features are new build-time validations, a new broken link report, and new or updated tools to provide author-time feedback on issues that aren't clear cut enough for build validation, such as some SEO guidelines.

CATS is fully deprecated as of June 30, 2020.

Final CATS deprecation: June 30, 2020

All priority CATS checks have now been enabled as build validations or alternative tooling, or will be turned on in Docs Build early in FY21. Some lower priority CATS tests are considered obsolete and will not be replaced.

The following CATS replacement validations were enabled on June 29, 2020:

- alt-text-missing
- column-header-missing
- duplicate-h1s
- duplicate-titles

The following CATS replacement validation is available for ad hoc runs, but has not been enabled in build because it isn't currently required by the Docs platform:

- alt-text-bad-length

The following CATS replacement validations are currently in testing/rollout and are expected to be enabled in build or made available as ad hoc validations at the end of July:

- alt-text-bad-value
- description-bad-length (ad hoc only - use Search Results Preview in the Docs Authoring Pack for VS Code to verify your titles and descriptions at author time)
- description-missing
- dev-language-missing
- duplicate-alt-text
- duplicate-descriptions
- duplicate-toc-entries
- path-duplication
- path-too-deep
- title-suffix-missing
- toc-breadcrumb-link-external
- toc-missing

To request an ad hoc run on your repo of any validations not currently enabled, or if you have questions about replacements for specific CATS tests, please contact mbradley.

Deprecated as of April 17, 2020: CATS broken link checks

CATS provided broken link checking on demand or on a schedule via the **Docs Content Validation** tab:

- **Broken internal links**
- **Broken bookmarks**
- **Broken TOC links**
- **Broken breadcrumb links**
- **Broken external links**
- **Broken video links**
- **Broken image links**
- **Broken sample links**

These tests have been replaced with a new broken link report that runs on all live Docs repos and can be accessed via Docs Portal. For more information about the new report, including how it differs from CATS, see Broken link report.

Deprecated as of November 20, 2019: Alt text length, insecure links, H1 headings

Alt text length test

The P2 **Alt text length test** under SEO rules gives outdated guidance, telling users to keep alt text between 30 and 70 characters, including spaces. The current maximum for alt text is 250 characters, with no specific minimum. For more information about alt text and other accessibility requirements for images, see Accessibility Guidelines for Multimedia.

Docs Build validation for alt text issues is coming early in 2020. Because the CATS test is P2 with incorrect results, we're removing it from CATS even though build validation isn't ready yet to avoid confusion and unnecessary work for authors.

Test for insecure links

The insecure-link validation has been added to the Docs Platform ruleset, which is enabled on all conceptual repos by default. This validation replaces the CATS **HTTPS Checking test**, which checks that links to Microsoft sites use http instead of https. The redundant CATS test is therefore deprecated.

H1 heading tests

The following validations have been added to the Docs Platform ruleset to replace the CATS **Missing or Extra H1 heading** test:

- h1-empty - checks for an H1 marker (#) with no text following it.
- h1-missing - checks for no H1 in an article.
- h1-no-space - checks for an H1 with no space after the hash (#), which Docs Build interprets as plain text.
- multiple-h1s - checks for more than one H1 in an article.

The redundant CATS test is now deprecated.

Docs Archived Content Validation

The Docs Archived Content Validation pane offered a set of tests specifically for migrating content from MSDN/TechNet to the Docs previous versions archive. These migrations are now complete, and the tests are no longer needed.

Deprecated as of October 11, 2019: Metadata tests and Side-by-side comparison tests

Required metadata populated by author

The following metadata must be present and valid: `author`, `ms.author`, `ms.date`, `ms.topic`, `title`, and `ms.service` OR `ms.prod`. `ms.subservice` is optional, but must be a valid pair with `ms.service` if specified. `ms.technology` is optional, but must be a valid pair with `ms.prod` if specified.

For **conceptual** content, these tests are already available as build validations and enabled for most repos that rely on these values for content reporting. For more information, see the Developer Relations/CGA reporting ruleset.

If you currently use CATS for metadata validation and do *not* yet have the full validation ruleset, file a request using this template.

For **reference** content, build validation for metadata is not yet available (ETA is FY19 Q3). However, there are other ways to verify your reference metadata. Author-added reference metadata is set in the docfx.json file, as described in this article. It's recommended that you set default values at the highest level possible, and that you use team aliases instead of individual aliases, to minimize churn in your values. Reference docsets are supported by the Metadata Health report, which will remain available until metadata build validation is enabled for reference. If you can't access the report or if your reference repo isn't shown, contact Khairun Jamal.

Missing or extra title

The `title` metadata field is also validated for conceptual content. It is required and can only be specified once. For reference content, `title` is autogenerated.

Required metadata populated by OPS

These tests were added in the early days of Docs when Docs component testing didn't cover system-generated metadata. The tests only fail if something goes wrong in the system, and there's nothing authors can do to fix failures. New component tests have been added to catch these issues so author-facing tests are no longer necessary in CATS.

APIScan metadata validation

APIScan metadata is automatically generated for .NET, UWP, and Java reference. This metadata is now covered by end-to-end component testing, so author-facing tests are no longer necessary in CATS.

Docs Side-by-side Comparison and MSDN/TechNet Migration Validation

These side-by-side comparison tests were created for use in Docs content migrations, but are no longer used by the migration team and will be removed from CATS. Future migration efforts will implement testing as part of the migration work.

FAQ

Will CATS tests go away without warning?

No. All deprecations will be announced two weeks before functionality is removed from CATS. To ensure you receive notifications, join the Docs Authoring Partner Announcements alias.

I depend on CATS test X. How do I know what you're planning to replace it with?

An overall CATS deprecation plan is posted here. You can review it for information about how we plan to replace various features of CATS tests. The plan is not finalized and some replacements may change before implementation. Check the linked features for current status of replacement functionality.

Is CATS supported during deprecation?

Yes, but at a lower priority than in the past. No new features will be accepted, and bugs will be prioritized based on how critical the functionality is. As features are replaced, they'll be removed from CATS.

DocsImageSorter

3/9/2021 • 2 minutes to read

The DocsImageSorter is a prototype tool that uses Azure Cognitive Services to identify issues with images. The prototype is a command line tool that you can download and run on your repo. Currently the tool distinguishes between complex and standard images, and raises a failure if a complex image lacks a long description. Additional functionality, including identifying images with bad contrast and identifying images that contain sensitive terminology, are planned for the future.

To use the tool, first download it from one of the following locations and extract the .zip file:

- Windows: https://authoringresources.blob.core.windows.net/image-sorter/DocsImageSorter-Win.zip
- Mac: https://authoringresources.blob.core.windows.net/image-sorter/DocsImageSorter-Mac.zip

Then, in a command prompt:

1. Navigate to the folder where you extracted the .zip.
2. Type the following command: `DocsImageSorter.exe -p "<repo-path>" -o "<csv-output-path>"`, such as:

```
DocsImageSorter.exe -p "C:\Users\mbradley\docs-help-pr" -o "C:\Users\mbradley\Desktop\image-output.csv"
```

3. Open the CSV output file. Review the columns:

COLUMN	DESCRIPTION
SOURCE	The path to the article that references the image.
IMAGE	The image file name.
MS.AUTHOR	The `ms.author` value for the source article.
REFERENCE TYPE	The type of Markdown image reference, as follows: - standard: A standard Markdown image in the format `![<alt text>](<folderPath>)`. - complex: The Docs `:::image:::` extension with `type="complex"`. - content: The Docs `:::image:::` extension with `type="content"`. - icon: The Docs `:::image:::` extension with `type="icon"`.
ACTUAL TYPE	The actual type of the image, based on the machine learning model. Currently the model only distinguishes between `complex` (meaning an image is complex and requires a long description in addition to alt text for accessibility) and `content` (all other images, including icons). `Undetermined` means the model was unable to confidently categorize the image.

COLUMN	DESCRIPTION
PASS/FAIL	Whether the image meets relevant guidelines. Currently, PASS means either that the image is complex and referenced as `:::image:::` with `type="complex"`, or the image is standard or icon and referenced appropriately in standard Markdown or the `:::image:::` extension. FAIL means the image is complex but is not referenced as `:::image:::` with `type="complex"`. Undetermined means the model can't confidently categorize the image.
PROBABILITY	The confidence, between 0 and 1, in the image categorization. If PROBABILITY is less than 0.92, ACTUAL TYPE is returned as "Undetermined".

4. Fix any failures by updating the image reference in the source article. That is, if you have a complex image that isn't referenced as complex, update the reference to use the `:::image:::` syntax with `type="complex"` and a long description as described in Complex images with long descriptions.

If any of your images are incorrectly categorized or "Undetermined", please help improve the model! File a bug via https://sitehelp.microsoft.com/ and provide:

- The source file path, including the repo.
- The image reference.
- The ACTUAL TYPE according to the model.
- The expected actual type (for example, "the tool says this is Undetermined, but it should be complex").
- Any other relevant details.

Known issues

- DocsImageSorter doesn't work as expected on recent versions of MacOS. The Engineering team can't prioritize fixes at this time.

Automated archive solution

4/16/2021 • 4 minutes to read

This article describes how to setup the machine to use the automated archive tool. The tool will allow archiving the content from current repo to archive repo. The tool works both in GitHub and Azure DevOps repo. The tool will create two PRs one in current repo and second one in archive repo.

System preparation and requirement

Install Microsoft .NetCore SDK: Go to https://dotnet.microsoft.com/download/dotnet-core/thank-you/sdk-2.2.207-windows-x64-installer and install version .Net 2.2

> **NOTE**
>
> If you currently have a more recent version (3.0, 3.1, etc.), install both, as 2.2 is currently required for the Archive tool.

Install Git: Install latest Git version from https://git-scm.com/download.

Setup Git: Run following command.

```
git config -list
```

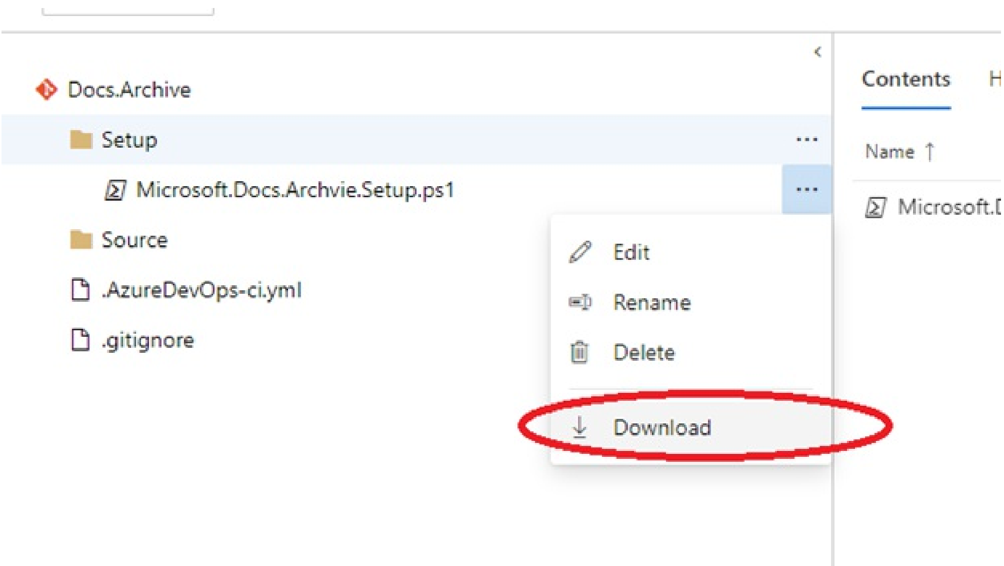
If "user email" and "user name" is not configured, run following command:

```
git config -global user.email "{your email address}"
```

```
git config-global user.name "{your user name}"
```

Download source files and setup scripts

Download Microsoft.Docs.Archive.Setup.ps1 from https://dev.azure.com/ceapex/Engineering/_git/Docs.Archive?path=%2FSetup%2FMicrosoft.Docs.Archive.Setup.ps1&version=GBmaster



Setup PowerShell execution policy Setup PowerShell execution policy to bypass, more information at https://docs.microsoft.com/powershell/module/microsoft.powershell.security/set-executionpolicy

Open PowerShell console as administrator

Run: Set-ExecutionPolicy Bypass

Unblock setup script

Open PowerShell console as administrator.

Unblock-File .\Microsoft.Docs.Archive.Setup.ps1

Installation

1. Open PowerShell window
2. Run the downloaded PowerShell script with "install" parameter.

```
.\Microsoft.Docs.Archive.Setup.ps1 install
```

How to create configuration file

To create a sample configuration file you can use following command.

```
PS C:\Users\username> docs.archive generateSampleConfiguration -o C:\Archive\sample
```

CONFIGURATION ELEMENTS	ELEMENT DETAILS
WorkingFolder	Setup the working folder. All file operations will happen under this folder. If not provided, current directory will be the working folder. E.g. D:\archive
Source	Current repo information
OpBuildUserToken	OpBuild User Token to operate on source repository. You can get the token from cookie in ops portal: https://ops.microsoft.com. The token lasts for 24 hour
PersonalAccessToken	Personal Access Token for AzureDevOps or Github. You can get the token from AzureDevOps or Github security settings.
SourceRepository	Archive source repository url. e.g. https://github.com/MicrosoftDocs/azure-docs-pr or https://cpubwin.visualstudio.com/_git/cortana.
DestinationBranch	The branch name that the archiving will be worked on and the final pull request targets. Live branch is not allowed.
DocsetName	Archiving source docset name.
Files	The file list to archive, example from Azure-docs-pr `articles\\storage\\common\\storage-import-export-tool-preparing-hard-drives-import-v1.md` . The tool also supports wild cards list.
TocNodes	Example from repo https://github.com/MicrosoftDocs/globalization "FilePath": "globalization\toc.yml", "TocPaths": "Globalization Overview\Documentation\Data Encoding"
Target	All information in this section is about archive repo

CONFIGURATION ELEMENTS	ELEMENT DETAILS
RepositoryType	AzureDevOps or Github
Account	Github or AzureDevOps account name. e.g. MicrosoftDocs, docs-archive
AzureDevOpsProject	AzureDevOps project name. e.g. docs-archive-project, For GitHub, leave empty.
RepositoryName	Target repository name, it can be an existing archive repository for example azure-docs-archive-pr or brand new one.
IsPrivate	the value can be "true" or "false"
DestinationBranch	The branch name that the archived content will be placed and the final pull request targets. Live branch is not allowed.
DocsetName	It can be an existing docset name for example Azure-docs-archive or new docset
BasePath	Base url path for the archiving target docset. e.g. /previous-versions/. Domain name https://docs.microsoft.com is not allowed. If the docset has already been provisioned, this is not required.
DocsetFolder	The folder name for archiving target docset. If the docset has already been provisioned, this is not required.
SubFolder	If SubFolder provided, the tool creates a sub folder in docset root and all archived items will be put into the sub folder. Not a required field.
GlobalMetadata	Any metadata that you want to add globally, you must enter "ROBOTS": "NOINDEX,NOFOLLOW" and "is_archived": "true"

Different run options

How to run archive tool

Open command window, run below command:

```
docs.archive archive -c <configuration file and path>
```

> **NOTE**
> One config maps one folder. If you run a specified archiving from working folder "A", you should not run other archive for different source repository or target repository on this "A" folder again. You should create a new folder or clean up "A" before you run another archive. But , if you run archiving against the same source and target repository again and again, you can still use the folder "A" and no need to clean it up.

How to get latest version of the tool

Run the downloaded Powershell script with parameter update `.\Microsoft.Docs.Archive.Setup.ps1 update`

How to uninstall the archive tool

Run the downloaded Powershell script with parameter uninstall install

```
.\Microsoft.Docs.Archive.Setup.ps1 uninstall
```

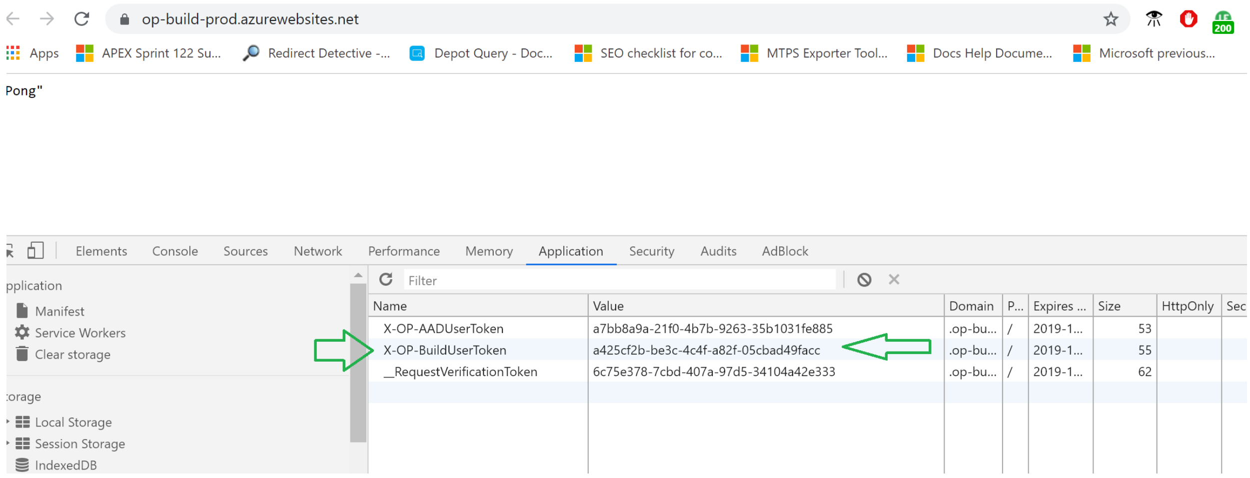
How to get the OpBuildUserToken

1. Open Docs portal

2. Sign in with "GitHub" or "Azure DevOps" according to your repository type

3. Go to https://op-build-prod.azurewebsites.net/ in Chrome and press F12 to open Developer Tool, find the cookie in Application\Cookies\https://op-build-prod.azurewebsites.net)

4. The value of the cookie is OpBuildUserToken and is used both in source and target section of the config file.



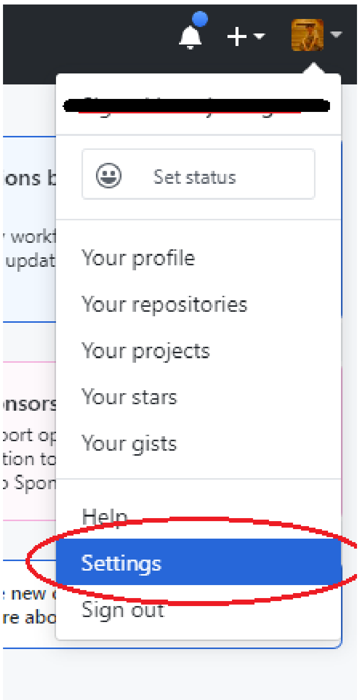
> **NOTE**
>
> The token will expire after 24 hours so make sure to get the new token every day otherwise you will get either 401 or you don't have access error.

How to get PersonalAccessToken

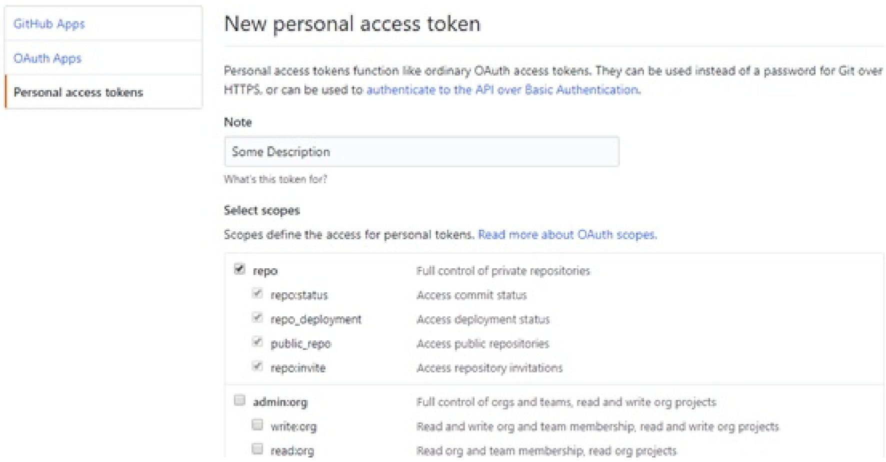
You will need to know this personal access token for both GitHub and Azure DevOps.

GitHub

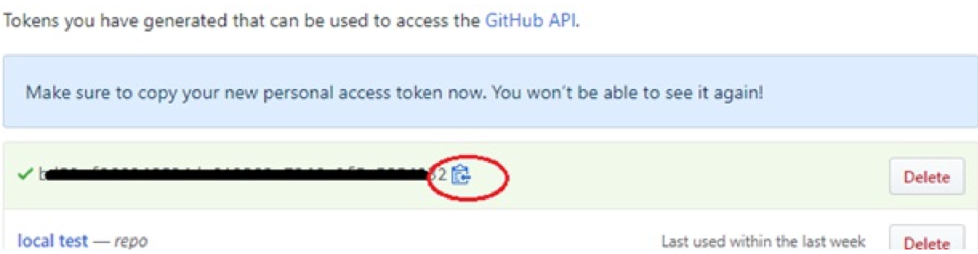
- First step is Logon to GitHub and go to settings.

- 

- Select "Developer settings"
- Select "Personal access tokens"
- Click on "Generate new token" (you may be required to enter your Github credential)
- Provide some description and make sure "repo" is selected, click "Generate token" button below the page.



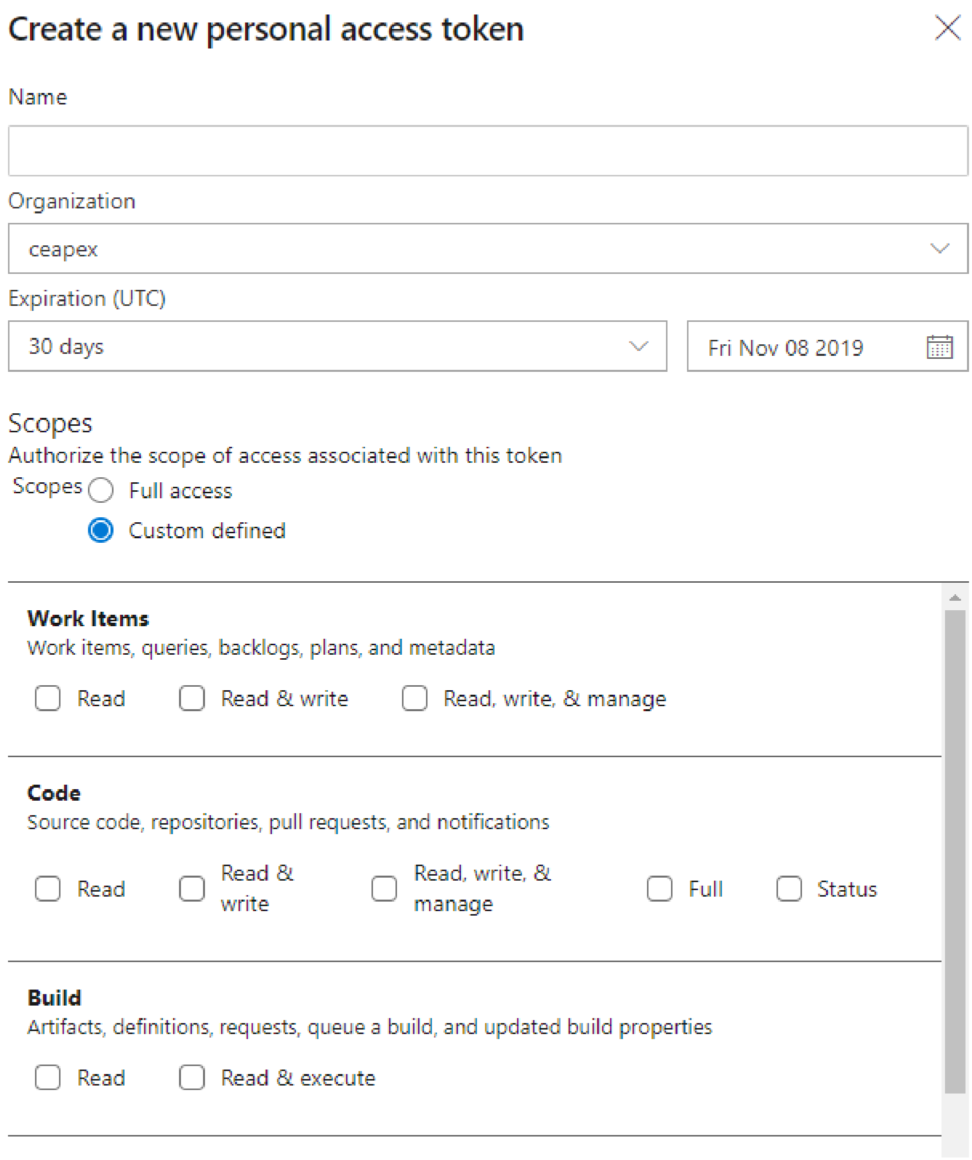
- Copy the token value and keep it in a safe location for future use, you will need this every time you will run archive tool.



Azure DevOps

- Open Azure DevOps project, e.g. https://dev.azure.com/ceapex/
- From Login drop down select security option.
- Select "Personal access tokens"
- Click on "New Token" button

- Provide token name, target organization and expiration.
- In Scopes section, make sure all the scopes under "Code" is checked.



- Click on "Create" button below
- Copy the token value

Use CodeFlow to review PRs

1/21/2020 • 3 minutes to read

Reviewers can use CodeFlow as a tool to review pull requests (PRs) in content repositories. CodeFlow is an alternative to the GitHub web UI PR review tools - users who already use CodeFlow may prefer to use a tool they're familiar with.

Get started with CodeFlow

If you decide to use CodeFlow as your PR review tool, you should review the articleWorking with GitHub Pull Requests. The article includes:

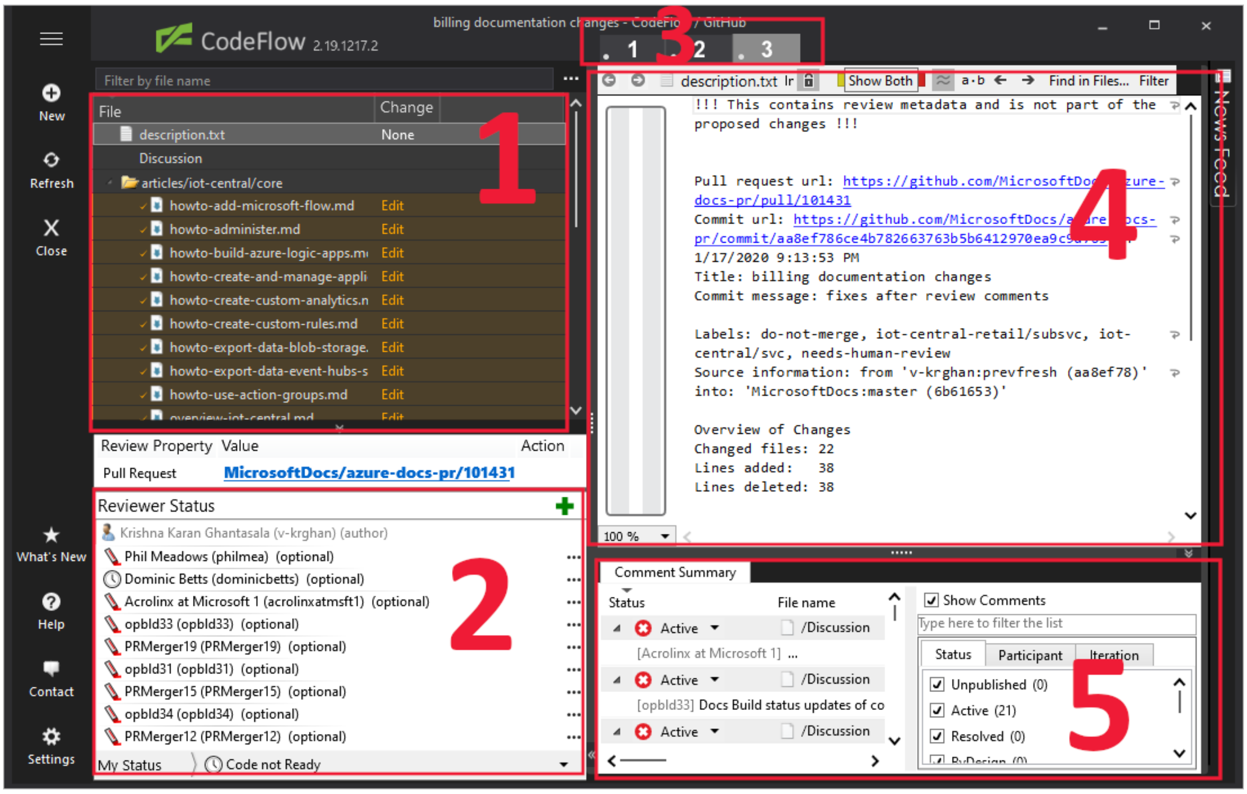
- One-time setup requirements to configure CodeFlow with your GitHub credentials.
- Opening pull requests using Chrome or Firefox extensions, or by using a console command.

> **TIP**
>
> The Edge browser can now install extensions from the Chrome Web Store.

View a PR in CodeFlow

The following screenshot shows the CodeFlow UI with an open PR from the **azure-docs-pr** GitHub repository:

The key areas of the UI are:



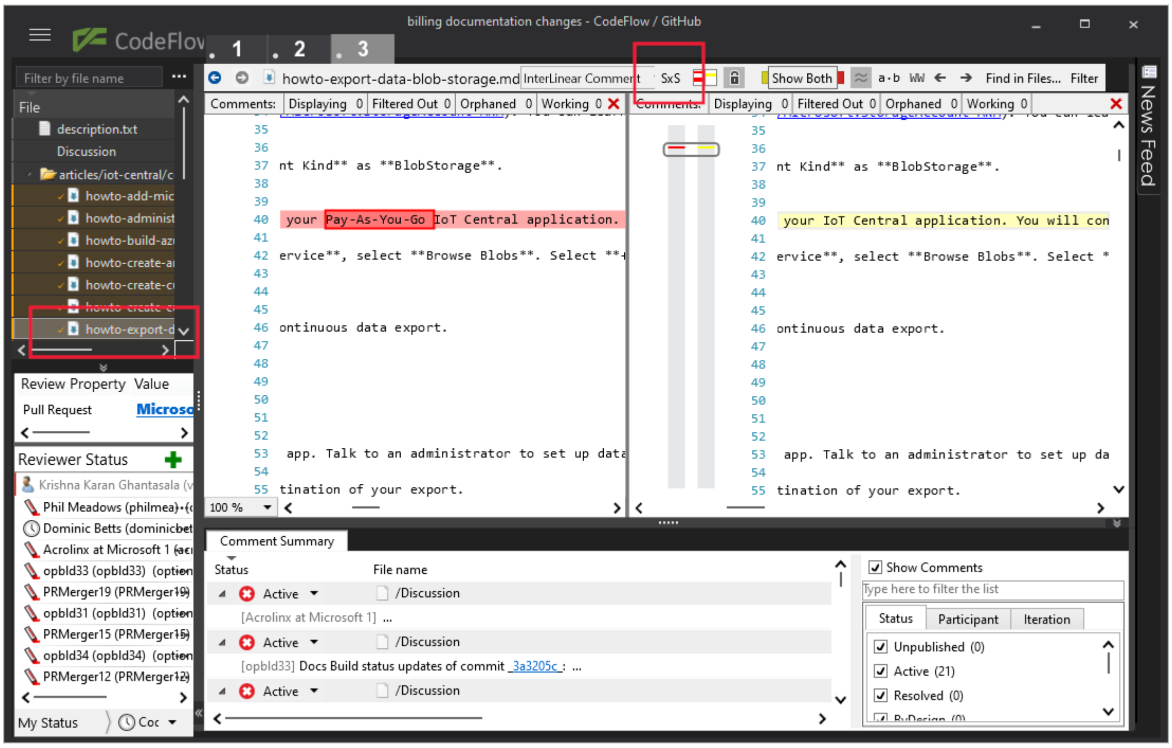
1. The list of modified files included in the PR. **Description.txt** contains generated metadata and isn't part of the proposed changes. **Discussion** includes the Acrolinx scorecard and staging links.

2. The list of reviewers associated with the PR. The PR author is listed first. Acrolinx and the validation bot

appear here when they have run. You can view the comments generated by Acrolinx and the validation bot if you choose **Discussion** in the file panel.

3. There's a tab for each commit in the PR that enables you to view the state of the files at that point. You can see the difference between two commits by Ctrl-clicking on two tabs. By default, CodeFlow displays the most recent commit with the final state of the files in the PR.

4. The file viewer/comment area enables you to view and comment on the files in the commit. Use the toolbar to control how the files, differences, and comments are displayed. The following screenshot shows the side-by-side *diff* view:



5. The **Comment Summary** area shows a list of the comments made in the PR. You can use the checkboxes on the right to filter by comment status, participant, and iteration. Iteration refers to the commit the comment was made against.

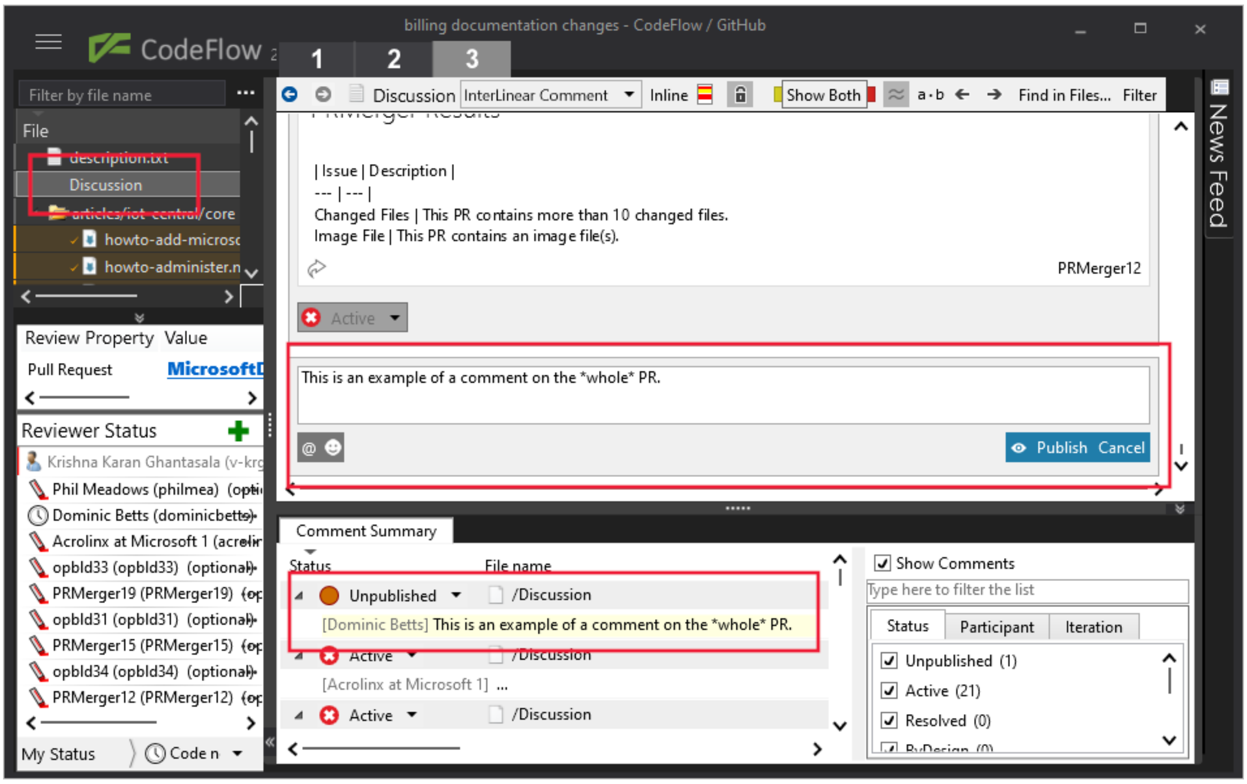
Add comments to a PR in CodeFlow

As a reviewer, you can add comments to a PR in CodeFlow in the following ways:

You can comment on the PR as a whole

1. Choose **Discussion** in the **File** panel. This panel is where the Acrolinx scorecard and validation links display.
2. Right-click in the file viewer/comment area and choose **Add comment**.
3. Enter your comment and choose **Publish**.
4. The comment appears in the **Comment summary** area in the **Discussion** iteration.

> **TIP**
>
> Use markdown to add formatting to your comments.

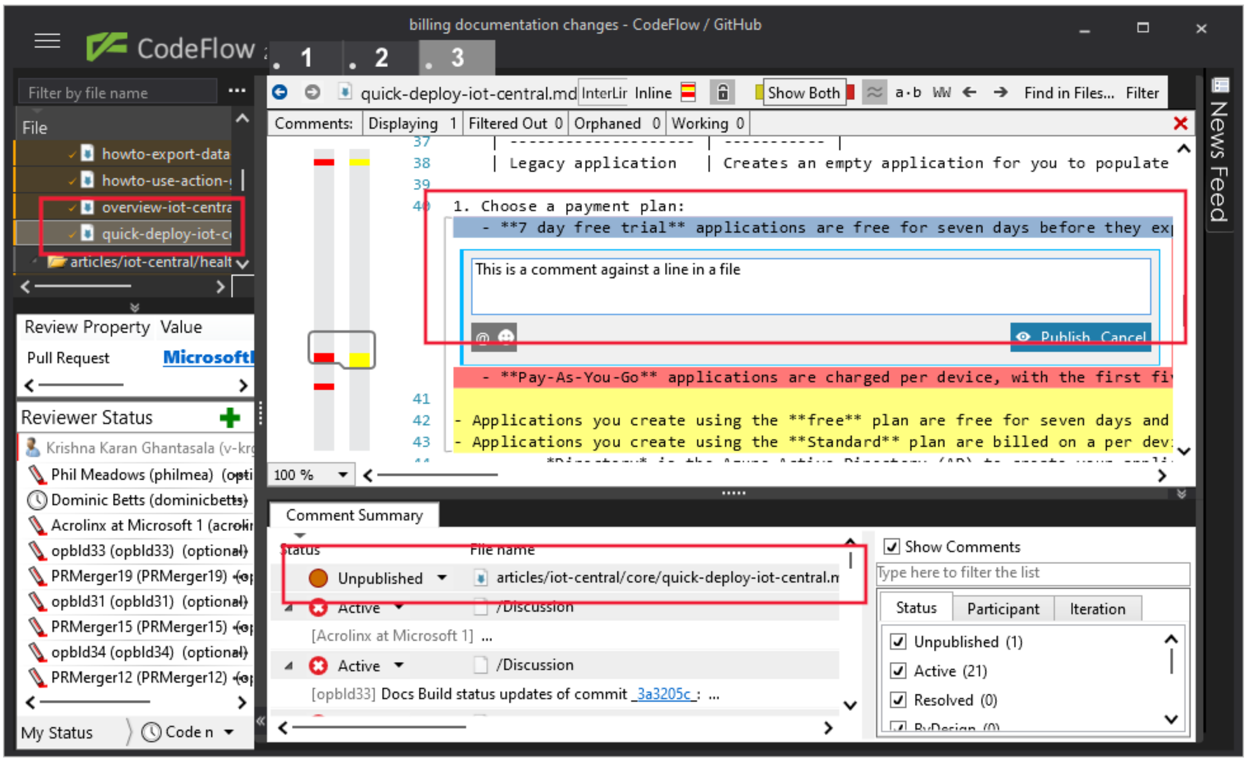


You can comment on a specific line in a file in the PR

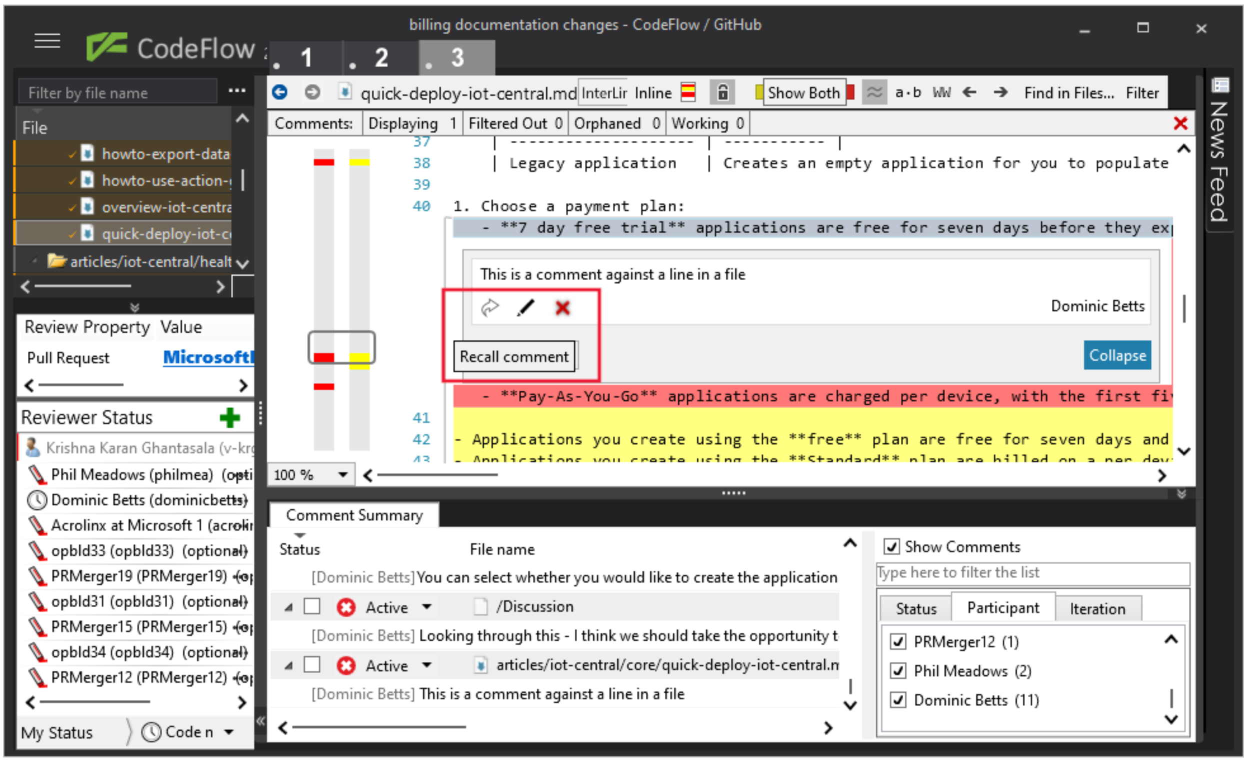
You can comment on any line in a file that's included in the PR, whether or not it was changed in the PR.

1. Choose the file you want to add a comment to in the **File** panel.
2. Right-click on a line in the file viewer/comment area and choose **Add comment**.
3. Enter your comment and choose **Publish**.
4. The comment appears in the **Comment summary** area in the **Discussion** iteration.

> **TIP**
>
> Use markdown to add formatting to your comments.

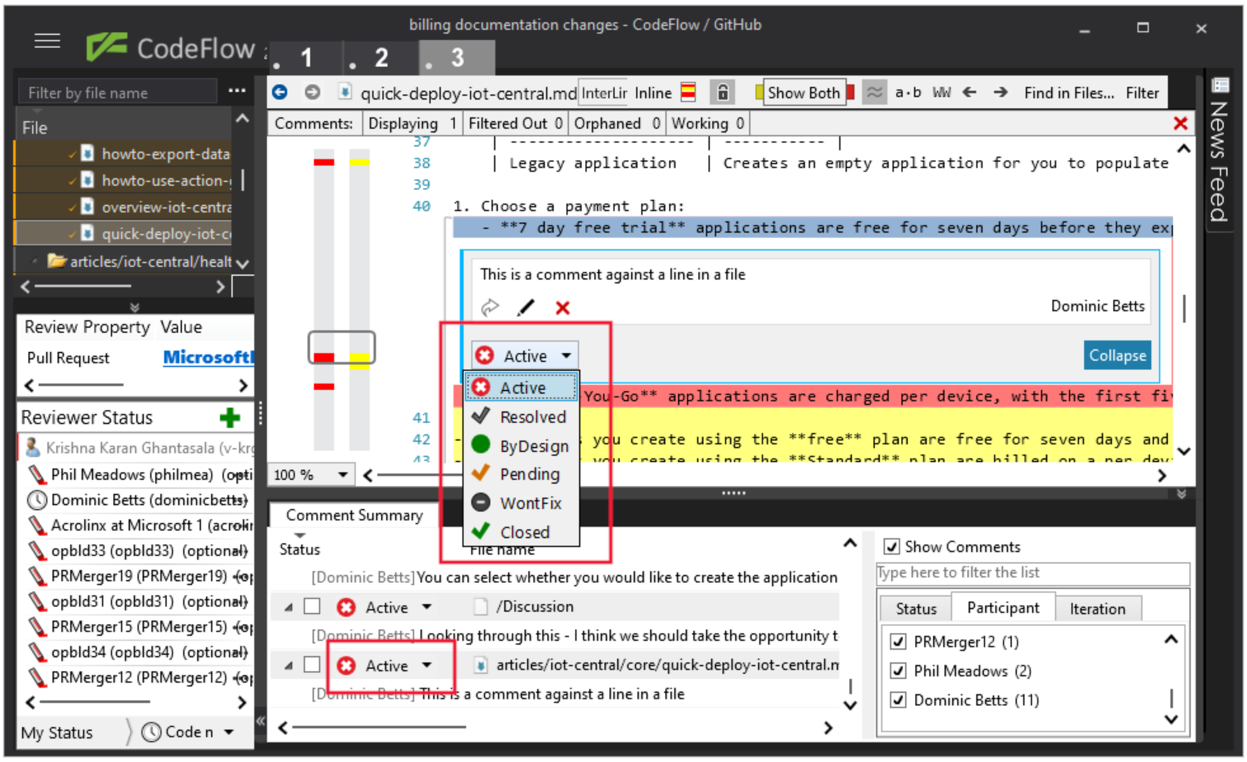


As a comment author, you edit or recall a comment you created. If the comment was created by another reviewer, you can post a reply:



All the comments are saved in the PR itself, CodeFlow doesn't use any external store when you use it to review a GitHub PR.

You can change the status of the comments you made in the PR:



You can change your status as reviewer in the **Reviewer Status** panel as follows:

- **Code not Ready** indicates that you expect the author to make changes before signing off the PR.
- **Approved** indicates that you approve the changes and the author can sign off the PR.

Manage the review process in CodeFlow

Invite reviewers

As the PR author, you can invite reviewers from within CodeFlow by adding a valid GitHub username in the **Reviewer status** area. As an alternative, you can email reviewers asking them to provide feedback on your PR.

In both cases, you must explain to users how to open the PR in CodeFlow. If you send an email, you can include the console command to run, for example:

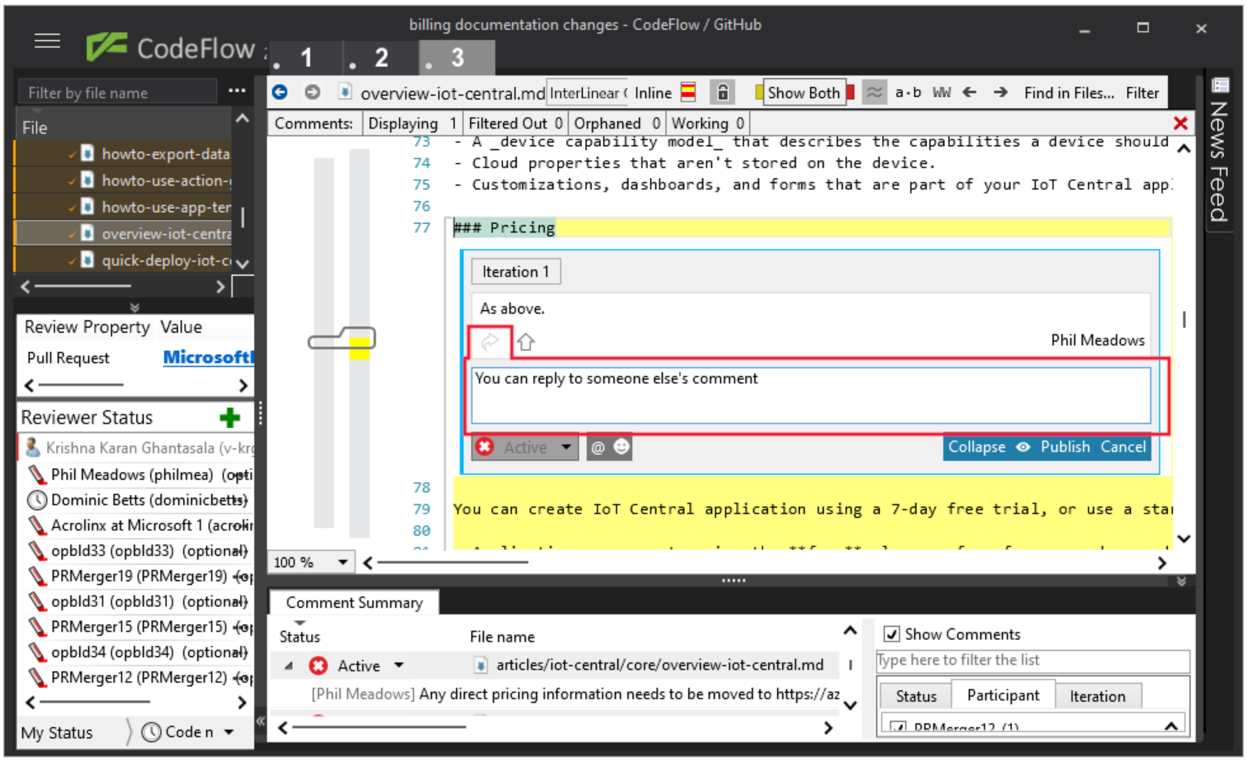
```
\\CodeFlow\public\cf.cmd openGitHubPR -webUrl https://github.com/MicrosoftDocs/azure-docs-pr/pull/16379
```

Manage comments

As the PR author, you can use the tools in CodeFlow to manage and track the comments added by your reviewers.

You can use the filters in the **Comment summary** area to find specific comments.

You can reply to comments made by your reviewers to request more information or a clarification:



Next steps

You can learn more about using CodeFlow to review GitHub PRs in Working with GitHub Pull Requests.

Acrolinx overview

6/9/2021 • 2 minutes to read

Acrolinx is editing software. It automates feedback on brand, terminology, grammar, spelling, punctuation, style, and voice. It's used by a partnership within Microsoft: Content & Learning (C&L), Microsoft 365 (MARVEL), Worldwide Learning (WWL), and the Business Applications Group (BAG). The goals are to help content contributors improve their writing and to make technical content on `docs.microsoft.com` clearer and more consistent.

- **GitHub**: You can use Acrolinx in GitHub. You get results when you submit a pull request (PR). Acrolinx runs automatically and writes results to the PR comments. Depending on your team, you must reach a minimum score before your PR is merged. For C&L, the minimum score is 80. You don't need to install anything to use Acrolinx in GitHub. If your team uses Acrolinx, they'll inform you of the process
- **Visual Studio Code**: You can use Acrolinx locally in Visual Studio Code. Many contributors prefer this option since they get feedback before submitting their PRs, and the experience is more interactive.

Criteria for using Acrolinx

Members of the partnership (C&L MARVEL, WWL, BAG) can add support to their repos since they've contributed to the budget. But their content must meet these criteria:

- Technical documentation for a Microsoft product.
- Conceptual in nature.
- In a GitHub repository and authored in Markdown.
- Published using the OPS publishing system.
- Published to `docs.microsoft.com`, `docs.microsoft.com/learn`, or to the official technical documentation library for a Microsoft product on another Microsoft-owned web property.
- Within the Microsoft Docs organization in GitHub. If the repository is in a different organization, a strong business justification must exist.
- The content team agrees to the minimum score set by their team.

Limitations:

- Acrolinx doesn't work with autogenerated reference content.
- Windows: Only these Windows technologies in C&L are covered by Acrolinx: Windows Developer (SDK), Windows Device Drivers (WDK), and Windows Server.
- Don't use Acrolinx for products outside the partnership because doing so violates our license.

Covered repositories

See the list of repositories and coverage.

Reporting bugs and false positive results

Report bugs and false positives at https://aka.ms/acrolinxbug. Issues are triaged once per week (and longer than that during holidays).

If you can't access the link above, you need to request access to CEAPEX.

Acrolinx in Visual Studio Code

6/1/2021 • 4 minutes to read

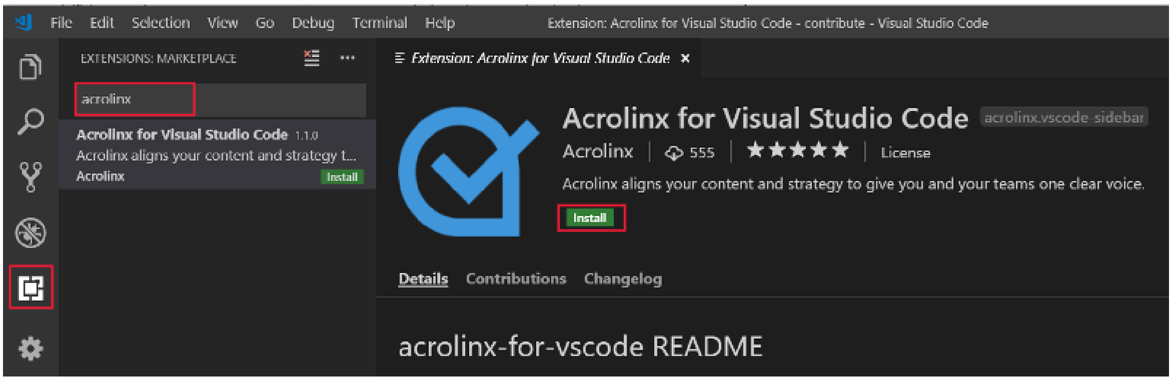
The Acrolinx extension for Visual Studio Code performs the same check that GitHub does: spelling, grammar, style, tone, clarity, and key terminology. This article explains how to install and run the Acrolinx extension in Visual Studio Code.

You can only use Acrolinx in Visual Studio Code in repositories that fit the criteria and are part of the partnership at Microsoft:

- Content & Learning (C&L)
- MARVEL
- Worldwide Learning
- Business Applications Group (BAG)

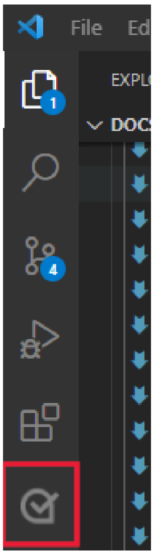
Install Acrolinx in Visual Studio Code

1. Start Visual Studio Code.
2. Update to the latest version if prompted.
3. Select the square icon in the left-hand bar to open the **Extensions** pane.
4. Type *acrolinx* in the **Search Extensions in the Marketplace** box, and select the **Acrolinx for Visual Studio Code** result.
5. Select the green **Install** button.



Configure Acrolinx

1. The first time you run Acrolinx, you need to configure the server details. To do so, open an *.md* or a *.txt* file to make the Acrolinx icon visible on the left sidebar:
2. Open an *.md* or a *.txt* file and select the Acrolinx icon from the left-hand menu:



An Acrolinx window appears, displaying the sign-in page.

3. Select **Change Server** and paste in the following **Server address**:

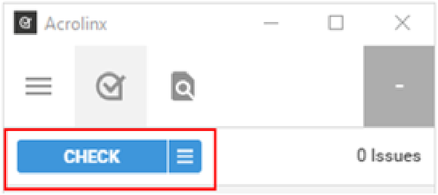
```
https://microsoft-ce-csi.acrolinx.cloud:443
```

4. Select **Connect** > **Sign In**.

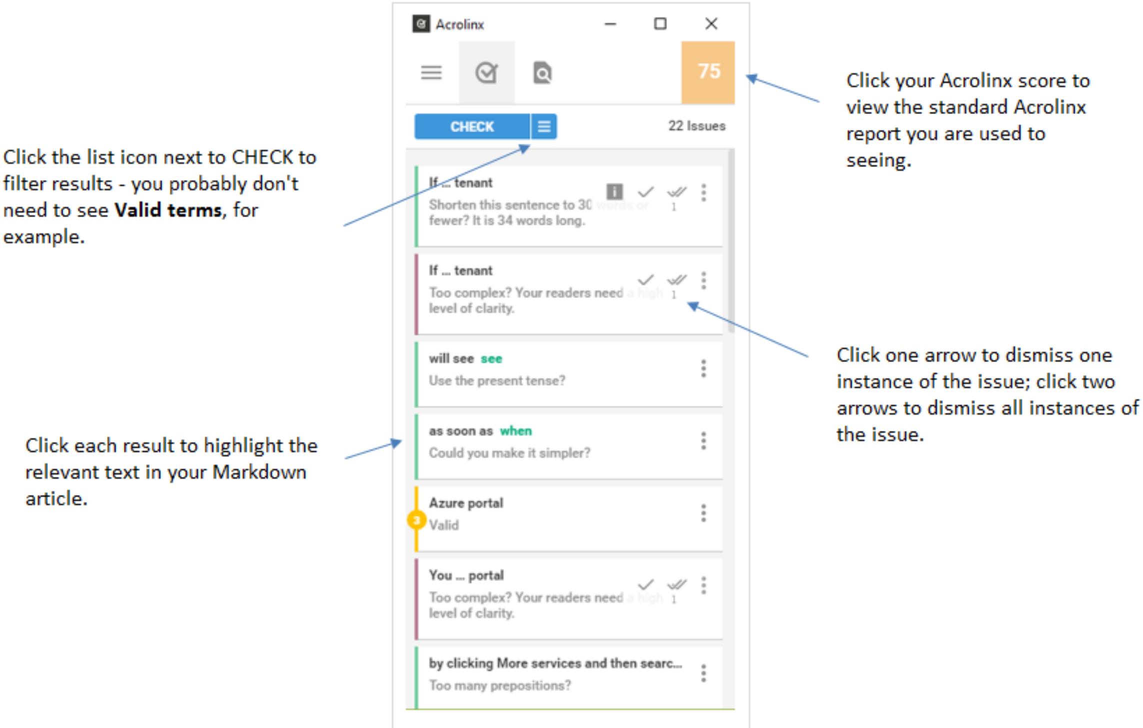
   Acrolinx uses Microsoft corpnet authentication, so sign in using your Microsoft credentials. As part of this process, you have to authorize Acrolinx for Visual Studio Code on the Acrolinx site - just follow the prompts.

5. Select **Options**. In the **Options** pane, select the correct guidance profile.

6. Select the **Check** button to check the document:



7. Review the results



Known issues with v1.1

- The Acrolinx extension version 1.1 for Visual Studio Code doesn't recognize Markdown files on a Mac. If you use a Mac, your only options are to rely on the GitHub integration, revert to version 1.0, or find a Windows computer.
- On Windows computers, you might receive a message that Acrolinx can't recognize the file type. The recommended workaround from Acrolinx is to select the Acrolinx sidebar "button" again before running a check. This needs to be done every time a different document is to be checked. These other tricks might also get you unblocked:
  - Open an XML file and try to run Acrolinx, then go back to the Markdown file and try again.
  - Change the editor layout to "split right" and back to "single."
- If you have multiple windows open, Acrolinx reports on the first window opened. To get it working, close all the editor windows that you don't want to analyze, then restart Visual Studio Code.

Install Version 1.0

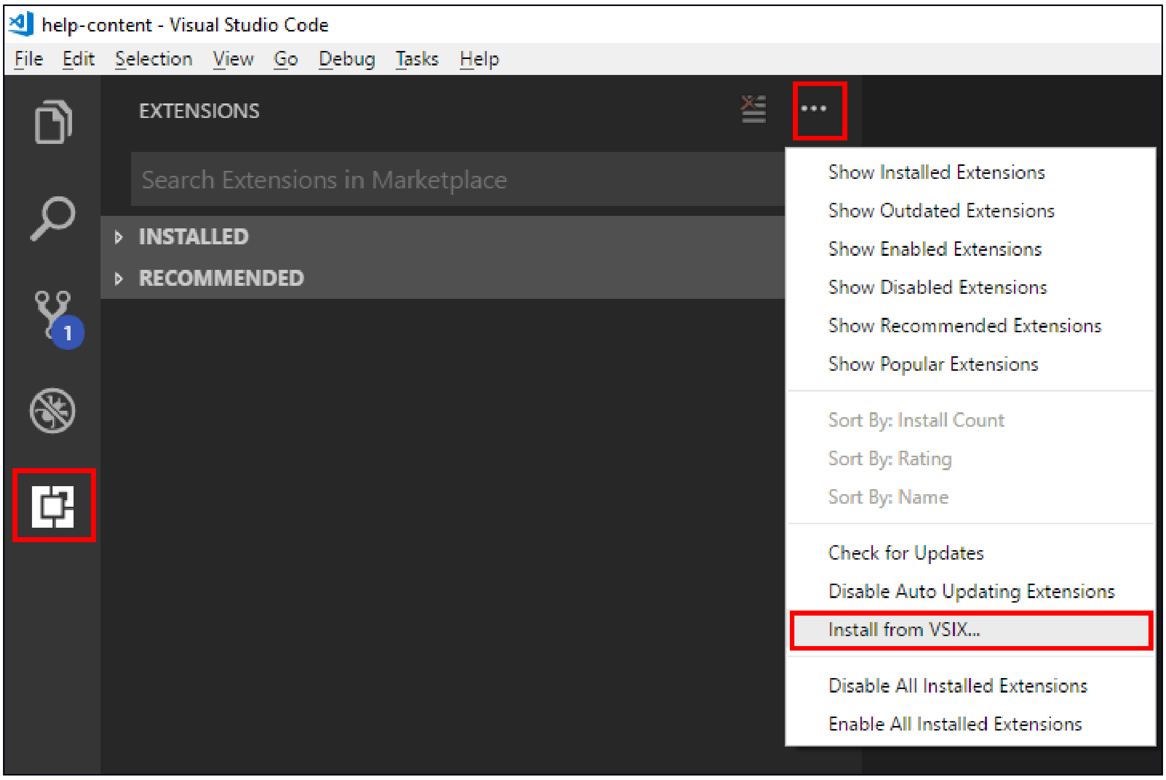
Prerequisite for Mac

On macOS, you need to copy the VSIX file locally using the SMB share before installation. To do this:

1. Open Finder and choose **Go** > **Connect to Server**.
2. Enter the **Server Address** of the file: `smb://dpspub/Acrolinx/`. Make sure you're connected to corpnet.
3. Enter your credentials in form `redmond\<alias>` (for `redmond` domain) and your password
4. Select **Connect**, and copy the file locally to your Mac.

Installation steps

1. Start Visual Studio Code.
2. Update to the latest version if prompted.
3. Select the square icon in the left-hand bar to open the **Extensions** pane.
4. Select the three dots for "More" and select "Install from VSIX..."



5. For Windows, paste the path `\\dpspub\Acrolinx\VisualStudioCodeV1.0.1_B87` into the **File** name box and select **Install**.

   On a Mac, locate the saved VSIX file and then select **Install**.

   > **NOTE**
   > The extension is large and can take about a minute to install. There is no indication of activity during this time.

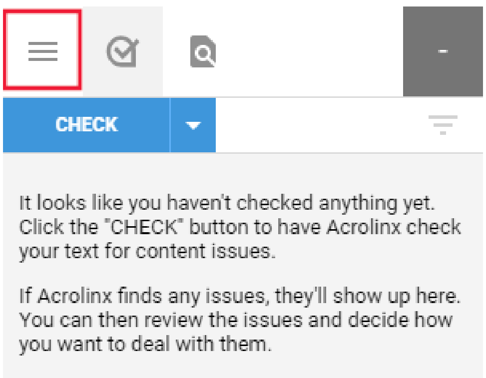
6. When installation is complete, Visual Studio Code may prompt you to reload. The first time you run Acrolinx, you need to configure it.

Change your guidance profile

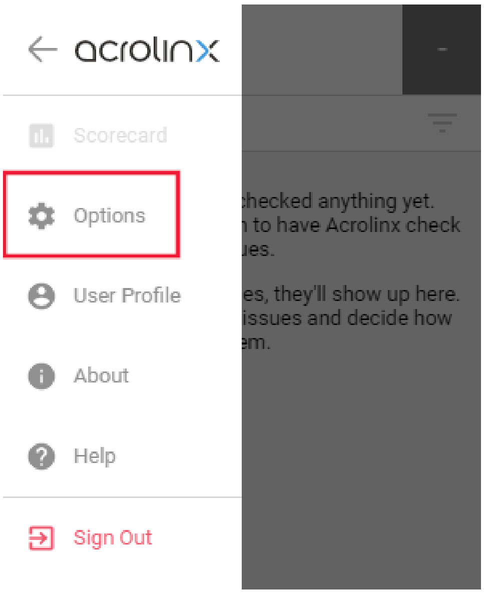
All content contributors must make sure they're using the correct guidance profile in Acrolinx. The guidance profile includes a specific term list and other guidelines that Acrolinx uses to score the content you're working on when you submit a pull request. Admins assign guidance profiles to each repo in GitHub.

If you work in repos for more than one content group, you must change your guidance profile each time you change orgs. Other guidance profiles may be added in the future.

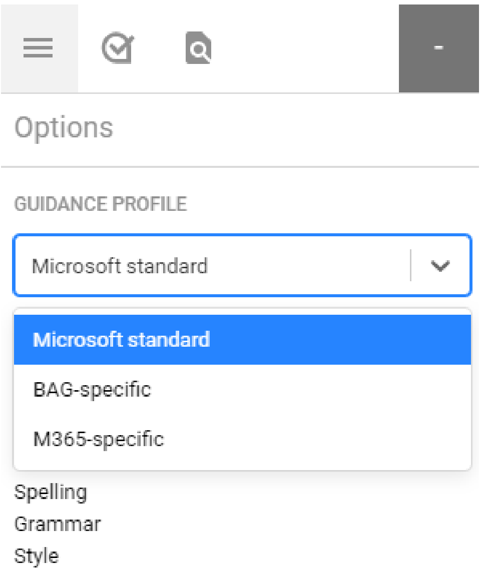
1. In Visual Studio Code, open an *.md* or a *.txt* file to make the Acrolinx icon visible on the menu bar. Open the Acrolinx sidebar and select the slide-out menu.



2. Select **Options**.



3. In the **Options** pane, select the correct guidance profile:



REPO	GUIDANCE PROFILE
C&L	Microsoft standard
MARVEL	M365-specific
BAG	BAG-specific

Report issues, bugs, or false positives

- Check the troubleshooting guide
- File a bug or suggestion using this link: https://aka.ms/acrolinxbug

Guidance for using Acrolinx and respecting the license

Use the Acrolinx extension only in repositories that are licensed for Acrolinx under the Content & Learning-Acrolinx contract.

- When you run Acrolinx locally using Visual Studio code, run it only on local repository clones for covered repositories.
- The path on your local computer must include the correct name of the authorized repository. For example, if you have a local clone of azure-docs-pr, the path on your computer must include "azure-docs-pr". The path is necessary for license compliance and accurate data on Acrolinx usage and scores. For repositories that have non-unique names ("docs", for instance), your local clone must be in a local path that includes the organization. The ASP.NET and .NET documentation repositories must be cloned into a location that includes aspnet/docs or dotnet/docs in the local file path.
- Don't put content from an unauthorized repo into an authorized repo just to get Acrolinx results. This screws up the data around scorecards and scoring for content in the repository you use to do this. It is outside the licensing agreement, and it corrupts the usage data.

Minimum Acrolinx scores

5/10/2021 • 3 minutes to read

For **Content & Learning**, in the repositories contributors work in, the pull request reviewers check all pull requests that require human review. They make sure each article in a pull request meets the minimum Acrolinx score before they merge the changes. All pull requests that require human review in C+L must meet the minimum Acrolinx score.

Note: Min score requirements for M365 and BAG will be added once we are all working from the same Acrolinx instance, scheduled for the beginning of November.

How the minimum score works for C+L

- The starting minimum score for C+L is 80. All the articles in a PR need to meet the minimum score for the PR review team to merge the PR.
- The minimum score applies on a going-forward basis - as you update content within your normal workflow, you will need to bring each article up to the minimum score (higher is better!). Everyone should address the spelling, tone of voice, and clarity feedback in Acrolinx reports. This feedback from Acrolinx is designed to make our content more trustworthy, easier to understand, and with a more conversational tone. The Acrolinx recommendations directly align with Microsoft's modern voice principles. We're doing this for our customers' benefit.
- You do have to fix pre-existing content as you make updates.

Dealing with false positive results

- We use the term "false positive" to refer to cases where Acrolinx flags something as incorrect or needing a fix, but where you know that the usage is correct for the context. The minimum score is intended to accommodate for some false positives because we have to live with some false positives. It's not possible or even desirable to add every term that generates a false positive to the system. You can request that terms be added when the terms are used broadly in a content set.
- **For Content & Learning,** if lots of false positives in a pull request scorecard put the score of an article below 80 and cause your PR to be blocked, take these steps:
  - File a bug for the false positives using this link: https://aka.ms/acrolinxbug. Make sure you include a link to the PR, a link to the scorecard, and a clear description of why the flags are false positives
  - Enter a comment in the pull request to let the PR reviewers know that a false positive in the results is causing the PR to have a score lower than 80. Include a link to the Acrolinx bug. As long as you have filed an Acrolinx bug for the false positives, the PR reviewers can review the PR against all other criteria and merge as appropriate.

Exceptions to the minimum score requirement

- **Bulk updates**: Bulk updates are exempt from the minimum score requirement. Examples of bulk updates include changing the author metadata across a group of articles or changing a product name across all the articles in a content set.
- **Localization fixes**: When a member of the localization team submits a fix, the localization PM is not responsible for updating the article to meet the Acrolinx min score requirement - that is outside the scope of their role. Usually, the comment in the PR makes it clear the change is a localization fix, and the

contributor's profile usually indicates they are affiliated with the localization team.

- **Emergency publishing/updates**: If you've been asked to make a high priority fix to an article to address a severe customer issue, or an executive, red zone, security, or privacy issue, you can request an exception to the score requirement if the issue is truly urgent and you cannot dedicate time to the fixes. This guidance isn't about false positives – this is about not fixing valid Acrolinx flags. (The process for handling false positives is described earlier in this article.) The content lead needs to approve the exemption. This will be handled on a case-by-case basis.

Acrolinx reports

9/24/2019 • 2 minutes to read

The Acrolinx report has been restored to the SkyEye site thanks to work by the CGA team and Acrolinx. This report gives scores and related details for articles with pull requests. To access the report, go to https://aka.ms/skyeye and select the Acrolinx Report tile.

What's in the Acrolinx report

- All content that has been run through the GitHub integration should be in the report
- VS Code does not write results to the report. So the report contains no info for repos that are public only and where VSCode is the only Acrolinx tooling

Baseline reports

Baseline reports give Acrolinx scores for all articles in a repo, not just articles with recent pull requests. This report is useful if you are trying to, for example, evaluate the quality of an existing repo that you inherited from another team.

- Baseline reports are available only for private repos currently
- The reports are run automatically when Acrolinx support is added to a repo
- You can request a baseline report for a repo with existing Acrolinx support

To request a baseline report file a request at https://aka.ms/acrolinxbug.

Questions

If you have questions, contact monicar.

Troubleshooting issues with Acrolinx

6/1/2021 • 2 minutes to read

If you have problems authenticating to Acrolinx or if you receive an error message, this info will help you resolve the issue.

Authentication

- **When you select a scorecard in GitHub or VS Code for the first time**, you need to authenticate as a Microsoft user. You'll be directed to the Acrolinx site and asked to sign in. Just use your Microsoft alias and select **Sign in**.
- **Occasionally you have to authenticate again**. This error can happen if you change to another repo or browser. Just sign in again.
- **If you still can't authenticate**, file a bug here: https://aka.ms/acrolinxbug

Outages

- If Acrolinx is down, file a ticket at the SiteHelp portal or sign in to the Docs Support Channel in Teams. (In the SiteHelp portal, select **Submit request or issue** > **Select Service** > **OPS** > **Select Service Category** > **Acrolinx down or Acrolinx not accessible**)

Error messages

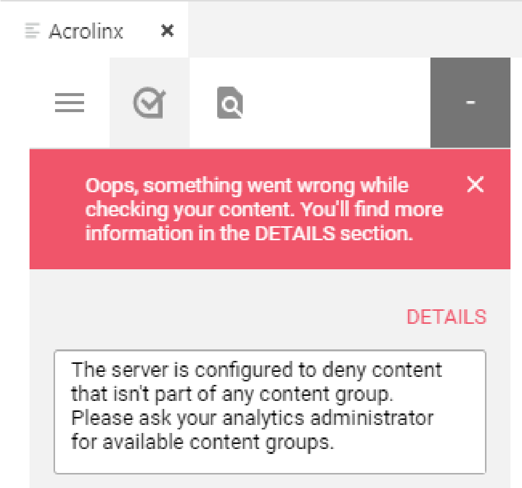
The most common error message appears if you're trying to run Acrolinx on content that doesn't have Acrolinx support. The error may also appear on content that isn't in the correct local path when using Acrolinx in VS Code.

When you're using Acrolinx in VS Code, after you select the **Check** button, you might receive the following message:

> *Oops, something went wrong while checking your content. You'll find more information in the DETAILS section.*

When you select **DETAILS**, you get the following additional information:

> *The server is configured to deny content that isn't part of any content group. Please ask your analytics administrator for available content groups.*



This problem occurs because you're trying to run Acrolinx on content that is not covered by the Acrolinx program. You can only use Acrolinx in VS Code in repositories that fit the criteria and are part of the partnership at Microsoft:

- Content & Learning
- MARVEL
- Worldwide Learning
- Business Applications Group

To resolve the problem:

- If you're working on content for a listed repository, you need to work on it within your local clone of the repository. The local clone's name must match the name of the upstream repository.
- If you're trying to run Acrolinx on content for a repository outside the Acrolinx program, you need to stop. Acrolinx can only be used for content in a covered repository. See the Acrolinx in Visual Studio Code article for details about how to use Acrolinx in alignment with our license.
- It's possible that you're using Acrolinx for content in a repository where Acrolinx is configured. The content group for your repository might be missing or have a problem. For this case, file a bug using https://aka.ms/acrolinxbug.

Use AutoHotkey automation for Git Issues and DevOps templates

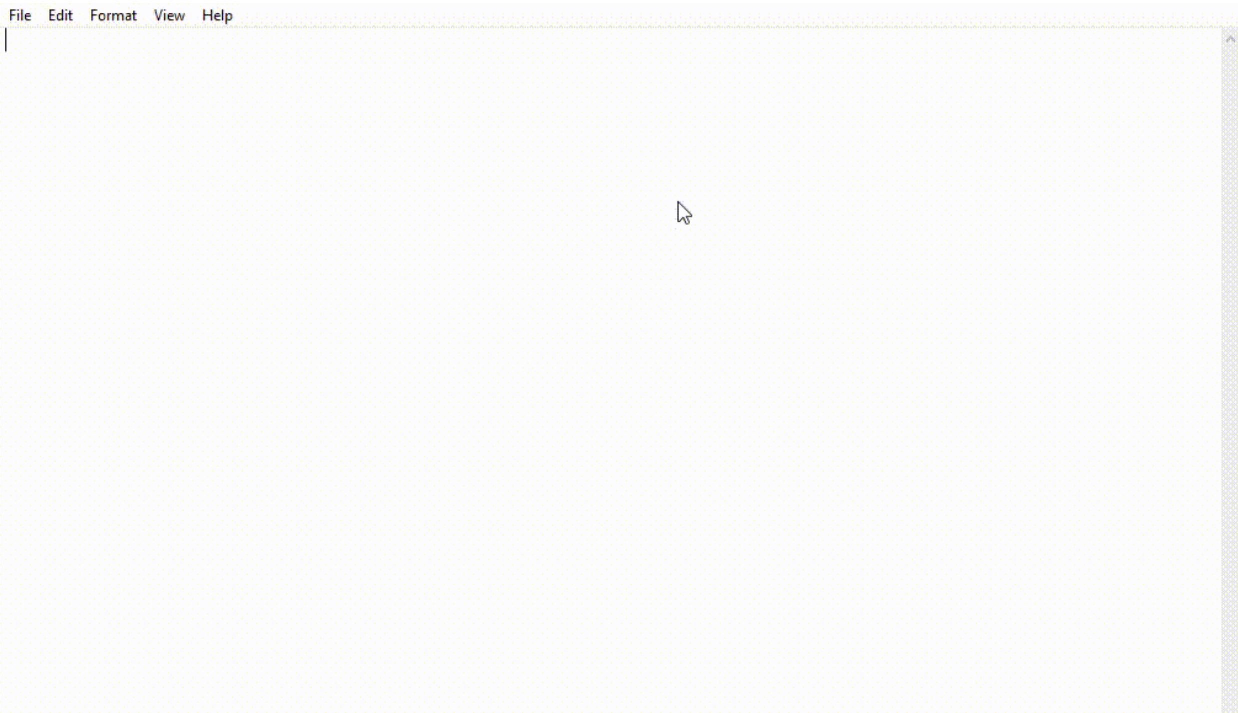
4/16/2021 • 23 minutes to read

AutoHotkey is a third-party OSS approved open-source automation tool that allows you to create templates from self-selected "keywords". The keywords are then automatically replaced with predefined text blocks and/or scripts in most text editors and browsers.

This topic describes how to use AutoHotkey to create automatic templates for responding to customers via Git Issues, and for populating Azure DevOps items. However, this tool can be used within most text editors, and scripts can be added manually to suit any other needs.

This topic is meant to be used as an introduction and foundation for configuring preliminary scripts. The scripts provided in this article are **completely modifiable**, and can be edited to suit your needs.

Video Example



When to use

AutoHotkey is an automation tool that facilitates the output of canned saved text. It can be used within anything that accepts text input, such as Azure DevOps items, web browsers, Word, Notepad, Notepad ++, etc. (except VS Code). We recommend using it to reply to Git Issues, and populate the "Description" section of an Azure DevOps item, such as within Azure Boards. This article is meant to be just a starting point to introduce you to AutoHotkey. You can create any type of script you want that suits any need you may have.

The scripts and Git Issue replies listed in this topic are just suggestions - they are by no means mandatory to use in either Git Issues or Azure DevOps items. However, they are likely to save you time and increase your efficiency.

How is AutoHotkey different?

- **VS Code / Docs authoring pack templates**: It is different than a VS Code template in that those are built into VS Code and just populate templates based on the content model, such as a quickstart. Autohotkey is not tied to a single program, and it populates text based on what you've configured it to.

- **Azure Boards Templates**: Templates used in Azure Boards create work items, and fill out fields such as the area and iteration path. AutoHotkey can be used to populate text within an Azure Board item, but it does not create new work items.

Installation

To use AutoHotkey, do the following:

1. Download and install AutoHotkey from: https://autohotkey.com/download/.
2. Create a new text file and change the extension from `.txt` to `.ahk`.
3. Copy various scripts from this topic into your `.ahk` file - modify keywords and templates as necessary. These templates were originally created for the SQL docs team so they've been generalized with underscores. You can either pick and choose individual script snippets, or you can copy the entire full script located at the end of the article.
4. Save the `.ahk` file.
5. Place a copy of the `.ahk` in your startup folder located at `C:\ProgramData\Microsoft\Windows\Start Menu\Programs\StartUp`. Security settings may prevent you from making changes to this file in this location, so you may need to have a working copy elsewhere that you use to replace the file in the startup folder after making changes to it.
6. Double-click the `.ahk` file to initiate the script - this step only needs to be done if you want to manually initiate the script (such as the first time you configure this, or if you're changing and testing the script). Otherwise, this step is done automatically for you each time your computer starts up as long as the .ahk file is in the startup folder.
7. Type in any of the keywords in any text editor to test the script. Unfortunately, AutoHotkey does not work in VS Code - for that, you'll need to create snippets within VS Code.

Azure DevOps scripts

Description section

The following script can be used as a template for the description section in an Azure Board item.

KEYWORD	OUTPUT
vnotee	Status ====== To do ====== Done ====== Git ====== Contacts ====== Details ======

Place the following script in your .ahk file to enable this keyword script:

```
:r0:vnotee::
(
Status
======

To do
======

Done
====

Git
====

Contacts
========

Details
=======
)
```

Discussion section

The following script can be used as a title for notes appended in the Discussion section. It will populate the current date and the word sprint - you'll need to manually add which sprint it is.

KEYWORD	OUTPUT
dnotee	`2019-05-01 | Sprint`

Place the following script in your .ahk file to enable this keyword script:

```
:r0:dnotee::

FormatTime, Time,, yyyy-MM-dd
Send %Time%
send {Space}| Sprint
return
```

Random tasks DevOps item

This is a useful template for a single monthly "Random Tasks" Azure Boards item meant to track Public PRs, Git Issues, and random tasks that are not substantial enough for their own user story. The top "To do" portion shows the tasks, Public PRs, and Git Issues still pending. Once an item is complete, it's moved down to the "Done" section.

KEYWORD	OUTPUT
rnotee	

KEYWORD	OUTPUT
====== To do ====== Tasks ------ 1. 2. 3. Public PRs ----------- 1. 2. 3. Git Issues ---------- 1. 2. 3. ====== Done ====== Tasks ------ Public PRs ----------- Git Issues ----------	

Place the following script in your .ahk file to enable this keyword script:

```
:r0:rnotee::
(
====================================================
                    To do
====================================================

Tasks
------
1.
2.
3.


Public PRs
-----------
1.
2.
3.


Git Issues
----------
1.
2.
3.


====================================================
                   Done
====================================================

Tasks
------

Public PRs
-----------

Git Issues
----------
)
```

Git Issue template scripts

For a new Git Issue

KEYWORD	OUTPUT

KEYWORD	OUTPUT
ishnew	Thanks for contacting Microsoft with feedback about our product documentation. Depending on the complexity, it could take from a few business days to a few weeks to look further into your question, issue, or suggestion. We'll contact you if we need further information, and you'll receive an email message each time we "@mention" you in a comment. If you're suggesting a simple change to the documentation, it might be faster to submit the suggestion yourself by creating a GitHub pull request (PR). Here's how: 1. On the page you have feedback about, select the **Edit** link at the top right. 1. On the next page, select the **Pencil** icon at the top right. 1. On the next page, in the **Edit file** text window, make your edits directly to the text you want to change. If you need help with formatting the new or changed text, see our Markdown Cheatsheet After you've made your edits, under **Commit changes**: a. In the first text box, enter a brief description of the change you've made. b. In the **Add an optional extended description** box, paste the link to your GitHub issue. 1. Select **Propose file change**. 1. On the **Comparing changes** page, select **Create pull request**. 1. On the **Open a pull request** page, select **Create pull request**. 1. On the **Open a pull request** page, select **Create pull request**. Allow a day or more for the article's author to review and approve your change or offer an alternative solution. By opening a pull request, you're helping the author add your approved change directly to the documentation. After the updated article is published, GitHub lists you as an article contributor. If your issue is more complex than a simple change in the documentation, that's okay. We'll continue to research your suggestion and then get back to you with our response. Thanks again for submitting your feedback. Your suggestions help improve our documentation.

Place the following script in your .ahk file to enable this keyword script:

```
:r0:ishnew::
(
Thanks for contacting Microsoft with feedback about our product documentation. Depending on the complexity,
it could take from a few business days to a few weeks to look further into your question, issue, or
suggestion. We'll contact you if we need further information, and you'll receive an email message each time
we "@mention" you in a comment.

If you're suggesting a simple change to the documentation, it might be faster to submit the suggestion
yourself by creating a GitHub pull request (PR). Here's how:

1. On the page you have feedback about, select the **Edit** link at the top right.
1. On the next page, select the **Pencil** icon at the top right.
1. On the next page, in the **Edit file** text window, make your edits directly to the text you want to
change.
    If you need help with formatting the new or changed text, see our [Markdown Cheatsheet]
(https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet).
1. After you've made your edits, under **Commit changes**:
    a. In the first text box, enter a brief description of the change you've made.
    b. In the **Add an optional extended description** box, paste the link to your GitHub issue.
1. Select **Propose file change**.
1. On the **Comparing changes** page, select **Create pull request**.
1. On the **Open a pull request** page, select **Create pull request**.

Allow a day or more for the article's author to review and approve your change or offer an alternative
solution.

By opening a pull request, you're helping the author add your approved change directly to the documentation.
After the updated article is published, GitHub lists you as an article contributor.

If your issue is more complex than a simple change in the documentation, that's okay.  We'll continue to
research your suggestion and then get back to you with our response.

Thanks again for submitting your feedback. Your suggestions help improve our documentation.
)
```

For a Git Issue intended for support

KEYWORD	OUTPUT
ishport	Thanks for contacting Microsoft about your issue. As we understand it, you're describing a problem with the product itself and not with the product documentation. To get assistance or support for your product issue, we recommend that you engage with the product community or open a ticket with Microsoft Support. For more information, see SQL Server help and feedback. To provide product suggestions or ideas for improvement, go to https://aka.ms/sqlfeedback. If you're submitting feedback about the product documentation, please reply to this comment to clarify your issue further. Otherwise, we'll proceed with closing out this Git issue within a few business days. We're sorry that the SQL Docs team was unable to assist you further with your issue, but we appreciate your reaching out to us. If you experience other issues in the future, we encourage you to contact us again.

Place the following script in your .ahk file to enable this keyword script. Replace the underscores with values that apply to your specific team:

```
:r0:ishport::
(
Thanks for contacting Microsoft about your issue. As we understand it, you're describing a problem with the product itself and not with the product documentation.

To get assistance or support for your product issue, we recommend that you engage with the product community or open a ticket with Microsoft Support. For more information, see __________________.

To provide product suggestions or ideas for improvement, go to ___________.

If you're submitting feedback about the product documentation, please reply to this comment to clarify your issue further. Otherwise, we'll proceed with closing out this Git issue within a few business days.

We're sorry that the _________ Docs team was unable to assist you further with your issue, but we appreciate your reaching out to us. If you experience other issues in the future, we encourage you to contact us again.
)
```

To close a Git Issue

KEYWORD	OUTPUT
ishclose	Thanks again for contacting Microsoft. We've submitted a documentation change based on your suggestion, and the updated article should be live by tomorrow. Please let us know if there’s anything more we can do for you. To follow up on this issue, leave us a comment with an @mention.

Place the following script in your .ahk file to enable this keyword script:

```
:r0:ishclose::
(
Thanks again for contacting Microsoft. We've submitted a documentation change based on your suggestion, and the updated article should be live by tomorrow. Please let us know if there’s anything more we can do for you. To follow up on this issue, leave us a comment with an @mention.
)
```

Templates for editing content

Editing a doc in azure-docs-pr

KEYWORD	OUTPUT

KEYWORD	OUTPUT
azprpr	It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following: 1. Navigate to the page of interest 2. Select "Edit" button at the top right 3. Modify the URL so that 'azure-docs' becomes 'azure-docs-pr' 4. Press enter to navigate to the newly-changed URL and reload the page 5. Select the pencil to edit the document 6. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet 7. Scroll down to the bottom, name your file change, and select "Propose file change" 8. Select "Create pull request" on the 'Open a pull request' page 9. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link within the PR 10. Once you're satisfied with your changes, either type #sign-off in the comments to start the merge/publish process, or if you'd like me to review the changes, then type @MashaMSFT in the comments so that I can review and sign off If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs organization. More information can be found here: https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master Feel free to ping me directly on Teams or over email if you run into any issues.

Place the following script in your .ahk file to enable this keyword script - modify the underscore to your GitHub alias:

```
:r0:azprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following:

1. Navigate to the page of interest
2. Select  "Edit" button at the top right
3. Modify the URL so that 'azure-docs' becomes 'azure-docs-pr'
4. Press enter to navigate to the newly-changed URL and reload the page
5. Select the pencil to edit the document
6. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
7. Scroll down to the bottom, name your file change, and select "Propose file change"
8. Select "Create pull request" on the 'Open a pull request' page
9. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link within the PR
10. Once you're satisfied with your changes, either type #sign-off in the comments to start the merge/publish process, or if you'd like me to review the changes, then type @______ in the comments so that I can review and sign off

If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs organization. More information can be found here:
https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master

Feel free to ping me directly on Teams or over email if you run into any issues.
)
```

Editing a doc in sql-docs-pr

KEYWORD	OUTPUT
sqlprpr	It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following: 1. Navigate to the page of interest 2. Select the "Edit" button at the top right 3. Modify the URL so that 'sql-docs' becomes 'sql-docs-pr' and 'live' becomes 'master' 5. Press enter to navigate to the newly-changed URL and reload the page 6. Select the pencil to edit the document 7. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet 8. Scroll down to the bottom, name your file change, and select "Propose file change" 9. Select "Create pull request" on the 'Open a pull request' page 10. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link within the PR 11. Once you're satisfied with your changes, either type #sign-off in the comments to start the merge/publish process, or if you'd like me to review the changes, then type @MashaMSFT in the comments so that I can review and sign off If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs organization. More information can be found here: https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master Feel free to ping me directly on Teams or over email if you run into any issues.

KEYWORD	OUTPUT

Place the following script in your .ahk file to enable this keyword script - modify the underscore to your GitHub alias:

```
:r0:sqlprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following:

1. Navigate to the page of interest
2. Select the "Edit" button at the top right
3. Modify the URL so that 'sql-docs' becomes 'sql-docs-pr' and 'live' becomes 'master'
5. Press enter to navigate to the newly-changed URL and reload the page
6. Select the pencil to edit the document
7. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
8. Scroll down to the bottom, name your file change, and select "Propose file change"
9. Select "Create pull request" on the 'Open a pull request' page
10. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link within the PR
11. Once you're satisfied with your changes, either type #sign-off in the comments to start the merge/publish process, or if you'd like me to review the changes, then type @______ in the comments so that I can review and sign off

If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs organization. More information can be found here:
https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master

Feel free to ping me directly on Teams or over email if you run into any issues.
)
```

Editing a doc in a public repo

KEYWORD	OUTPUT
pubprpr	It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following: 1. Navigate to the page of interest 2. Select the "Edit" button at the top right of the page 3. Select the "Pencil" icon on the right 4. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet 5. Name your file change and provide a description, if necessary 6. Select "Propose file change" 7. Select "Create pull request" on the 'Comparing changes' page 8. Select "Create pull request" on the 'Open a pull request' page 9. Feel free to type @MashaMSFT in the comments so I can review the change as well if I'm not the author of the page Feel free to ping me directly on Teams or over email if you run into any issues.

Place the following script in your .ahk file to enable this keyword script - modify the underscore to your GitHub alias:

```
:r0:pubprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the following:

1. Navigate to the page of interest
2. Select the "Edit" button at the top right of the page
3. Select the "Pencil" icon on the right
4. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet
5. Name your file change and provide a description, if necessary
6. Select "Propose file change"
7. Select "Create pull request" on the 'Comparing changes' page
8. Select "Create pull request" on the 'Open a pull request' page
9. Feel free to type @_____  in the comments so I can review the change as well if I'm not the author of the page

Feel free to ping me directly on Teams or over email if you run into any issues.
)
```

Miscellaneous templates

The current date

KEYWORD	OUTPUT
datep0	05/01

Place the following script in your .ahk file to enable this keyword script:

```
:r0:datep0::
  Date := A_Now
  FormatTime, nDate, %Date%, MM/dd
  Send, %nDate%
Return
```

The date two days from now

KEYWORD	OUTPUT
datep2	05/03

Place the following script in your .ahk file to enable this keyword script:

```
:r0:datep2::
  Date := A_Now
  Date += 2, Days
  FormatTime, nDate, %Date%, MM/dd
  Send, %nDate%
Return
```

FAQ

How do I modify keywords?

The keyword is the second word between the colons. So in `:r0:datep0::`, the keyword is datep0. If you want to modify the keywords, just modify the text between the colons for the script you're interested in changing.

How are scripts managed?

All the scripts are located within your .ahk file, in text format. To review or edit your existing scripts, or add new scripts, simply open the .ahk file, make your changes, save the .ahk file, and then double-click it to initiate the script to enforce your new changes. If your .ahk file lives in your startup folder and you made your changes elsewhere, you'll need to replace the existing .ahk file in your startup folder with the one you just changed.

Do I need to recreate the scripts on every machine?

Theoretically, yes. But in practice, keep the same .ahk file on your Onedrive so it's the same file on every machine, and then just place this file in the startup folder of each machine. If you need to make changes, make changes to the file on Onedrive, and then replace the existing file in the startup folder. Autohotkey does need to be installed manually on each machine.

Is it possible to make the scripts more complex?

Yes. Check out the scripting showcase for more information: https://autohotkey.com/docs/scripts/.

Full script

Instead of copying each individual script snippet, you can also bulk copy the entire script and save it in your `.ahk` file.

Place the following script in your .ahk file to enable all the keywords found within this topic. Don't forget to modify the underscores to relevant values.

```
:r0:vnotee::
(
Status
======

To do
======

Done
====

Git
====

Contacts
========

Details
=======
)

:r0:dnotee::

FormatTime, Time,, yyyy-MM-dd
Send %Time%
send {Space}| Sprint
return

:r0:rnotee::
(
====================================================
                    To do
```

```
====================================================

Tasks
------
1.
2.
3.


Public PRs
-----------
1.
2.
3.


Git Issues
----------
1.
2.
3.


====================================================
                   Done
====================================================

Tasks
------

Public PRs
-----------

Git Issues
----------
)


:r0:ishnew::
(
Thanks for contacting Microsoft with feedback about our product documentation. Depending on the complexity, it could take from a few business days to a few weeks to look further into your question, issue, or suggestion. We'll contact you if we need further information, and you'll receive an email message each time we "@mention" you in a comment.

If you're suggesting a simple change to the documentation, it might be faster to submit the suggestion yourself by creating a GitHub pull request (PR). Here's how:

1. On the page you have feedback about, select the **Edit** link at the top right.
1. On the next page, select the **Pencil** icon at the top right.
1. On the next page, in the **Edit file** text window, make your edits directly to the text you want to change.
    If you need help with formatting the new or changed text, see our [Markdown Cheatsheet](https://github.com/adam-p/markdown-here/wiki/Markdown-Cheatsheet).
1. After you've made your edits, under **Commit changes**:
    a. In the first text box, enter a brief description of the change you've made.
    b. In the **Add an optional extended description** box, paste the link to your GitHub issue.
1. Select **Propose file change**.
1. On the **Comparing changes** page, select **Create pull request**.
1. On the **Open a pull request** page, select **Create pull request**.

Allow a day or more for the article's author to review and approve your change or offer an alternative solution.

By opening a pull request, you're helping the author add your approved change directly to the documentation. After the updated article is published, GitHub lists you as an article contributor.

If your issue is more complex than a simple change in the documentation, that's okay.  We'll continue to research your suggestion and then get back to you with our response.
```

```
research your suggestion and then get back to you with our response.

Thanks again for submitting your feedback. Your suggestions help improve our documentation.
)

:r0:ishport::
(
Thanks for contacting Microsoft about your issue. As we understand it, you're describing a problem with the
product itself and not with the product documentation.

To get assistance or support for your product issue, we recommend that you engage with the product community
or open a ticket with Microsoft Support. For more information, see __________________.

To provide product suggestions or ideas for improvement, go to ___________.

If you're submitting feedback about the product documentation, please reply to this comment to clarify your
issue further. Otherwise, we'll proceed with closing out this Git issue within a few business days.

We're sorry that the _________ Docs team was unable to assist you further with your issue, but we appreciate
your reaching out to us. If you experience other issues in the future, we encourage you to contact us again.
)

:r0:ishclose::
(
Thanks again for contacting Microsoft. We've submitted a documentation change based on your suggestion, and
the updated article should be live by tomorrow. Please let us know if there’s anything more we can do for
you. To follow up on this issue, leave us a comment with an @mention.
)

:r0:azprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the
following:

1. Navigate to the page of interest
2. Select  "Edit" button at the top right
3. Modify the URL so that 'azure-docs' becomes 'azure-docs-pr'
5. Press enter to navigate to the newly-changed URL and reload the page
6. Select the pencil to edit the document
7. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-
here/wiki/Markdown-Cheatsheet
8. Scroll down to the bottom, name your file change, and select "Propose file change"
9. Select "Create pull request" on the 'Open a pull request' page
10. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View"
link within the PR
11. Once you're satisfied with your changes, type @_________ in the comments so that I can review and sign
off to approve the changes

If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs
organization. More information can be found here:
https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master

Feel free to ping me directly on Teams or over email if you run into any issues.
)

:r0:sqlprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the
following:

1. Navigate to the page of interest
2. Select the "Edit" button at the top right
3. Modify the URL so that 'sql-docs' becomes 'sql-docs-pr' and 'live' becomes 'master'
4. Press enter to navigate to the newly-changed URL and reload the page
5. Select the pencil to edit the document
6. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-
here/wiki/Markdown-Cheatsheet
7. Scroll down to the bottom, name your file change, and select "Propose file change"
8. Select "Create pull request" on the 'Open a pull request' page
9. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link
```

```
9. Once the build finishes (should take about 5-10 minutes), review your changes by clicking the "View" link
within the PR
10. Once you're satisfied with your changes, type @________ in the comments so that I can review and sign
off to approve the changes

If you're getting a 404, you may need to link your Github account to Microsoft, and join the MicrosoftDocs
organization. More information can be found here:
https://review.docs.microsoft.com/help/contribute/contribute-get-started-setup-github?branch=master

Feel free to ping me directly on Teams or over email if you run into any issues.
)


:r0:pubprpr::
(
It may be faster for you to edit the content yourself via a Pull Request (PR). To do so, please do the
following:

1. Navigate to the page of interest
2. Select the "Edit" button at the top right of the page
3. Select the "Pencil" icon on the right
4. Modify the text within the text box in markdown format - more info: https://github.com/adam-p/markdown-
here/wiki/Markdown-Cheatsheet
5. Name your file change and provide a description, if necessary
6. Select "Propose file change"
7. Select "Create pull request" on the 'Comparing changes' page
8. Select "Create pull request" on the 'Open a pull request' page
9. Feel free to type @_____  in the comments so I can review the change as well, if I'm not the author of
the page

Feel free to ping me directly on Teams or over email if you run into any issues.
)

:r0:datep0::
  Date := A_Now
  FormatTime, nDate, %Date%, MM/dd
  Send, %nDate%
Return

:r0:datep2::
  Date := A_Now
  Date += 2, Days
  FormatTime, nDate, %Date%, MM/dd
  Send, %nDate%
Return
```

Twitter App (aka.ms/tweet) (Preview)

4/16/2021 • 6 minutes to read

The Twitter App allows for the simplified use and management of public-facing Twitter accounts. Once a Twitter account is added to the app by the account manager, anyone from the company can propose tweets. Doing so adds the tweet to the account manager's **Tweet queue**, letting the account manager decide which tweets publish to Twitter.

This app allows multiple people to tweet from the same Twitter account without sharing credentials, but still controls all tweets in a single person's hands. You can use this app for your own personal Twitter account or use it for larger accounts, such as @docsmsft or @SQLDocs.

> **IMPORTANT**
>
> The strength of this app is its simplicity, along with both Microsoft and DevRel-specific feature integration. However, Twitter accounts with more than 1000 followers should also be using Sprinklr, a broader and more complex feature set.

First-time setup

Add a Twitter account to allow other users to tweet to it, pending your approval. Nothing is posted to this account without your consent. You can remove the account at any time either in the app or via your Twitter third-party application permissions.

The account that you're currently signed into Twitter with is the account added to the app. If you're already logged into an account, but want to set up a different one, then you need to go to Twitter, sign out of the current account, and log into the account you want to add.

Follow these steps to add a Twitter account to the Twitter app:

1. Navigate to the **Management portal** from aka.ms/tweet after signing in. You can also select **Account** from the top navigation menu.
2. Select the **Sign-in with Twitter** button and follow the screens to allow the TwitterApp to access your Twitter account.
3. Select **Return to portal** once your account has been added successfully.

> **NOTE**
>
> The account that you're currently signed into Twitter with is the one that gets added to the portal. If you need to add a different account, be sure to sign directly into Twitter, sign out of the one account, and then sign into the account you want to add to the app.

Account Management

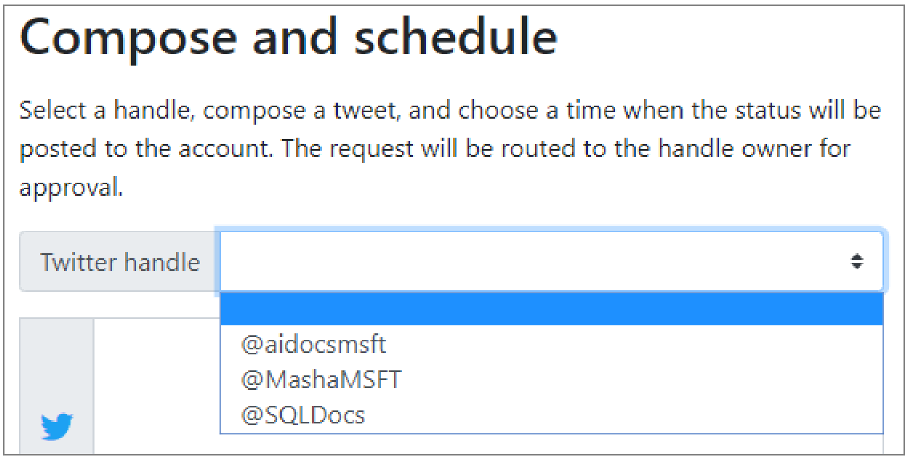
As a Twitter account manager, you first need to configure your Twitter account for first use. After that, you can approve, edit, and delete tweets using the tweet queue.

To manage your twitter account, expand **Settings** from the **Management portal**.

Here you can set your account to be either public or private, and enable automatic retweets of mentions.

Privacy

Setting your account to private means that only you can tweet to that Twitter handle, but setting it to public means that anyone can tweet to the private handle from within the app. Only your own private account and accounts set to public are visible in the **Twitter handle** drop-down in the Tweet portal:



How to compose a tweet

Follow the steps below to tweet to any available account:

1. Sign in to your Twitter account.
2. Navigate to the **Tweet portal** from aka.ms/tweet.
3. Select the Twitter account you want to use from the drop-down.
4. Select a date and time you like your tweet to go live (optional).
5. Compose a tweet and then select **Create Tweet** to add it to the account manager's tweet queue.

> **NOTE**
>
> The character limit matches that of Twitter. You have 280 characters per Tweet. URL character counts are calculated at a constant value of 23 characters per URL regardless of the URL length. This means that a URL of 500 characters long is perfectly valid in a Tweet, as it only counts as 23 characters. You may also have multiple URLs in a Tweet. If you have two URLs with a single space between them, your Tweet would be 47 characters long.

Tweet guidelines

Add a URL in a tweet

If you are adding a URL to your tweet, then please follow the guidelines listed below.

1. Remove the locale (ex: en-us) from the URL.
2. Add one of the following dashboard extensions to the URL. The dashboard extensions are used to help collect data.
   a. For @docsmsft add: ?WT.mc_id=docsmsft-twitter
   b. For @WindowsDocs add: ?WT.mc_id=windowsdocs-twitter

For example: https://docs.microsoft.com/rest/api/?WT.mc_id=windowsdocs-twitter

Evangelize your tweets

- Tag any appropriate PM or team that could help amplify the tweet.
- Use relevant hashtags.
- Include an image if possible.

Ideas to tweet about and some sample tweet formats:

Some sample ideas to tweet.

- New or updated topic on cool tech.
- An interesting blog post.

- Quote tweeting an existing tweet.
- Poll about what more info users want.
- An interesting piece of history about our technology.
- A question about how devs are using technology.
- Ideas on how to use UI design.
- Tiny 4-6 step tutorials in a series of tweets.

Here are some sample tweets.

- "Did you know you can do a cool thing with X? Here's how..."
- "Great news! X just got updated - now it has Y, and here's a list of all the other new stuff."
- "Have you ever found yourself wishing for X? Lucky you - we just released X. Here's more information."
- "This is how to do X, which is a cool thing, in four steps."
- "What do you think of X? How could we improve it?"

For more information about tweet submissions, visit SocAmp Submssions Onenote

For more information about guidlines, visit Social Media Guidlines Onenote.

Setting a time to send a tweet

If you leave the date blank, the tweet goes live as soon as the account manager approves it.

If you provide a date and time, the tweet goes live at the designated time frame, as long as it's been approved by the account manager.

The account manager for a specific handle can edit your tweet.

If you @mention another account that's been added to the app (such as @sqldocs), and that account has the retweet mentions feature enabled, your tweet is added as a retweet to that account managers queue. The account manager can't edit it.

Retweet mentions

Twitter gives you the ability to mention another Twitter handle, such as @sqldocs. If you enable automatic retweet of mentions, then anytime *anyone* mentions your twitter account (either through the app or through Twitter), that tweet is added to your queue as a retweet. Since this functionality uses Twitter's underlying retweet API, these tweets are uneditable. When you approve the tweet, it shows up as a retweet on your Twitter page and displays the original user's Tweet.

For example, this screenshot shows the @sqldocs account retweeting @MashaMSFT's tweet, who mentioned @sqldocs in the tweet:

The original tweet and the retweet were done through the Twitter App.

Tweet queue

The queue in the Management portal is where the Twitter account manager can approve, edit, or delete tweets. Select **Tweet Queue** from the top navigation menu to access the queue for all of the Twitter accounts associated with your TwitterApp account.

Remarks

- You can approve a tweet by selecting the **Approve** button.
- You can edit the content of a tweet by selecting the **Edit** button.
- Once a tweet is approved, it goes live immediately or at its scheduled time.
- Use the trash can to delete a tweet, either before or after it's been approved.
- You can't currently modify the date and time of a scheduled tweet.
- You can't edit the content of a retweet.

Sprinklr

Sprinklr is the platform of record for managing Microsoft's social media accounts and is required for all publishing and engagement for any social account with more than 1,000 fans/followers.

Sprinklr has a robust feature set and is geared towards social media interaction and engagement. You can schedule tweets, follow trends, interact with customers, and run comprehensive social media presence analysis.

Although Sprinklr allows multiple people to tweet to the same account, it doesn't have the same unique request/approval process as the Twitter app. Additionally, there are plans to integrate GitHub functionality into the Twitter app, such as automatically tweeting when adding new articles and other DevRel specific things. However, the TwitterApp and Sprinklr aren't mutually exclusive and can be used in conjunction, with Sprinklr bringing added complex functionality.

Contact

You can contact the SocAmp V-team team for questions and comments.

- Twitter App info
- Feedback / feature request.
- See what's currently in development via the DevOps dashboard.

- Email: TwitterApp@microsoft.com

Next steps

- Submit feedback and feature updates.

Banners on docs.microsoft.com

5/21/2021 • 4 minutes to read

Banners are used on `docs.microsoft.com` to promote events or announcements. They're limited to a specific set of documentation related to the notice or promotion, with the exception being large-scale events like Ignite and Build. These banners have been proven to be successful in reaching Docs users.

This article details the guidelines for when a banner may be used and instructions for requesting a new banner. Requests are approved and processed by the PM team, and then the Dev team creates the banner.

Primary uses and scenarios

Banners are run at the discretion of the Docs product team, and every request will require approval. These situations and events are the primary uses for banners on `docs.microsoft.com`:

- Emergency situations (for example, Security issues)
- Tier 1 events (for example, Build or Ignite)
- Marquee/Learn TV events (for example, events run by DevRel or cloud advocates)

How to fill out the banner request form

To submit a request, fill out the banner request form at aka.ms/BannerRequest. Read through the instructions on the form carefully and be as specific as you can when filling out the form.

> **IMPORTANT**
>
> The minimum lead time for banners is 10 days prior to the desired go-live date. Requests must be submitted at least 10 days before the banner is expected to go live. This gives the PM team the necessary time to review the request and pass it on to the Dev team, if approved.

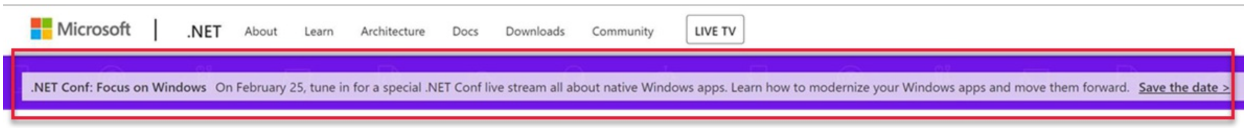
You'll need the following information to fill out the form:

- Name of requester.
- Reason for banner.
- Banner information:
  - Date/time: Provide the date/time that the banner should go live and the date/time it should end.
  - Copy title: Provide the title of the banner.
  - Copy description: Provide the text of the banner.
  - CTA text: At the end of the banner text, there is a CTA (Call To Action) link for users to select. In this field, provide the text of the link.
  - CTA destination URL: Provide the URL that the link should go to.
- All Docs/Learn URL paths where the banner should be displayed.
- Any additional questions, comments, or information that you want to communicate.

After you've finished filling out the form, click **submit**. The form will send an email to the PM team in charge of banners.

Example banner and form entries

An example of a typical Learn TV event banner and the banner information that was submitted for this request:



- **Copy title:** .NET Conf: Focus on Windows
- **Copy Description:** On February 25, tune in for a special .NET Conf live stream all about native Windows apps. Learn how to modernize your Windows apps and move them forward.
- **CTA text:** Save the date
- **CTA destination URL:** aka.ms/TrackingLinkExample

Timing of banners

Allow two full business days after a request is submitted to hear back from the PM team. If the request was received before the required 10-day lead time, the Dev team will meet that deadline.

Depending on the type of promotion, and at the discretion of the product team, a banner will typically be live on the site for 2-4 weeks.

> **NOTE**
>
> If the request is submitted after the 10-day lead time of when the banner will go live, please submit the form and contact Isa Hoban (ihoban) immediately. The PM team will do their best to meet the deadline, but cannot guarantee the request will be fulfilled if the form is not submitted 10 days prior to the desired go-live date.

Evaluating banner performance

For banners with a CTA (Call to Action), use a specific tracking URL so you can observe how the links from `docs.microsoft.com` performed compared to other referrers to the destination page. You'll share this information with the PM team after the banner is removed and be tagged in an Azure DevOps work item, along with instructions for the data you're expected to report.

FAQ

Can I include images?

No. Specialized banners are currently reserved for Tier 1 events. Otherwise, expect a single-colored banner with black text.

Can I use a banner to promote on-demand content following an event?

No. We don't approve the use of banners for promoting on-demand content following an event. The team is beginning discovery on potential new features for notifications and promotions within a docset, outside of banner usage, and these may be a more appropriate option for promoting on-demand content in the future.

Are banners persistent or dismissible? Will a banner reappear after being dismissed?

Banners for events must be dismissible. If a user selects the banner close button, the banner will be hidden, and the user action is stored in browser local storage. This means that the banner won't reappear unless a user removes the key "dismissed-banner" from local storage.

Are banners localized?

Yes, if needed, the banners will be localized.

Can I run a second banner on a page that already has a banner?

No. Two event banners are never run at the same time. There are some cases where an event banner may appear with a security/emergency banner, but these will be decided and prioritized at the discretion of the PM team.

Questions and feedback

Contact Isa Hoban (ihoban) for further questions or concerns.

Docs Authoring Pack for VS Code

4/8/2021 • 7 minutes to read

The Docs Authoring Pack is a collection of VS Code extensions to aid with Markdown authoring for docs.microsoft.com. The pack is available in the VS Code Marketplace and contains the following extensions:

- Docs Markdown: Provides Markdown authoring assistance for docs.microsoft.com (Docs) content, including basic Markdown support and support for custom Markdown syntax in Docs, such as alerts, code snippets, and non-localizable text. Now also includes some basic YAML authoring assistance, such as inserting TOC entries.
- markdownlint: A popular Markdown linter by David Anson to help make sure your Markdown is valid.
- Code Spell Checker: A fully offline spell checker by Street Side Software.
- Docs Preview: Uses the docs.microsoft.com CSS for more accurate Markdown preview, including custom Markdown.
- Docs Article Templates: Allows users to scaffold Learn modules and apply Markdown skeleton content to new files.
- Docs YAML: Provides Docs YAML schema validation and auto-complete.
- Docs Images: Provides image compression and resizing for folders and individual files to help authors of docs.microsoft.com.

Prerequisites and assumptions

To insert relative links, images, and other embedded content with the Docs Markdown extension, you must have your VS Code workspace scoped to the root of your cloned Open Publishing System (OPS) repo. For example, if you have cloned the docs repository to `C:\git\SomeDocsRepo\`, then open that folder or a subfolder in VS Code: **File** > **Open Folder** menu, or `code C:\git\SomeDocsRepo\` from the command line.

Some syntax supported by the extension, such as alerts and snippets, are custom Markdown for OPS. Custom Markdown will not render correctly unless published via OPS.

How to use the Docs Markdown extension

To access the **Docs Markdown** menu, type ALT+M. You can click or use the up and down arrows to select the command you want. Or you can type to start filtering, then hit ENTER when the function you want is highlighted in the menu.

See the Docs Markdown readme for an up-to-date list of commands.

How to generate a master redirect file

The Docs Markdown extension includes a script to generate or update a master redirection file for a repo, based on the `redirect_url` metadata in individual files. This script checks every Markdown file in the repo for `redirect_url`, adds the redirection metadata to the master redirection file (*.openpublishing.redirection.json*) for the repo, and moves the redirected files to a folder outside the repo. To run the script:

1. Select F1 to open the VS Code command palette.
2. Start typing "Docs: Generate..."
3. Select the command `Docs: Generate master redirection file`.
4. When the script finishes running, the redirection results will show in the VS Code output pane, and the

removed Markdown files will be added to the Docs Authoring\redirects folder under your default path.

5. Review the results. If they are as expected, submit a pull request to update the repo.

How to assign keyboard shortcuts

1. Type `CTRL+K` then `Ctrl+S` to open the **Keyboard Shortcuts** list.
2. Search for the command, such as `formatBold`, for which you want to create a custom key binding.
3. Click the plus that appears near the command name when you mouse over the line.
4. After a new input box is visible, type the keyboard shortcut you want to bind to that particular command. For example, to use the common shortcut for bold, type `Ctrl+B`.
5. It's a good idea to insert a `when` clause into your key binding, so it won't be available in files other than Markdown. To do this, open *keybindings.json* and insert the following line below the command name (be sure to add a comma between lines):

   ```
   "when": "editorTextFocus && editorLangId == 'markdown'"
   ```

   Your completed custom key binding should look like this in *keybindings.json*:

   ```
   [
       {
           "key": "ctrl+b",
           "command": "formatBold",
           "when": "editorTextFocus && editorLangId == 'markdown'"
       }
   ]
   ```

   > **TIP**
   >
   > Place your key bindings in this file to overwrite the defaults

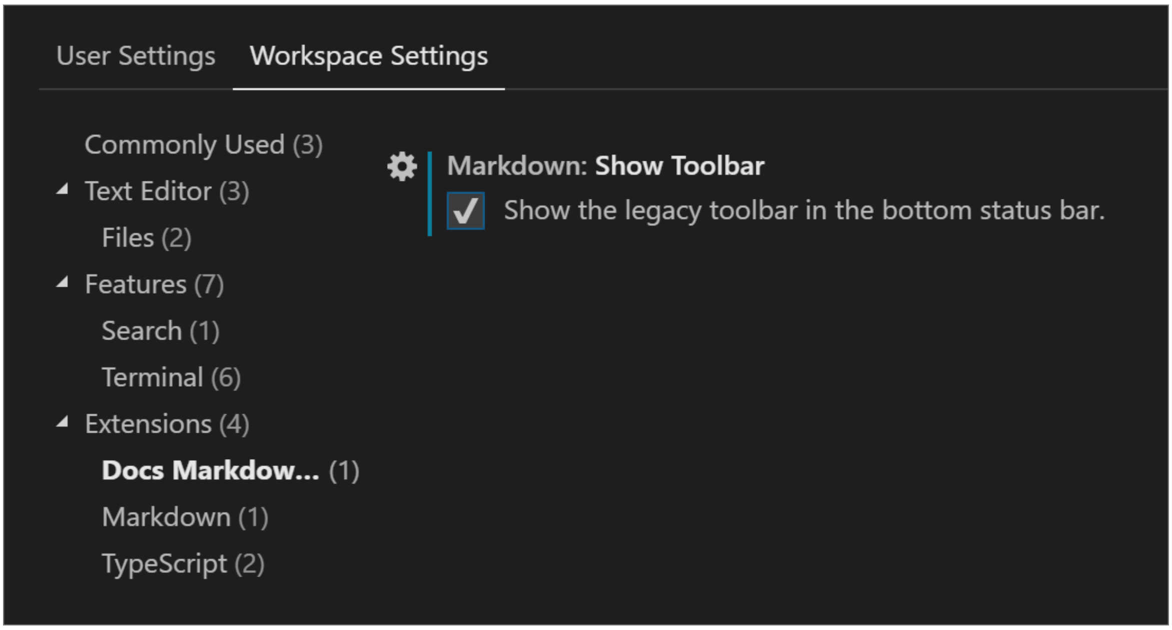
6. Save *keybindings.json*.

For more information on key bindings, see the VS Code docs.

How to show the legacy "Gauntlet" toolbar

Former users of the extension code-named "Gauntlet" will notice that the authoring toolbar no longer appears at the bottom of the VS Code window when the Docs Markdown Extension is installed. This is because the toolbar took up a large space on the VS Code status bar and did not follow best practices for extension UX, so it is deprecated in the new extension. However, you can optionally show the toolbar by updating your VS Code settings.json file as follows:

1. In VS Code, go to **File** > **Preferences** > **Settings** or select `Ctrl+,`.
2. Select **User Settings** to change the settings for all VS Code workspaces or **Workspace Settings** to change them for just the current workspace.
3. Select **Extensions** > **Docs Markdown Extension Configuration**, and then select **Show the legacy toolbar in the bottom status bar**.



Once you've made your selection, VS Code updates the *settings.json* file. You will then be prompted to reload the window for the changes to take effect.

Newer commands added to the extension will not be available from the toolbar.

How to use Docs templates

The Docs Article Templates extension lets writers in VS Code pull a Markdown template from a centralized store and apply it to a file. Templates can help ensure that required metadata is included in articles, that content standards are followed, and so on. Templates are managed as Markdown files in a public GitHub repository.

> **TIP**
> The core Docs templates described in the Docs template finder article are available in the Docs Article Templates extension.

To apply a template in VS Code

1. Ensure the Docs Article Templates extension is installed and enabled.
2. If you don't have the Docs Markdown extension installed, click `F1` to open the command palette, start typing "template" to filter, then click `Docs: Template`. If you do have Docs Markdown installed, you can use either the command palette or click `Alt+M` to bring up the Docs Markdown QuickPick menu, then select `Template` from the list.
3. Select the desired template from the list that appears.

To add your GitHub ID and/or Microsoft alias to your VS Code settings

The Templates extension supports three dynamic metadata fields: author, ms.author, and ms.date. That means that if a template creator uses these fields in the metadata header of a Markdown template, they will be auto-populated in your file when you apply the template, as follows:

METADATA FIELD	VALUE
`author`	Your GitHub alias, if specified in your VS Code settings file.
`ms.author`	Your Microsoft alias, if specified in your VS Code settings file. If you are not a Microsoft employee, leave unspecified.

METADATA FIELD	VALUE
`ms.date`	The current date in the Docs-supported format, `MM/DD/YYYY`. The date is not automatically updated if you subsequently update the file - you must update it manually. This field is used to indicate the "article freshness".

To set author and/or ms.author

1. In VS Code, go to **File** > **Preferences** > **Settings** or select `Ctrl+,`.
2. Select **User** settings to change the settings for all VS Code workspaces, or **Workspace** settings to change them for just the current workspace.
3. In the Default Settings pane on the left, find **Docs Article Templates Extension Configuration**, click the pencil icon next to the desired setting, then click Replace in Settings.
4. The **User** settings pane will open side by side, with a new entry at the bottom.
5. Add your GitHub ID or Microsoft email alias, as appropriate, and save the file.
6. You might need to close and restart VS Code for the changes to take effect.
7. Now, when you apply a template that uses dynamic fields, your GitHub ID and/or Microsoft alias will be auto-populated in the metadata header.

To make a new template available in VS Code

1. Draft your template as a Markdown file.
2. Submit a pull request to the templates folder of the MicrosoftDocs/content-templates repo.

The docs.microsoft.com team will review your template and merge the PR if it meets docs.microsoft.com style guidelines. Once merged, the template will be available to all users of the Docs Article Templates extension.

Demo several features

Here's a short video that demonstrates the following features of the Docs Authoring Pack:

- **YAML files**
  - Support for "Docs: Link to file in repo"
- **Markdown files**
  - Update “ms.date" Metadata Value context menu option
  - Code auto-completion support for code-fence language identifiers
  - Unrecognized code-fence language identifier warnings / auto correction support
  - Sort selection ascending (A to Z)
  - Sort selection descending (Z to A)

Contribution expectations

The Docs Authoring Pack extension is open source and available for contributions to anyone with a GitHub account. There is a dedicated internal Microsoft team that actively works on this project. This team monitors issues and pull requests. The service level agreement (SLA) and expectation of getting a pull request reviewed is currently one week. The team is undergoing automation efforts to improve this turn around time.

Next steps

Explore the various features available in the Docs Authoring Pack, Visual Studio Code extension.

- Dev lang completion

- Image compression
- Metadata updates
- Reformat table
- Smart quote replacement
- Sort redirects
- Sort selection

Dev lang completion

3/3/2020 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

Contributors need assistance determining the valid language identifiers (dev langs) that can follow triple-backticks (code fence openings) in a Markdown file. Unfortunately, build-time validation of dev langs doesn't exist. The result is disparate representations of a single language within the same conceptual docset.

Consider C# as an example. Contributors have used `c#`, `C#`, `cs`, `csharp`, and others as dev lang representations of the language. Which of the preceding representations is correct?

The *Dev lang completion* feature dispels the confusion by displaying a list of known dev langs. Upon selecting a dev lang name from IntelliSense:

- The code fence is closed.
- The caret is positioned in the code fence.

Preferences

It's not possible to disable this feature. The following settings are available:

- Display commonly used dev langs
- Display all known dev langs

Display commonly used dev langs

Only a subset of the valid dev langs will be used in a single docset. To enhance the user experience:

1. In Visual Studio Code, open the docset to the root directory.
2. Select **File** > **Preferences** > **Settings** and filter by *Docs Markdown Extension*.
3. Click the **Edit in settings.json** link in the **Markdown: Docset Languages** section.
4. Add the following `markdown.docsetLanguages` property to the *settings.json* file:

   ```
   {
       "markdown.docsetLanguages": [

       ]
   }
   ```

5. Position your caret in the property's empty array, and activate IntelliSense (via `Ctrl` + `Space`). A list of known dev lang names appears.
6. Add dev lang names to the array until you're satisfied with the list. For example, the following list will display four dev lang names to the user after typing triple-ticks:

```
{
    "markdown.docsetLanguages": [
        ".NET Core CLI",
        "C#",
        "Markdown",
        "YAML"
    ]
}
```

7. Save your changes to the *settings.json* file.

> **WARNING**
> An empty `markdown.docsetLanguages` array causes all known dev langs to display.

Display all known dev langs

By default, all known dev lang names are displayed in IntelliSense. This setting overrides the `markdown.docsetLanguages` property described in Display commonly used dev langs.

To change this setting:

1. Select **File** > **Preferences** > **Settings** and filter by *Docs Markdown Extension*.
2. Toggle the setting in the **Markdown: All Available Languages** section.

In action

Below is a brief demonstration of this feature:

Image compression

3/5/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, Visual Studio Code meta extension is comprised of multiple sub extensions. This feature is included in the Docs Images extension. The Docs Markdown extension is part of the Docs Authoring Pack, there is no need to install it separately.

Summary

All documentation is provided via the web, with the exception of PDF versions of docs. When serving static content, it is best to minimize the number of bytes sent over the wire. One way to do that is to compress images at rest.

The Docs Authoring Pack extension includes image compression context menu items. The following image types / extensions are supported:

- **.png*
- **.jpg*
- **.jpeg*
- **.gif*
- **.svg*
- **.webp*

The lossless image compression algorithms are used, where applicable.

Compress image

From the **Explorer** navigation pane, right-click on an image file - then select the **Compress image** option. The image is then compressed.

Compress images in folder

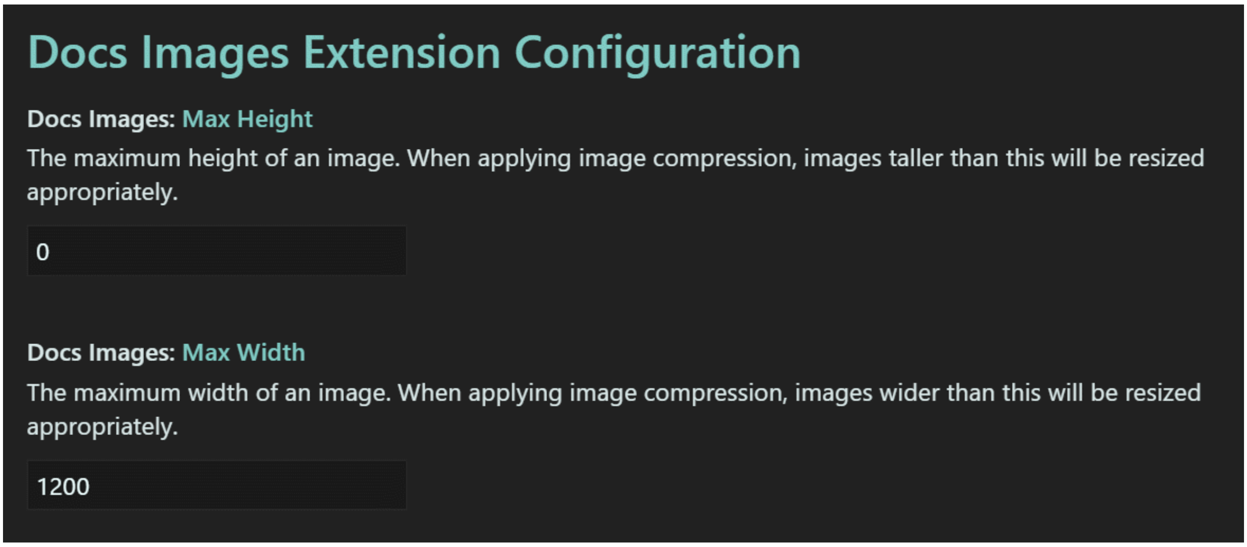
From the **Explorer** navigation pane, right-click on a folder containing images - then select the **Compress images in folder** option. All images in the folder are compressed.

Considerations

Large resolution images are implicitly resized. The maximum dimensions are based on the platform suggested max width, suggested here. The max is only used when images are larger than they are recommended to be, and they will maintain the aspect ratio when automatically resized.

Preferences

The maximum dimensions are configurable, but a default max width of `1200` pixels exists. To configure the max dimensions, select **File -> Preferences -> Settings** and filter by `"Docs Image Extension"`.

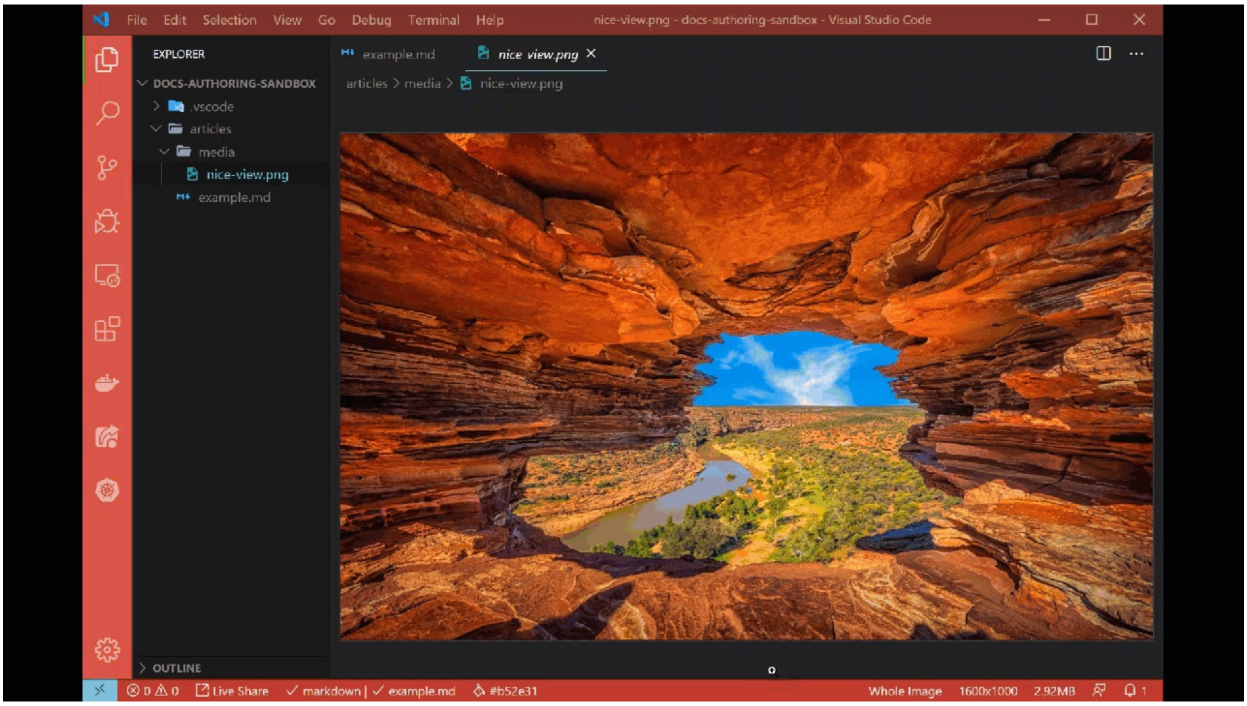


NOTE

A value of `0` in either the **Max Width** or **Max Height** will simply ignore resolution variances.

In action

Below is a brief demonstration of this feature.



Metadata explorer

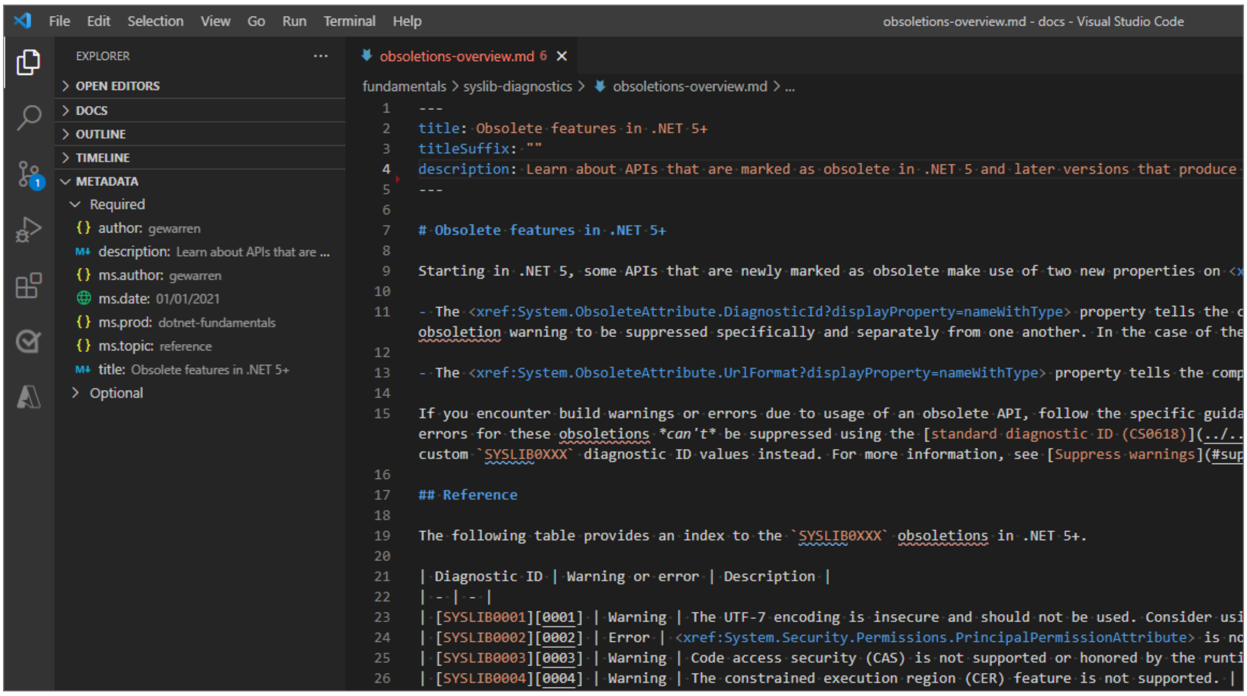
6/11/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

The Metadata explorer appears automatically in the Explorer side bar of Visual Studio Code when you open a Docs Markdown file. It has two main sections: required metadata and optional metadata.



Metadata sources

The Metadata explorer shows metadata from three sources: the YAML front matter of the Markdown file that's open in the editor, and the `fileMetadata` and `globalMetadata` sections of the docset's *docfx.json* file. Icons help to quickly identify the source of the displayed metadata. If your article is missing any required metadata, that metadata key appears with a warning icon.

ICON	METADATA SOURCE
	YAML front matter of active Markdown file
	`fileMetadata` section of *docfx.json* file
	`globalMetadata` section of *docfx.json* file
	This required metadata is *missing completely*

Precedence

The Metadata explorer only shows unique metadata keys. If more than one value can apply for the same key, the explorer uses the same precedence settings as DocFx does. The order of precedence in decreasing priority is:

1. YAML front matter
2. `fileMetadata`
3. `globalMetadata`

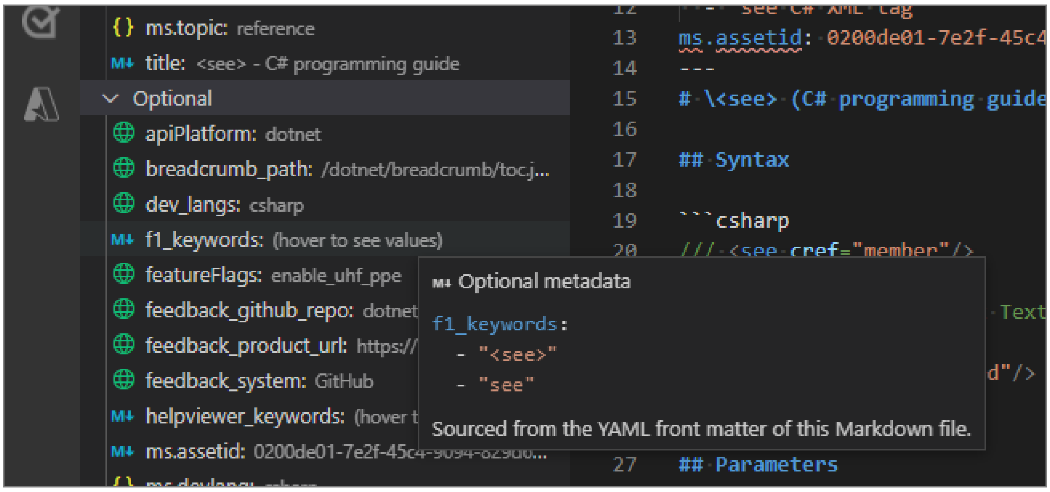
If multiple folder-based glob patterns apply to a single article, the entry that appears last in the *docfx.json* file is selected. For example, consider the file at *C:/docs/csharp/reference/keywords.md* and the following entries in the *docfx.json* file:

```
"ms.topic": {
  "csharp/**/*.md": "conceptual",
  "csharp/reference/*.md": "language-reference"
}
```

In this case, while both glob patterns include the path to the file, the second value of `language-reference` is selected.

Tooltip

You can hover over a value to see an expanded display in the tooltip. This is especially useful for keys that have multiple values in an array. The information also includes the source of the metadata.



Refresh

If you have auto-save disabled in Visual Studio Code, the Metadata explorer refreshes automatically when you save your Markdown file. If you have auto-save enabled, you can refresh the displayed metadata by clicking the **Refresh** button at the top of the explorer.

When you open a different Markdown file, the explorer refreshes automatically.

See also

- How to apply metadata
- Metadata attributes

Update metadata

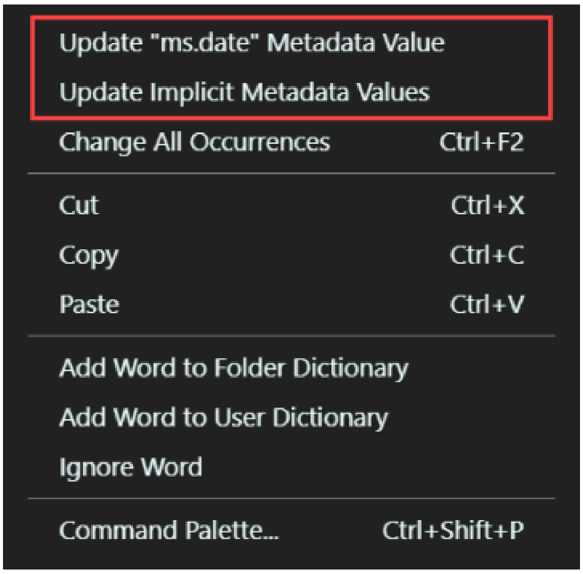
11/2/2020 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

In a Markdown (**.md*) file, there are two contextual menu items specific to metadata. When you right-click anywhere in the text editor, you will see something similar to the following menu items:



Update `ms.date` metadata value

Selecting the **Update** `ms.date` **Metadata Value** option will set the current Markdown files `ms.date` value to today's date. If the document does not have an `ms.date` metadata field, no action is taken.

Update implicit metadata values

Selecting the **Update implicit metadata values** option will find and replace all possible metadata values that could be implicitly specified. Metadata values are implicitly specified in the *docfx.json* file, under the `build/fileMetadata` node. Each key value pair in the `fileMetadata` node represents metadata defaults. For example, a Markdown file in the *top-level/sub-folder* directory that omits the `ms.author` metadata value could implicitly specify a default value to use in the `fileMetadata` node.

```json
{
    "build": {
        "fileMetadata": {
            "ms.author": {
                "top-level/sub-folder/**/**.md": "dapine"
            }
        }
    }
}
```

In this case, all Markdown files would implicitly take on the `ms.author: dapine` metadata value. The feature acts on these implicit settings found in the *docfx.json* file. If a Markdown file contains metadata with values that are explicitly set to something other than the implicit values, they are overridden.

Consider the following Markdown file metadata, where this Markdown file resides in **top-level/sub-folder/includes/example.md**:

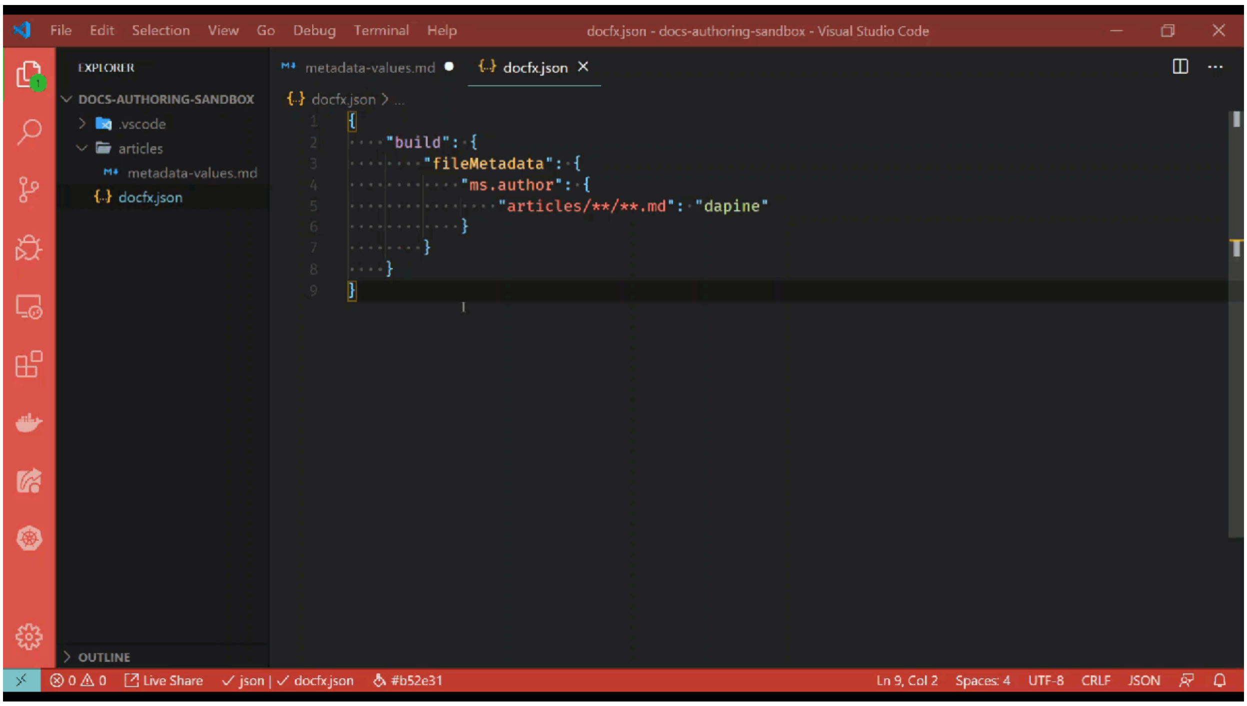
```markdown
---
ms.author: someone-else
---

# Content
```

If the **Update implicit metadata values** option was executed on this file, with the assumed *docfx.json* content from above the metadata value would be updated to `ms.author: dapine`.

```markdown
---
ms.author: dapine
---

# Content
```

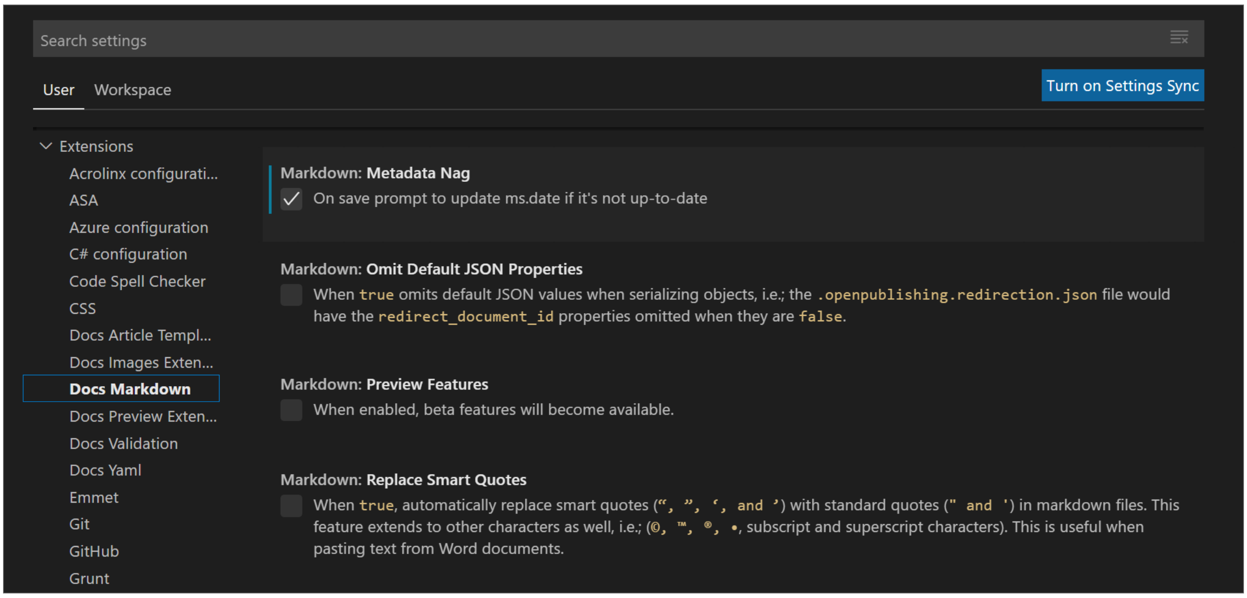
In action

Below is a brief demonstration of this feature.

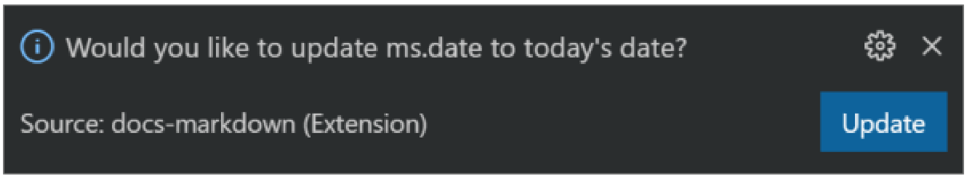


Get prompted to update `ms.date`

You can be prompted to update the date automatically by enabling the **Markdown: Metadata nag** setting:



When a file is saved, the following prompt appears if `ms.date` has not been updated:



To update the date to the current date, select **Update**. If you're making small changes that don't warrant an updated date, select the **x** to close the prompt. The prompt only appears once per file, per Visual Studio Code session.

Reformat Markdown tables

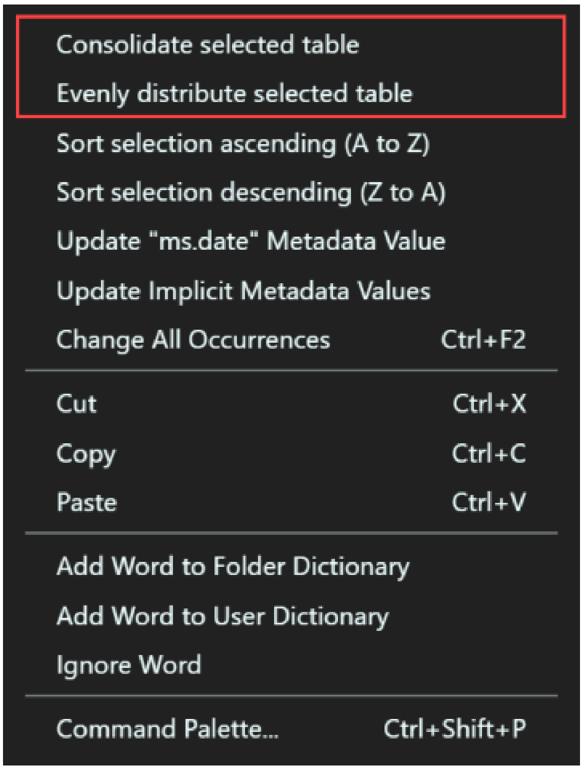
3/3/2020 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

In a Markdown (**.md*) file, when you select a complete table - two table formatting context menu items are now available. Right-click on the selected Markdown table to open the context menu. You will see something similar to the following menu items:



> **TIP**
>
> This feature **does not** work with multiple table selections, but rather is intended for a single Markdown table. You must select the entire table, including headings for desired results.

Consolidate selected table

Selecting the **Consolidate selected table** option will collapse the table headings and contents with only a single space on either side of each value.

Evenly distribute selected table

Selecting the **Evenly distribute selected table** option will calculate the longest value in each column and evenly distribute all the other values accordingly with space.

Considerations

The feature will not impact the rendering of the table, but it will help to improve the readability of the table - thus making more maintainable. The reformatting table feature will keep column alignment intact.

Consider the following table:

```
| Column1 | This is a long column name | Column3 |  |
|--:|---------|:--:|:----|
||         |  |         |
|     |  |         |   a value      |
||         |         |         |
|     |         | This is a long value |       but why? |
|     |         |         |         |
|     |                                           |         | Here is something |
|  |         |   |         |
```

After being "evenly distributed":

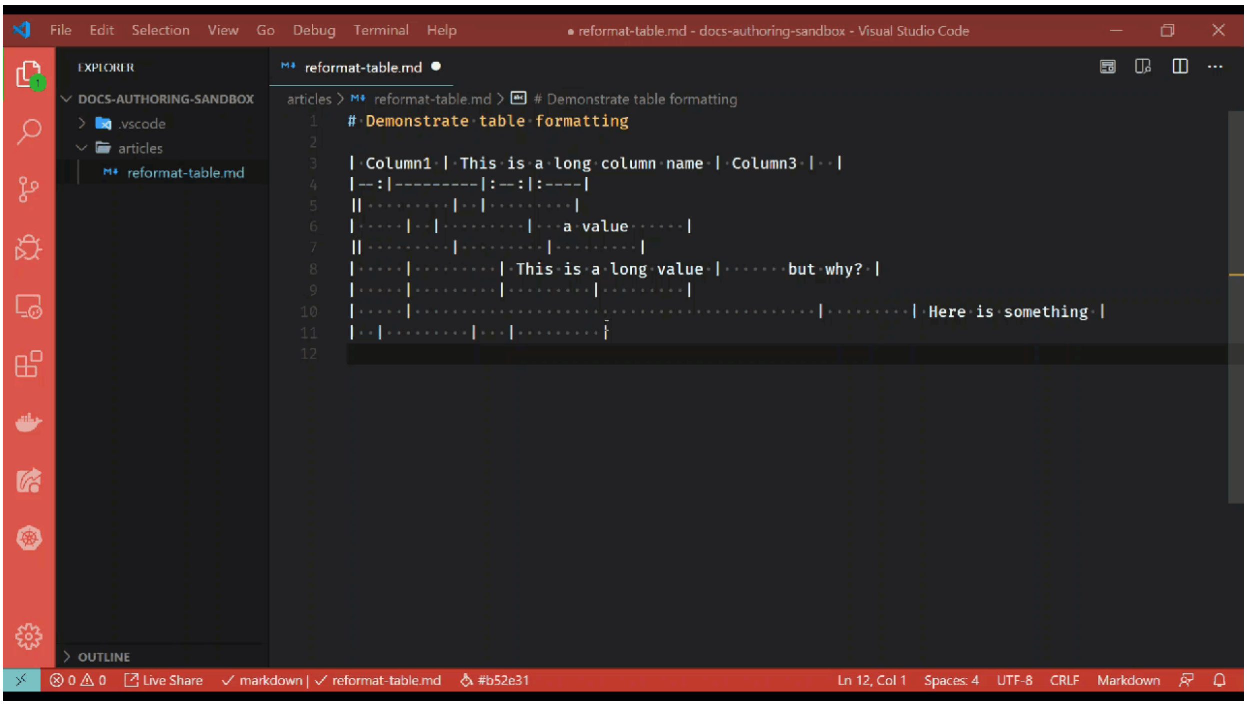
```
| Column1 | This is a long column name | Column3             |                   |
|--------:|---------------------------:|:-------------------:|:------------------|
|         |                            |                     |                   |
|         |                            |                     | a value           |
|         |                            |                     |                   |
|         |                            | This is a long value | but why?          |
|         |                            |                     |                   |
|         |                            |                     | Here is something |
|         |                            |                     |                   |
```

After being "consolidated":

```
| Column1 | This is a long column name | Column3 |  |
|-:|--|:-:|:-|
|  |  |  |  |
|  |  |  | a value |
|  |  |  |  |
|  |  | This is a long value | but why? |
|  |  |  |  |
|  |  |  | Here is something |
|  |  |  |  |
```

In action

Below is a brief demonstration of this feature.


File Edit Selection View Go Debug Terminal Help
● reformat-table.md - docs-authoring-sandbox - Visual Studio Code
EXPLORER
DOCS-AUTHORING-SANDBOX
.vscode
articles
reformat-table.md
OUTLINE
reformat-table.md ●
articles > reformat-table.md > # Demonstrate table formatting
Demonstrate table formatting
Column1	This is a long column name	Column3	
			a value
		This is a long value	but why?
			Here is something
0 0 Live Share ✓ markdown | ✓ reformat-table.md #b52e31
Ln 12, Col 1 Spaces: 4 UTF-8 CRLF Markdown

Smart quote replacement

3/3/2020 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

Content developers are responsible for authoring some of the most advanced features of modern technology and intelligence, yet our tooling today prefers the usage of "dumb quotes" in Markdown. The antonym of course being "smart quotes". Smart quotes are common with ideal typographies, but not preferred with Markdown and rendered HTML.

When working on a Microsoft Word document for example, you may have noticed that when you hold the `Shift` and type a " Microsoft Word quickly replaces the `"` character with a smart quote equivalent `“` character.

DESCRIPTION	UNICODE	SMART QUOTE	REPLACEMENT
Double left quote	`\u201c`	`“`	`"`
Double right quote	`\u201d`	`”`	`"`
Single left quote	`\u2018`	`‘`	`'`
Single right quote	`\u2019`	`’`	`'`

In a Markdown (**.md*) file, when you paste in text or as you update content - this feature will actively evaluate the content and automatically replace values accordingly.

> **NOTE**
>
> The smart quote replacement feature also replaces other characters, such as but not limited to; (`©, ™, ®, •`, subscript and superscript characters). This is useful when pasting text from Word documents.

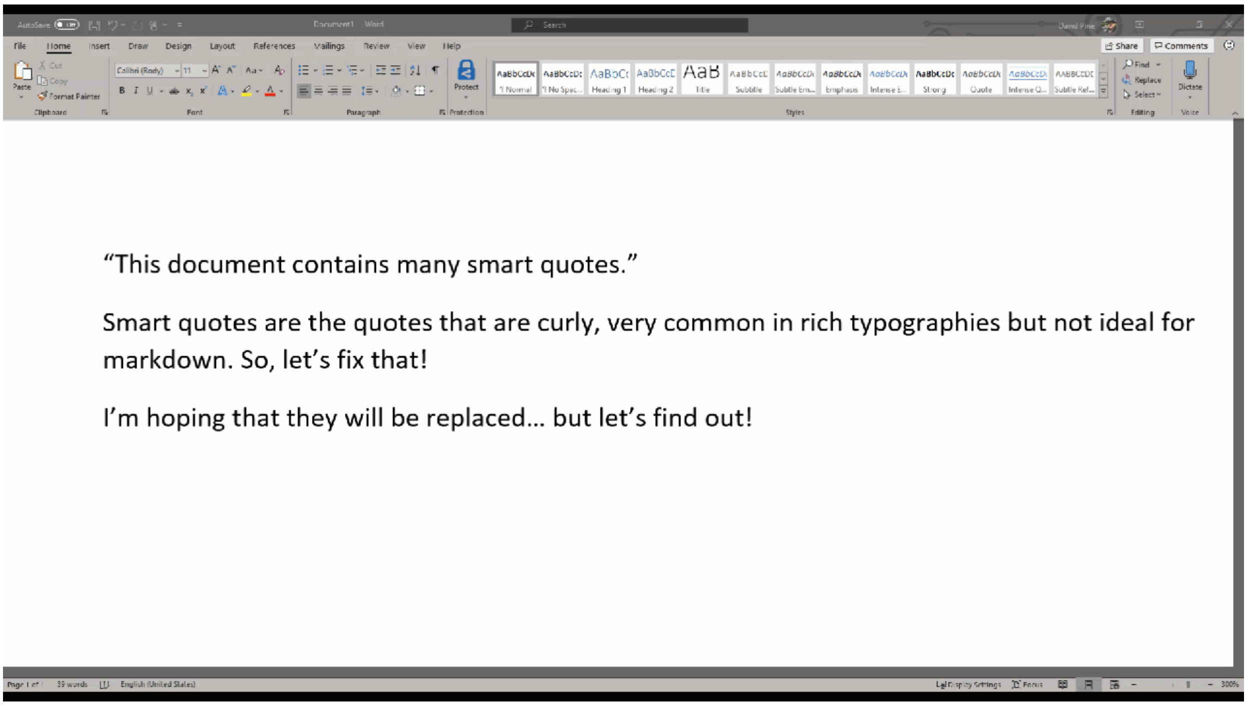
Preferences

This feature is optional, but defaults to `true`. To toggle this feature on or off:

1. Select **File** > **Preferences** > **Settings** and filter by *Docs Markdown Extension*.
2. Toggle the setting in the **Markdown: Replace Smart Quotes** section.

In action

Below is a brief demonstration of this feature.



“This document contains many smart quotes.”

Smart quotes are the quotes that are curly, very common in rich typographies but not ideal for markdown. So, let’s fix that!

I’m hoping that they will be replaced… but let’s find out!

Sort redirects

3/3/2020 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

With the evolution of a docs.microsoft.com docset, some Markdown files eventually will be deleted. When a Markdown file is deleted, we're required to provide a redirect so that any reference to the deleted article is properly resolved via the redirect. Redirections are specified in the *.openpublishing.redirection.json* file.

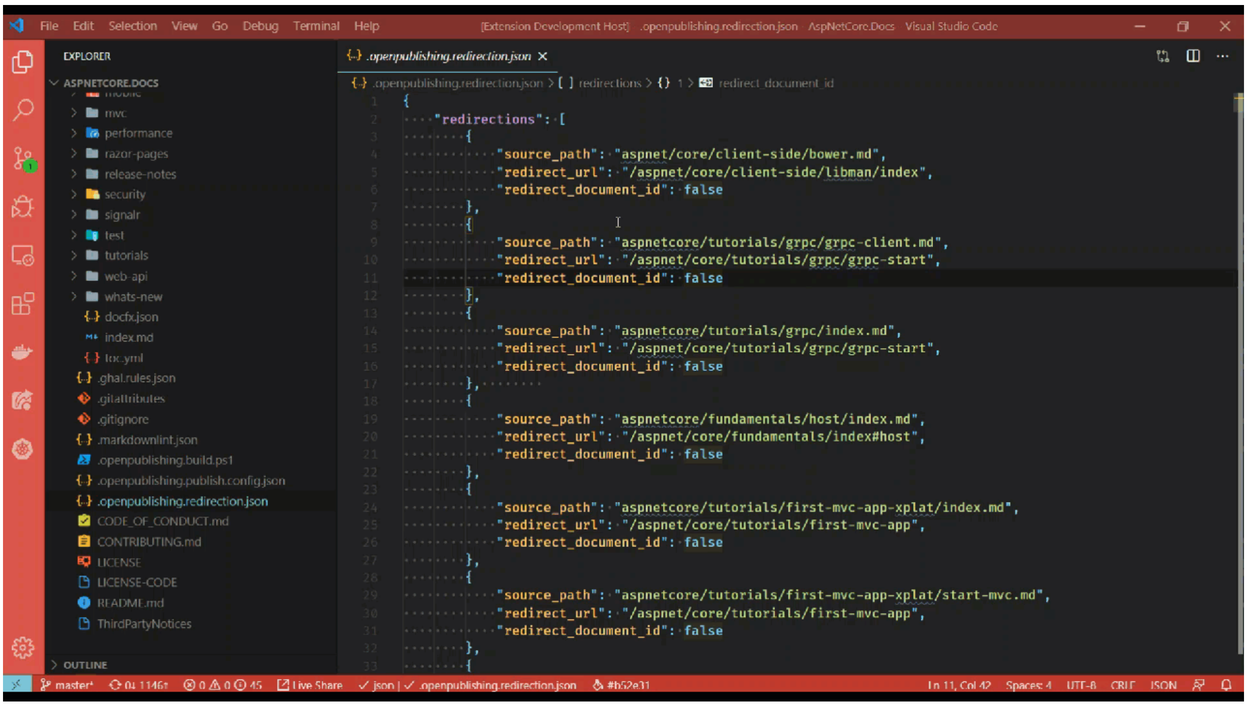
1. Open the Command Palette, press `F1` (or ⇧⌘P on macOS)
2. Type: `Docs: Sort master redirection file`
3. Select the command to execute it
4. Observe changes to *.openpublishing.redirection.json* file

Considerations

The concept of "daisy chaining" exists with how the *.openpublishing.redirection.json* file was originally designed. Over time, files added as a redirect will eventually become stale. This happens when file A is deleted and needs a redirect to file B, then later file B is deleted and then redirects to file C. Ideally, both entries would point to C - so that A redirects to C, and B remains the same. This is a minor performance gain, and the feature is actively being worked on.

In action

Below is a brief demonstration of this feature.



Sort selection

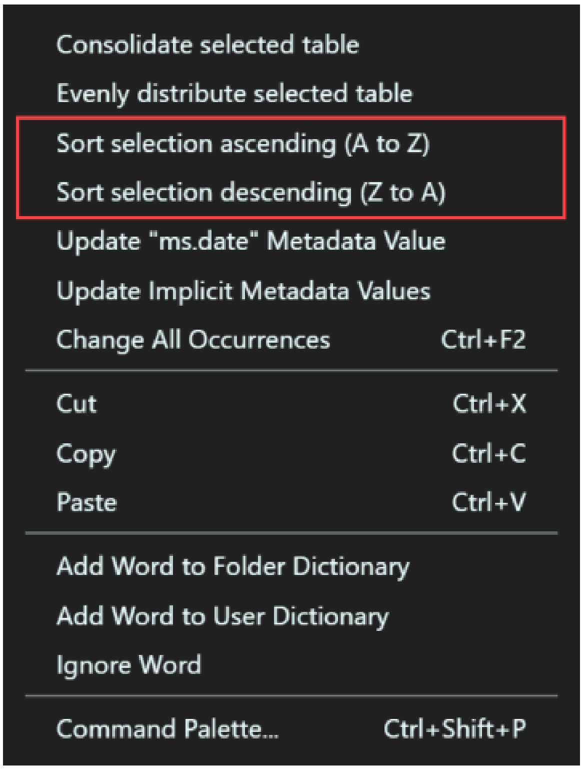
3/3/2020 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

Summary

In a Markdown (*.md*) file, when you've made a selection - two sorting context menu items are now available. Right-click on the selected text to open the context menu. You will see something similar to the following menu items:



TIP

The sorting context menu items are hidden until there is a valid selection in the Visual Studio Code text editor.

Sort selection ascending (A to Z)

Selecting the `Sort selection ascending (A to Z)` option will sort the entire selection ascending, alphabetically from A to Z.

Sort selection descending (Z to A)

Selecting the `Sort selection descending (Z to A)` option will sort the entire selection ascending, alphabetically from Z to A.

Considerations

The underlying sorting mechanism uses *natural language* sorting. This makes it more powerful and comprehensive than standard sorting. Consider the following table:

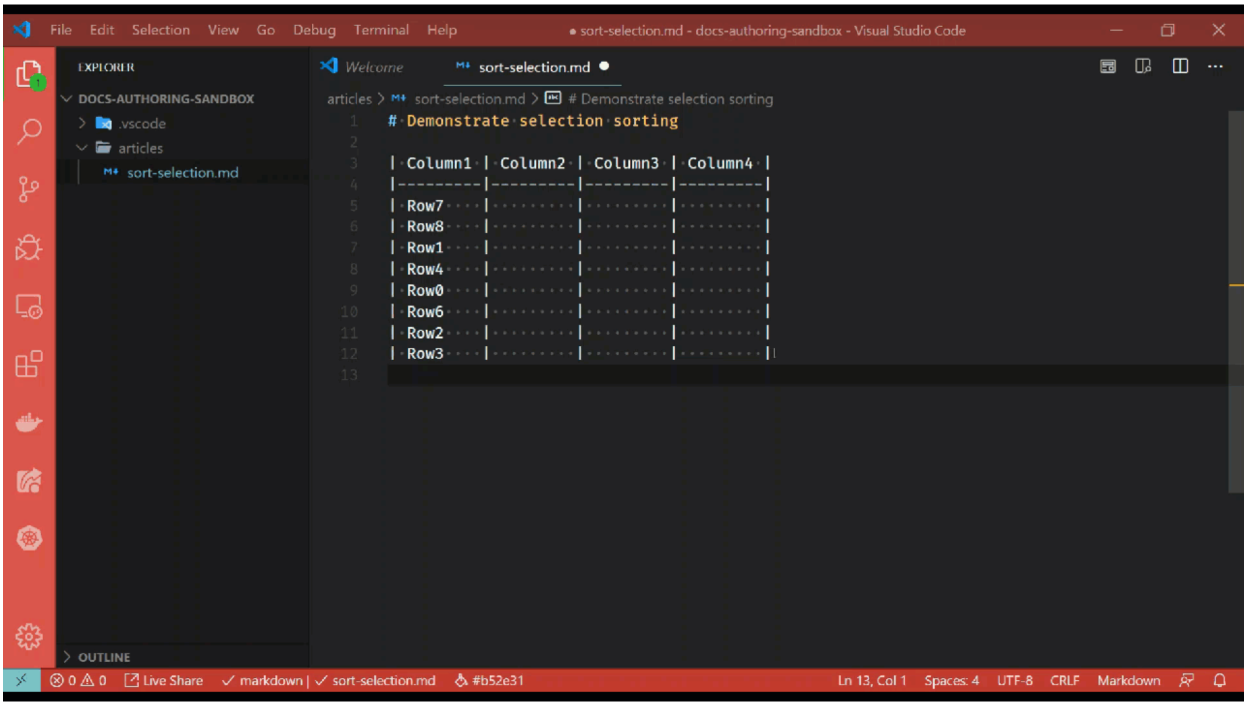
```
| Column1 | Column2                                |
|---------|----------------------------------------|
| 1       | Number 1                               |
| Aa      | The first letter in the alphabet       |
| Ab      | The first letter in the alphabet       |
| C       | The a letter after A in the alphabet   |
| M       | Somewhere in the middle?               |
| 2       | Number 2                               |
| X       | The alphabet letter is towards the end |
| Z       | The last letter in the alphabet        |
| 11      | Number 11                              |
```

Without natural language sorting, the order for `Column1` would have been 1, 11, 2, etc. but instead it understands that 11 is greater than 2 - resulting in the following ascending order:

```
| Column1 | Column2                                |
|---------|----------------------------------------|
| 1       | Number 1                               |
| 2       | Number 2                               |
| 11      | Number 11                              |
| Aa      | The first letter in the alphabet       |
| Ab      | The first letter in the alphabet       |
| C       | The a letter after A in the alphabet   |
| M       | Somewhere in the middle?               |
| X       | The alphabet letter is towards the end |
| Z       | The last letter in the alphabet        |
```

In action

Below is a brief demonstration of this feature.



Insert and update content from a Jupyter notebook

3/5/2021 • 2 minutes to read

Extension name

The Docs Authoring Pack, a Visual Studio Code extension, is comprised of multiple sub-extensions. This feature is included in the Docs Markdown extension. The Docs Markdown extension is part of the Docs Authoring Pack, so you don't need to install it separately.

> **IMPORTANT**
>
> Insert and update from a Jupyter notebook functionality does not currently work on a Mac.

Summary

Jupyter Notebooks are a standard interactive way of creating and sharing code in the Python world. The notebook contains a combination of Python code, markdown, and optionally output from the code.

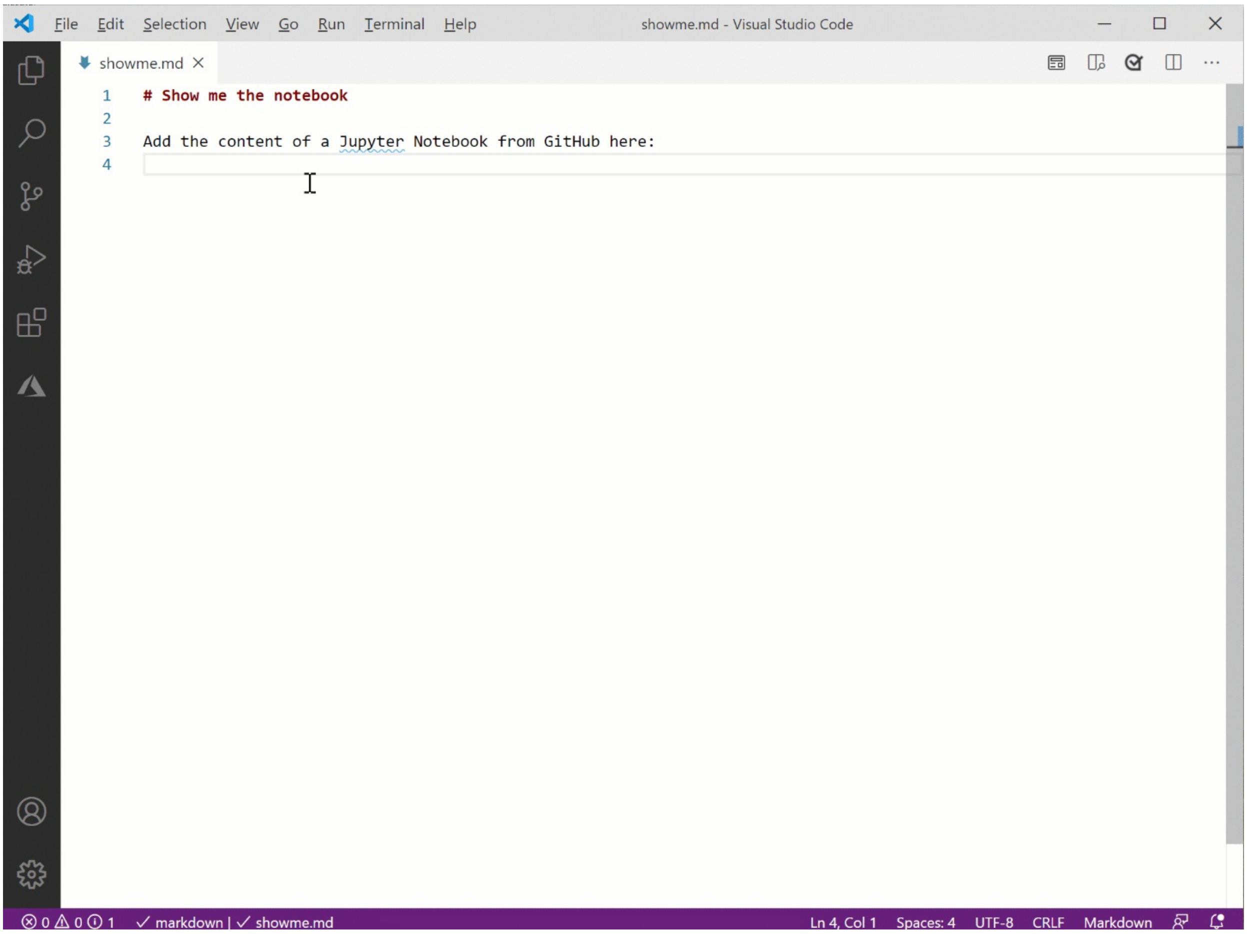
The Docs Authoring Pack extension includes functionality to put a static markdown version of a Jupyter notebook into your document:

Docs: Insert Jupyter notebook: Enter the URL of the notebook. A markdown version of the notebook is added to the document at the position of your cursor. Do not modify the start or end tags; it's what the next function uses to update the notebook.

Docs: Update Jupyter notebook: This function will replace the previously inserted notebook content between start and end with the latest version. No need to enter the URL, it's recorded in the start tag. Update functionality assumes there is a single notebook in the document. Don't add multiple notebooks to a single document.

In action

Below is a brief demonstration of this feature.



Troubleshooting

IMPORTANT

Insert and update from a Jupyter notebook functionality does not currently work on a Mac.

These functions need Python, `jupyter`, and `nbconvert` installed on your machine.

To see if Python is installed, open a VS Code terminal and run:

- Windows - `where python`
- Linux/Mac - `which python`

If the return is one or more paths, Python is installed. If not, install Python now.

Next make sure `jupyter` is installed:

- Windows - `where jupyter`
- Linux/Mac - `which jupyter`

If a path is not returned, install `jupyter`:

```
pip install --upgrade jupyter
```

Once both Python and `jupyter` are installed, install `nbconvert`:

```
pip install --upgrade nbconvert
```

Repo cleanup tool (CleanRepo.exe)

5/8/2021 • 3 minutes to read

The CleanRepo.exe command-line tool helps you clean up your repo. To run the tool, you must have a local copy (clone) of the repo you want to run it against. It can:

- Find articles that aren't linked from a TOC file.
- Find, and optionally delete, orphaned media (.png, .jpg, .gif, and .svg) files.
- Map images to the files that reference them.
- Find, and optionally delete, orphaned "shared" Markdown files (include files).
- Format the redirection file using Serialize with pretty printing.
- Find and replace links to redirected files in markdown files.
- Replace "site-relative" links with file-relative links in markdown files.

Installation

To install the tool, either clone the cleanrepo repo and build it yourself, or download and unzip the executable files to a folder on your computer. Then, open a command prompt and change to the directory where you created or copied the executable files.

Usage examples

The next sections show you the command-line options to use with the tool for different functions. You must have a local copy (clone) of the repo you want to "clean" using the tool.

For the functions that modify files, for example, replacing links or deleting orphaned files, don't worry too much about doing something irreversible. It's simple to open up Git Bash (or Visual Studio Code, if you use that for Git operations) and undo the changes. For example, `git checkout -- .` undoes all the changes, including file deletions. To undo file changes in a specific directory, use a command like `git checkout -- pipelines/.`.

> **TIP**
> Consider the execution order of commands. For example, if you intend to delete orphaned images and orphaned articles, you should delete orphaned articles first, then delete orphaned images. The reason for this is that deleting articles could potentially orphan further images.

Orphans

The tool has the ability to delete orphaned markdown files and orphaned media files.

Orphaned articles

This option finds Markdown files that aren't back linked from a TOC.yml or TOC.md file. It ignores Markdown files in directories named *includes* or *_shared*.

Find and delete orphaned articles recursively (that is, in the specified directory and any subdirectories):

```
CleanRepo.exe --orphaned-topics --start-directory c:\repos\visualstudio-docs-pr\docs\ide
```

Orphaned images

Find and delete orphaned .png/.jpg/.gif/.svg files (recursive):

```
CleanRepo.exe --orphaned-images --start-directory c:\repos\visualstudio-docs-pr\docs\ide
```

Orphaned include-type files

Find and delete shared Markdown files that are orphaned (recursive):

```
CleanRepo.exe --orphaned-includes --start-directory c:\repos\visualstudio-docs-pr\docs\ide
```

> **NOTE**
>
> This option considers markdown files in a directory named *includes* or *_shared* to be shared markdown files. It ignores markdown files in directories that aren't named *includes* or *_shared*.

Cataloging

This option generates a JSON file that lists each image (.svg/.jpg/.png/.gif) along with a list of all the files that reference the image.

```
CleanRepo.exe --catalog-images --start-directory c:\repos\visualstudio-docs-pr\docs
```

For example, the output might look like this:


```
{
  "c:\\users\\gewarren\\test-repo\\docs\\debugger\\dbg-tips.png": [
    "c:\\users\\gewarren\\test-repo\\docs\\dual-stack.md",
    "c:\\users\\gewarren\\test-repo\\docs\\dual-stack.md"
  ],
  "c:\\users\\gewarren\\test-repo\\docs\\media\\stratus.png": [],
  "c:\\users\\gewarren\\test-repo\\docs\\media\\two-forest.png": [
    "c:\\users\\gewarren\\test-repo\\docs\\landing.yml"
  ],
  "c:\\users\\gewarren\\test-repo\\docs\\media\\vs.png": [
    "c:\\users\\gewarren\\test-repo\\docs\\configure-ldaps.md"
  ],
  "c:\\users\\gewarren\\test-repo\\docs\\debugger\\dbg-tips.jpg": []
}
```


Redirects

These options relate to redirect links and the redirection JSON file.

Format redirection file

Formats the redirection JSON file by deserializing and then reserializing it using "pretty printing" indentation.

```
CleanRepo.exe --format-redirects --docset-root c:\users\gewarren\dotnet-docs\docs
```

Replace redirected links

Find articles with backlinks to redirected articles and replace the links with their target URL:

```
CleanRepo.exe --replace-redirects --start-directory c:\repos\visualstudio-docs-pr\docs\ide
```

Site-relative links

Replace any site-relative links (that is, those that start with `/<docsetname/`) with file-relative links (for example, `../misc/index.md`). File-relative links are validated at build time, whereas site-relative links are not.

```
CleanRepo.exe --relative-links --start-directory c:\repos\dotnet-docs\docs\core
```

TIP

The Docs Authoring Pack provides functionality to sort and remove daisy chains from a redirection file. For more information, see Docs Authoring Pack: Sort redirects.

See also

- Readme file for CleanRepo.exe
- Docs Authoring Pack: Sort redirects

NOTE

CleanRepo is an "in-house" tool written by a content developer, so you may find bugs or issues with the tool. If you do, please file an issue or submit a PR to fix the problem.

SpineEdit

4/28/2021 • 2 minutes to read

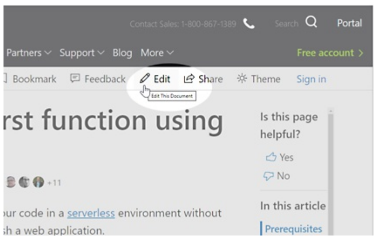
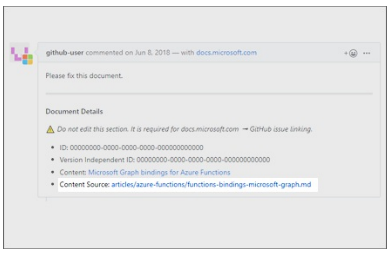
SpineEdit is a browser extension that makes it easy for you to edit docs in the private repository - all in the browser.

Prerequisites

SpineEdit requires you to have read permissions in the private MicrosoftDocs repository you want to access. To learn how to request access to this repository, see this GitHub account setup article.

Make an edit

With the extension installed, you can open an editor in the article's private repository.

WEBSITE	CLICK ON THE...	EXAMPLE
docs.microsoft.com	*Edit* button	
github.com	*Content Source* link in an article issue	

NOTE

This workflow only works if you have access to the destination repository.

Installation

Install via the Google webstore.

Feedback and support

For feature requests and bug reports, create an issue in the GitHub repository.

Get to know the Global Experiences Content Localization team

5/10/2021 • 2 minutes to read

DevRel Global Experiences team is localizing all software, sites, technical and Learn content for the Cloud & AI division at Microsoft. The products we support range from Azure to SQL, Visual Studio, .NET, Intune, System Center, to PowerBI and more. In addition to owning the localization of products and services, we are the product owners of the localized technical and Learn content published to docs.microsoft.com with the goal of promoting our content usage.

General Localization Process

C+AI docs localization follows the Continuous Publishing model (CPUB). Most loc projects run in the regular cadence automatically.

For a new localization request, the Global Experiences Content PM team defines the loc cadence, localized language set and loc file set. It takes us one week to onboard the new repository or new docset to the localization system. The initial translation time varies depending on the repo scope and language set. Afterwards, the localization turns to CPUB mode. Contact your project PM owner about your project.

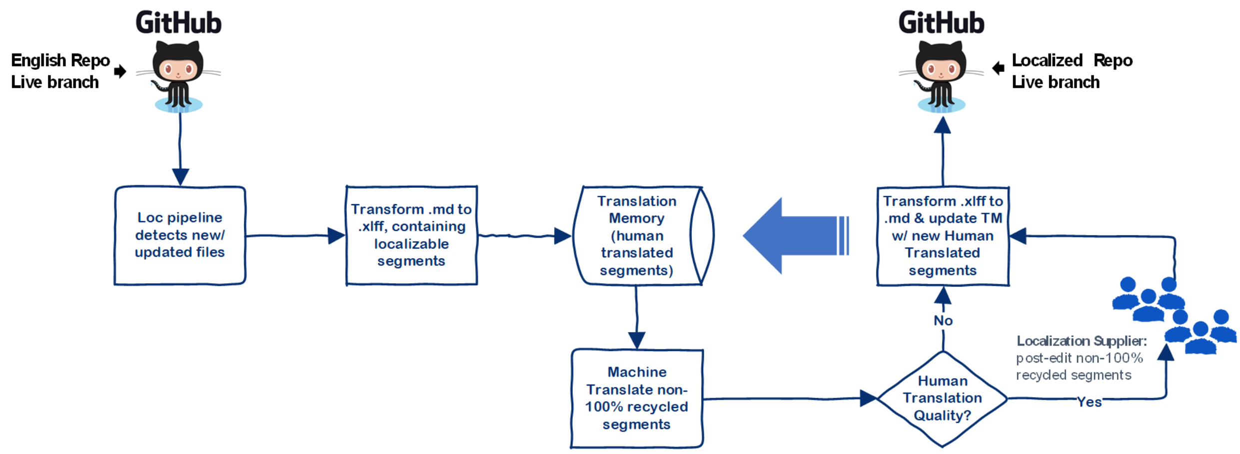
> **IMPORTANT**
> DevRel Global loc team supports the localization of the content authored by C+AI teams. When you have a new set of content haven't been published to DOCS and you are not sure whether they are onboarded to localization pipeline, always contact your content loc PM before merge files to the publishing branch! We'd like to avoid over-localization, thus cross-charge to your team afterwards.

Besides the translation, the localization process supports:

- **Terminology Change Request**: this process defines what to do when the loc team receives the customer or internal feedback about one term translation for a specific language.
- **Source Issue Escalation**: through this process, we provide the translators a channel to escalate questions about the source sentences when they pursue the accurate meaning.
- **Policheck**: the process to check politically sensitive terms in the localized version is a required process for any Microsoft product and content releases. The Content & Learning loc team runs Policheck every month and reviews the false positives term list.

Localization Automation workflow

The DevRel localization tooling automatically detects new or updated files within our localization scope. Loc files are created and pre-translated by using our Translation Memory, which contains all of our previously human translated content. Any segments that cannot be recycled are then machine translated by our own Azure Translation service. All files set to be human translated are sent to our localization supplier, who reviews and post-edits the pre-translated segments to ensure premium quality human translations.



Localization contacts

Contact the DevRel Global Experiences Loc Pillar PM for general localization questions. The following list shows the PM owner for each content area:

- Content Localization escalation: Éric Van Thorre for Azure Content; Sonja Saltzman for non-Azure Content and Learn
- Microsoft Learn/LearnShared: Nobuko Kurashige
- Azure Docs: Sunny Deng
- Azure Devlang: Chris Niccoli
- Developer Tools & Services: Jason Card
- PowerBi/PowerQuery: Alisha Acharya
- SQL Server: Chris Niccoli
- System Center: Sunny Deng
- DOCS roots/template/tech profile: Alisha Acharya
- Enterprise Mobility: Nobuko Kurashige
- Windows Dev Content: Chris Niccoli
- DOCS Legal: Sonja Saltzman
- DOCS Blog: Alisha Acharya

Frequently asked questions (FAQ) about localization

3/9/2021 • 11 minutes to read

The Developer Relations Global Experiences team owns the localization processes, tooling, and work model for strategic C+AI content. We provide a predictable release schedule and systematize the GitHub workflow for localization. This article addresses common questions about localization.

General localization for C+AI

Do I have a specific localization contact?

Existing repos have 1-2 loc PMs assigned: see Localization contacts. If you cannot identify your loc contact there, email to DevRel Global Experiences Loc Pillar PM.

How can I request localization for new content?

- If the content is not/will not be published to docs.microsoft.com, contact the loc PM assigned to your product area (see Localization contacts). If you can't find your product area, email the general team: DevRel Global Experiences Loc Pillar PM.
- If the content is/will be published to docs.microsoft.com and you submitted an onboarding request, @mention your localization contact in the onboarding ticket (you will receive a link to the ticket after you complete the form).

> **NOTE**
>
> Not every new repo or docset is localized by default—make sure to notify your loc PM. The PM will go through the loc selection criteria and make the final decision. See How do we select content to localize for details.

What languages do you support?

Technically, we support almost all languages for localization. But different projects may support different sets of languages, including: the strategic requirements; the necessity to align with the SW support language; Microsoft sub and marketing team feedback etc. The final decision is made by the Global Experiences team—refer to the C+AI Language Set for the language set concept.

See below for Azure-specific language availability.

Will *all* my articles be localized for *all* locales?

If a content set is scoped for localization, all articles in that set will be localized, either via machine translation or human translation. Each Global Experiences PM works with product and content team stakeholders to decide which method to use for which articles, depending on the priority of the content and the available budget. We try to align our locales with software and UI localization.

For detailed information, read about the general selection process and also which Azure topic types are prioritized.

I have a new repository. Will my content be localized?

Not every new repo is localized by default. To request localization, see How can I request localization?. The Global Experience PM will go through the loc selection criteria and make the final decision.

If you work with the Docs Contributor Program or a Content Team member, they can help guide you through this step.

My content was added to an existing repository. Do I need to tell the loc team?

Localization scope is defined by file metadata or the repo's folder structure. If new content uses the predefined metadata or folder names, then the Global Experiences team tooling will detect the file-level changes automatically. Here are some repo-specific scope examples:

- Azure content localization is defined by ms.topic.
- Windows Dev Content localization is defined by ms.localizationpriority.
- SQL Server and DevDiv localization is defined by the folder structure.

To be safe, contact your loc PM when you create a new folder in an existing repo because this can affect their budget and the localization persistency.

I have an existing translation for my article in Word format. Can I pass this along to you for publishing?

We prefer you submit feedback directly in the localized repo (for example, as a GitHub issue). You can find this repo if you select **Edit** on a Docs page from the corresponding locale. After we receive the translation, we run a linguistic quality review to make sure the translation complies with the Microsoft Style Guide and terminology definitions.

If the translated language is not supported for that project, contact your project loc owner who will add the scenario in the backlog for the future EOL evaluation.

When do you start translation of my article?

For repos and folders where localization is configured, the loc pipeline will detect changes when your article is merged into the live branch of the source repo. This kicks off the handoff depending on the loc cadence for that repo/project. Then the translator will receive the new or updated articles through the localization pipeline.

We define the loc cadence when we onboard a project. For example, Azure docs handoff kicks off at 5:00 pm every Wednesday. Contact your project loc owner for more about your project.

Why don't I see the localized version one month after I published my article?

The reasons vary by project. Contact your project loc owner to discuss timelines or request a faster release.

Do I need to tag my content somehow to ensure it is localized?

It depends on the repo. See My content was added to an existing repository...

Do you localize art and image files?

Art localization is slow and expensive. The only art localized by default is for Microsoft Learn modules, and the overview and quickstart topics for Azure and Power BI. All other art is by default shown in English.

Occasionally, we make an exception. The acceptance criteria, however, is strict. For example, we will localize art to avoid potential legal/compliance issues or to comply with Microsoft branding in each international market.

To request an exception, contact your project loc owner.

How can I make sure the right words are being translated?

Make sure to capitalize only what should be capitalized, for example, product or service names. This will help the loc team identify a possible proper noun and not translate words unnecessarily. If a word is translated that shouldn't be, it will be confusing to the local reader. Likewise, if a word is not translated but should be, it is equally confusing.

Will you translate my code?

The loc team will never localize anything formatted as code block or code fence. You can use `inline code style` to prevent translation of code elements within a sentence.

How can I exclude a word from localization/translation?

Some product details and labels need to stay in English, regardless of the reader's locale. In these scenarios, you can identify content to exclude with the wrapper `:::no-loc text="Words to exclude":::`. See the New Hope guide for detailed information. This feature only works for Markdown topics.

Another option is to use inline code style, as explained above, but note that this will change the font formatting.

How do I respond to GitHub issues and public PRs in other languages/locales?

Going forward, DevRel Loc team will partner with other teams at Microsoft (e.g. subsidiaries, events, marketing, ...) and focus on driving usage of the localized technical documentation. We will no longer make GitHub repos public for localized content due to this change. Any existing PRs and issues will be closed, with the exception of azure-docs, which will remain open for community contribution for human translated topics for 6 languages (Simplified Chinese, French, German, Japanese, Korean, Spanish). In addition, starting this summer, our localization pipeline will be updated and will no longer be based on GitHub repos, which will make the current community contribution process obsolete.

We still respond on-demands feedback of the localized content. Click https://aka.ms/devrellocfeedback to create ADO feedback ticket or refer details about Provide Feedback to DevRel Localization Team.

Will you localize legal content, such as our policies or license information?

No. Any content with legal, policy, or compliance implications are outside the scope of the Global Experiences team. This content needs to go through the CELA team first and is localized by special contractors. Individual product teams incur this cost. For detailed information, see our localization guidelines for legal articles.

CELA hosts pre-localized legal text in a separate repo. When possible, you should point to existing legal content that has already been localized. This includes:

- End User License Agreement (EULA)
- Terms of Service (TOS)
- Terms of Use (TOU)
- Licenses
- Privacy statement
- THIRD-PARTY SOFTWARE NOTICES

> **IMPORTANT**
> Don't put legal documents in a technical documentation repo. They will be translated, which can have legal implications for the company. If you plan to publish legal documentation, regardless of localization needs, please see Legal content.

My service will onboard to Azure China (code name Mooncake). Can you help release the content to docs.azure.cn?

Yes. DevRel Global Experiences team handles content adaptation, localization, and release to Azure China docs.

The adaptation refers editing Azure docs to be updated according to Azure China features, and Azure branding will be adapted for Chinese market. All reference site links and reference article links will be validated for the Chinese end users.

The adptation team monitors Cayman to identify the newly onboarded service to Azure China, synchronizes the global content from the Azure repo live branch three weeks before the launch date, they contact the service Engineering PM or you to understand the specific Azure China feature. The adaptation takes one or two weeks depending on the article number or the adaptation complexity. Then the localization takes one week.

For any previously onboarded service to Azure China, the adaptation team also maintains the content and checks for updates every two weeks.

Our Azure China content structure aligns with the global Azure version, however we don't support extra articles or TOC entries in Azure China docs. Contact Sunny Deng with any questions.

Learn localization

Which languages are supported for Learn localization?

Learn Azure

Almost all Learn Azure and derivatized Azure modules are localized in 17 languages, which cover learn-pr repo folders of \azure, \advocates, \ai-school, \aspnetcore, \cmu-cloud*, \commercial-marketplace, \data-ai-cert, \databricks, \github, \iot, \language, \quantum, \student-evangelism, and \reactors etc.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, Simplified Chinese, Czech, Dutch, Hungarian, Swedish, Turkish, Polish, and Portuguese (Portugal)

> **NOTE**
>
> Due to the popularity of some specific modules, we also support additional languages: all the modules in **Azure Fundamental** learning path are localized in Danish, Finnish, and Norwegian; **Introduction for Python** are localized in French (Canadian), Hindi and Arabic.

Learn Windows

All Learn Windows modules in \windows folder of learn-pr repo are localized in 10 languages.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, and Simplified Chinese

Learn SQL Server

All Learn SQL Server modules in \sqlserver folder of learn-pr repo are localized in 10 languages.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, and Simplified Chinese

Learn Xamarin

All Learn Xamarin modules in \xamarin folder of learn-pr repo are localized in 10 languages.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, and Simplified Chinese

Learn Power BI

All Learn Power BI modules in \powerbi folder of learn-bizapps-pr repo are localized in 23 languages.

- German, French, Italian, Spanish, Portuguese (Brazil), Russian, Japanese, Korean, Traditional Chinese, Simplified Chinese, Hungarian, Swedish, Czech, Dutch, Iberian Portuguese, Polish, Turkish, Danish, Greek, Slovak, Finnish, Thai, and Norwegian

Azure localization

Which languages are supported for Azure docs localization?

We localize 17 languages for Azure docs, the language set is the same as Ibiza portal language set.

- **APEX10**: German, French, Italian, Spanish, Russian, Portuguese (Brazil), Japanese, Korean, Traditional Chinese, and Simplified Chinese;
- **APEX13(+3)**: Czech, Polish, and Turkish;
- **APEX17(+4)**: Dutch, Hungarian, Portuguese (Portugal), and Swedish.

Do you localize articles and includes for *all* Azure services/folders in the same language sets?

We localize for all Azure services in the same language sets except the following content sets:

- The content in Azure Government Document stays in English for all languages.
- The content in Azure Germany Document stays in English for all languages except German.
- The content in Azure China 21Vianet Document stays in English for all languages except Simplified Chinese.

For all other services that we do localize, we prioritize articles by topic type (see next FAQ).

How do you prioritize the articles to localize for Azure docs?

We use the article's metadata **ms.topic** value to prioritize the articles for the localization language set and the release cadence.

- **Azure High Impact** content also known as **Azure HI**: refers to the article *ms.topic* value in either 'overview', 'quickstart', 'tutorial' or 'sample'.
  - For several non_MVC services, that is, nonstandard content, **Azure HI** also refers to the *ms.topic' value is either 'hero-article' or 'get-started-article'
- **Azure NON High Impact** content also known as **Azure nonHI**: refers to the other value of *ms.topic* of the articles.

When do you start to localize my article?

We currently pick up the English updates from the azure-docs-pr repo **LIVE branch** only. We schedule the hi-pri handoff at **every Tuesday 6:00pm PST** when we kick off the translation process.

- If your article stays in your own branch, the localization system can't detect the updates, thus there will be no localization.
- If you publish the article before Wednesday 3:00pm PST, then we pick up your article in that Wed handoff, and deliver to the translators.
- If you publish the article after Wednesday 3:00pm PST, then we pick up your article in the *next* Wed handoff, and deliver to the translators one week after.

How long will it take for the localized version to be published, for example, in Japanese?

As long as the localized files are delivered back in loc pipeline, it takes no more than 30 minutes for the content to be published live.

Azure docs is a CPUB project, we pre-defined the delivery cadence.

- For the **Azure HI** articles, the delivery is in five working days after the handoff, thus the next Wednesday after the handoff;
- For the **Azure nonHI** articles, the delivery is in 10 working days after the handoff, thus the second Wednesday after the handoff.
- For all the TOC and Index pages, the delivery is three workings days after the handoff.

Where can I find Simplified Chinese articles?

There are two Simplified Chinese versions for Azure docs:

- The global Simplified Chinese version document is translated from Azure docs English content, you may go to the global zh-CN version: `https://docs.microsoft.com/zh-cn/azure/index`
- The Azure China technical document is adapted from the Azure docs English content according to Azure China functions, you may go to Microsoft Azure 21st Vnet.

DevRel Authoring Checklists for Localization

3/5/2021 • 7 minutes to read

This article describes a series of checklists and guidelines for writers to keep in mind for localization in most common scenarios. Use these checklists and guidelines to work with your localization contact as you plan for new content or changes to existing documentation.

Who is your Localization Contact?

Find your Localization PM (loc PM) contact here.

Ask Loc Team Lead Sonja Saltzman if you are not sure who your loc PM is from link above.

New Products or Services

Contact your loc PM in advance before publishing content for new products or services.

- Is your content using any new product names, service names, branding names, or terminology?
- Is there a new repository created? Use @ Mention your loc PM in the repo onboarding ticket for visibility.
- Is there a new docset created?
- Is there a new sub-folder created in the existing repo?
- When is the English release date?

Major release

Contact your loc PM in advance before major content release such as below:

- **BUILD**, **Ignite** and **Connect**
- Major feature release with substantial update at one time
- Rebranding of product or Azure service

- Is there any confidential content that should **not** be localized? For example, **GA** or **pre-release**.
- Is there a release branch to be created for the content manager to sign off before merging to Master branch?
- Are there any files being moved among sub-folders?
- Keep term changes to a minimum including renaming of product, feature, or service.
- Is your content using any new product names, service names, branding names, or terminology?
- Is there any branding changes?
- When is the English release date?

SIM-SHIP

SIM-SHIP is to release or publish English and localized versions simoutanously, or within 24 hours on Microsoft DOCs. We usually do not support SIM-SHIP unless there's a mandatory requirement from the country regulation, the security risk, or any legal obligation. This is because sim-shipping requires additional manual work and coordinations.

To support SIM-SHIP, loc team must be notified **six weeks** before the SIM-SHIP day and the English "Content Complete" day must meet loc team's request.

- Is there a strong business justification to request for SIM-SHIP?

- File SIM-SHIP loc request ticket.

Content Migration

Contact your loc PM in advance before migration of your content to new website, repo, publishing environment:

- Is your content migrating from one repo to another? If so, is the new repo in same or different GitHub org?
- Are there any files being moved among sub-folders?
- Is there any new or renamed docsets?
- Is there any content being archived?

Doing something different, new, or experimental

Contact your loc PM in advance before using new or updated features such as below:

- Is there any file format changes for index, includes, Leanrning Path, or Module Unit files? For example, .md changes to .yml.
- Is there any content using a new file format to the existing repo?
- Is there any new or updated UHF, Zone pivots, or tabbed conceptual content?
- Is there any new or updated content for an AB test?

Content not on docs.microsoft.com

Contact your loc PM if your content to be localized is not publishing to docs.microsoft.com. Loc team will investigate and determine the loc approach, which is different from the DOCS localization.

- For exmaple, in-box release, GB18030 documents, Intellisense, blogs, etc.

- What is the file format?
- Where is your content published or hosted?
- When is the Content Complete date?
- When is the release date?
- How frequently will this content be updated?
- Who is the content owner to update the content?
- Who is the publishing owner that loc PM may contact to setup the localization pipeline?

Legal documentation

Legal documents is the content of EULA, Terms of Service, Terms of Use, Licenses, Privacy statement, or THIRD-PARTY SOFTWARE NOTICES. Legal documents are not localized by the same team and process as technical documentation.

- Is there any legal documentation living in the technical document repository? If so, **REMOVE them!**
- Is there any legal documentation to be localized? See Legal article localization guideline.

Common mistakes that seem fine in English but break or affect localization

English repo shouldn't contain any content not published to LIVE site

Localization pipeline automatically detects file changes in the live branch of your content repo. Every localizable files are handed off for translation unless they are specifically configured to be excluded. To avoid unnecessary translation cost, use of pipeline bandwidth, and translation resources, remove any content that is not being published from your repo.

- Is there any art files, media files, module units, articles, or includes files in your content repo that are not being referenced or published to LIVE site?

Unprotected Code

- Use correct syntax to prevent over-translation of code, command line, API names, etc.
- See How to include code in docs.

Broken Includes

- Use correct Includes Syntax!
- Not doing so will leave the content in Includes file unlocalized. Your topic will display mix of localized content with content from Includes files displayed in English!
- *INCLUDE* must be ALL CAPS, with a space after it. Like this: `[!INCLUDE [<title>](<filepath>)]`

Don't use "en-us" in URLs

- Don't hardcode locale codes such as `en-us` in your links.
- This rule applies to links to docs.microsoft.com or Azure.com where the language selection UI is available on the target site.
- See Locale codes in external links.

Don't use "aka.ms" links

- Don't use aka.ms links.
- See Don't use aka.ms link.

Make sure metadata "ms.topic" match the Standard ToC

- Is your content using the Standard TOC?
- If so, keep the metadata `ms.topic` (overview/quickstarts/tutorials/sample) match the ToC nodes (Overview/QuickStart/Tutorials/Sample).
- Besides being used for content metrics reporting, loc uses topic type to determine whether to use machine or human translation services.

Incorrect metadata

- Keep your metadata `author`, `ms.author`, and `manager` current and correct.
- Everyone, uses these values to quickly identify the topic owner. Loc depends on this information for when they have a translation bug or question.

Remarks on Art and Media files

When creating new badges or trophies for Learn content or icons or other conceptual images for technical documentation, please ensure the art pieces are not offensive or their intent is not misunderstood in certain markets. Note that some images are not localized (badges, trophies, icons), but some are potentially localized (screenshots, conceptual images):

- Are trophies, badges, icons globalized?
- Review the checklist created and maintained by the CELA team: Image Review Checklist to ensure your images are acceptable worldwide. If you have further questions or are not sure about your specific image, please contact your global readiness contact.
- Are Arts files linked correctly? See Feature Usage.
- If Art syntax is incorrect, then localization platform will not recognize that Art file as dependency to the

Article. Then that Art file will not get localized when it should be.

- Is the update to conceptual art/image files necessary?

- Localization of Art files takes longer time and more expensive than text.
- Any update to an Art file is same as adding a new art file from localization perspective.

- Is there any same Art files saved in different file name in different folder or in different file format?

- Avoid having duplicated Art files to reduce localization cost.

- Is renaming art file necessary?

- Don't change between **upper case** and **lower case** in file name. It causes the localization pipeline pending where a manual data fix is needed.
- Our pipeline cannot differentiate such changes and will consider the renamed file as new art file.

- Are the UI strings in content consistent with the UI strings in screenshot?

- If there is inconsistency between the text and screenshot in the content, it will not only confuse the translator, but also the customers. And fixing those inconsistencies later will also increase localization cost.

- Avoid using dynamic gif.
- The video should be published on RedTiger where localization tool supports.

Project-Specific Requests

Here are some project-specific requests for each product family:

Microsoft Learn

Contact your loc PM if your Learn content meet following scenarios:

- New Module Group is created
- Module is shared across multiple repos
- Content in one repo is shared between different localization groups

Azure Docs

- Are all **service names** using title case and capitalized consistently?

- This request is critical for localization. Especially when the product/service name shouldn't be translated, but common noun of the same term should be translated. Inconsistency in capitalization will not only confuse translators, but also confuse customers.

- Is the new service metadata added to whilelist through new ms.service value request form?

- This request is critical when Loc PM reviews BI data (for example, Page Views) to break down the BI data based on correct service metadata.

SQL Server / Dev Tools & Services

Contact your loc PM when any major refactoring is done to existing source content.

- Adding new folder, renaming folder name, or moving content from one folder to another can cause the files in those folders to be unlocalized.
- The folder level change could also cause human translated content to be machine translated, or vice versa.

Windows Dev Content

- Is metadata **ms.localizationpriority** up-to-date?

- The metadata **ms.localizationpriority** is used to define the translation quality only when Loc team and

Content owners agreed beforehand. The standard practice is if it is **high**, we will localize in Human Translation quality while if it is **medium** or **empty**, we will localize the content as Machine Translation quality.

Localization guidance for Docs writers

11/2/2020 • 2 minutes to read

Do not force-push files in the en-US source repo

When the en-US source repo is force-pushed (`git push -f <remote> <branch>`), there can be conflicts. These conflicts, while unresolved, can block the syncing of the localized repo structures.

The OpenLoc localization tool (OpenLoc) is constantly monitoring the specified branch (typically /live branch for cpub projects, /master for milestone projects) and any file moves or deletions performed by writers on the en-US repo are then automatically done on the loc repos for all locales. If there are repo merge conflicts on the english repo, the loc tool is blocked from working properly.

If you ever really need to force push in the en-US repo, please contact your localization team first, so we can involve the dev team to support this on the loc repo side without breaking our tooling.

What are impacts of source repo Force-Push?

- Git conflicts when syncing with remote source repo. OpenLoc Handoff and structure-syncing are blocked until conflicts were manually resolved.
- Last processed commit in OpenLoc cannot be found in current force-pushed live branch. OpenLoc cannot calculate revision number based on last processed commit.

How to fix Manually resolve merge conflicts in all OpenLoc local repos Fix all OpenLoc file revision data according to current force-pushed live-branch.

NOTE: Avoid Force-Push unless necessary Force-push in the en-US source repo requires heavy data-fixing on the OpenLoc system data. It should be avoided unless absolutely necessary.

Localization of legal content

5/10/2021 • 4 minutes to read

The Global Experiences Content Localization team does not manage localization of any product- or service-specific legal content. The current loc supplier (MOR) is certified for technical documentation translation only under the Content & Learning SOW.

The exceptions are site legal documents for the Microsoft Azure site and the Microsoft Docs site. There is a standard template for the en-US EULA legal document, which already exists in most or all of the localized locales. If the standard template is used, then there is no need to arrange for localization of an additional document.

Only the LCA certified translation vendors can meet the legal translation quality. **Product teams who require localization for new legal content must coordinate and fund translations using certified vendors.** Before you start this process, work with your CELA representative to confirm the need for your new legal documentation and corresponding translations.

What is considered legal content

The following document types are legal content:

- End User License Agreement (EULA)
- Terms of Service (TOS)
- Terms of Use (TOU)
- Licenses
- Privacy statement
- THIRD-PARTY SOFTWARE NOTICES
- Any other content that requires CELA review or that poses a legal risk to the company.

For more information about identifying, scoping, or publishing legal content, see Legal content.

Legal localization process

Content owners are responsible for coordinating and funding the translation of legal topics as needed. Once translated, you submit a work item to Content Production Services (CPS) to merge and publish the translated files in each localized counterpart of the DocsLegal-pr repo. Use the following steps to request translations and publish your localized legal topics.

> **IMPORTANT**
> Before you start this process, make sure your updates have been published to the main (en-us) legal repo MicrosoftDocs/DocsLegal-pr. The localization repos are configured to automatically remove any files that are not mirrored in the en-us repo.

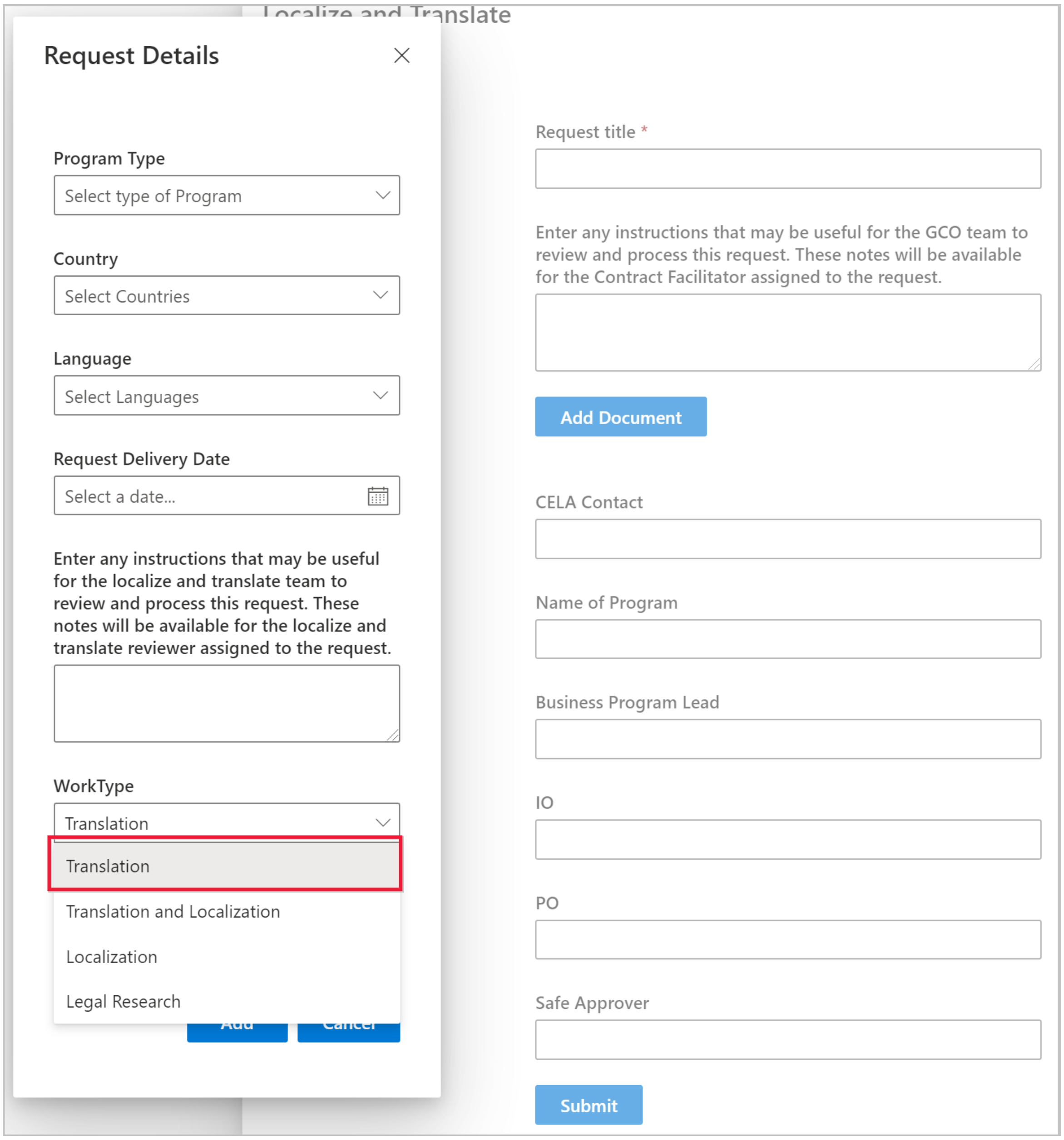
1. Confirm necessary locales for translation with your CELA representative. In some cases, there is no requirement to localize a legal topic.

2. Submit a localization request via the Legal Contracting Experience portal. This includes paying for the translation work.

   a. Under CONTRACTING HELP, select **Localize & Translate**.

   b. Complete the form fields.

   c. When you choose **Add Document**, the Request Details form appears. Complete the fields, specify the countries and locations, and for **WorkType** choose Translation.



3. When the translations are ready, use this template to create a work item for CPS. The template includes instructions to attach the translated files and confirm the locales and folder locations.

> **NOTE**
>
> The publication SLA is 48 hours. CPS will assume that this content is ready to go live when you create the work item.

4. The CPS vendor will commit the topics to each repo and then create a PR from master to live to validate the build before publishing. The vendor and/or the Onboarding team might contact you in the ticket or a PR if they need help with resolving an issue.

5. When the vendor completes the work item, review the published pages to make sure they render as expected.

If you have any questions, contact the CPS leads.

FAQ

- **Can I save legal documents in my product repository?**

  No.

  If you save legal documents in a product repo, such as sql-docs-pr, the current loc system automatically detects those files and queues them for localization by the technical document translation vendor. As explained above, the translation is not legally valid.

  Instead, you must host the legal document in the DocsLegal-pr repo. See Legal content for detailed requirements and publication steps.

- **Can I host all languages/localized versions of a legal document in the one DocsLegal-pr repo?**

  No.

  The publishing system (OPS) defines the language publish per repo. If the article is to be published in multiple languages, you must save each translated article in its respective localized repo. The good news is we have already set up 68 localized Docs-Legal-pr repositories.

- **Which languages should I target for localization?**

  Work with your CELA representative to confirm your legal localization needs. You may need to localize legal documents for the same languages that your product or service is localized.

List of localized DocsLegal-pr repos

There are 68 localized versions of the DocsLegal-pr repo:

- DocsLegal-pr.ar-sa - Arabic
- DocsLegal-pr.eu-es - Basque (Basque)
- DocsLegal-pr.bs-Cyrl-BA - Bosnian-Cyrillic (Bosnia and Herzegovina)
- DocsLegal-pr.bs-Latn-BA - Bosnian-Latin (Bosnia and Herzegovina)
- DocsLegal-pr.bg-bg - Bulgarian
- DocsLegal-pr.ca-es - Catalan (Catalan)
- DocsLegal-pr.zh-hk - Chinese-Hong Kong
- DocsLegal-pr.hr-hr - Croatian
- DocsLegal-pr.cs-cz - Czech
- DocsLegal-pr.da-dk - Danish
- DocsLegal-pr.nl-nl - Dutch
- DocsLegal-pr.nl-BE - Dutch (Belgium)
- DocsLegal-pr.en-AU - English (Australia)
- DocsLegal-pr.en-CA - English (Canada)
- DocsLegal-pr.en-IE - English (Ireland)
- DocsLegal-pr.en-my - English (Malaysia)
- DocsLegal-pr.en-nz - English (New Zealand)
- DocsLegal-pr.en-sg - English (Singapore)
- DocsLegal-pr.en-za - English (South Africa)
- DocsLegal-pr.en-in - English India
- DocsLegal-pr.en-gb - English-United Kingdom of Great Britain and Northern Ireland
- DocsLegal-pr.et-ee - Estonian (Estonia)
- DocsLegal-pr.fil-PH - Filipino (Philippines)

- DocsLegal-pr.fi-fi - Finnish
- DocsLegal-pr.fr-fr - French
- DocsLegal-pr.fr-BE - French (Belgium)
- DocsLegal-pr.fr-CH - French (Switzerland)
- DocsLegal-pr.fr-CA - French-Canada
- DocsLegal-pr.gl-es - Galician (Galician)
- DocsLegal-pr.de-de - German
- DocsLegal-pr.de-at - German (Austria)
- DocsLegal-pr.de-CH - German (Switzerland)
- DocsLegal-pr.el-gr - Greek (Greece)
- DocsLegal-pr.he-il - Hebrew (Israel)
- DocsLegal-pr.hi-in - Hindi
- DocsLegal-pr.hu-hu - Hungarian
- DocsLegal-pr.is-is - Icelandic
- DocsLegal-pr.id-id - Indonesian
- DocsLegal-pr.ga-IE - Irish-Ireland
- DocsLegal-pr.it-it - Italian
- DocsLegal-pr.it-ch - Italian (Switzerland)
- DocsLegal-pr.ja-jp - Japanese
- DocsLegal-pr.kk-kz - Kazakh
- DocsLegal-pr.ko-kr - Korean
- DocsLegal-pr.lv-lv - Latvian
- DocsLegal-pr.lt-lt - Lithuanian
- DocsLegal-pr.lb-L - Luxembourgish-Luxembourg
- DocsLegal-pr.ms-my - Malay
- DocsLegal-pr.mt-MT - Maltese-Malta
- DocsLegal-pr.nb-no - Norwegian (Bokmål)
- DocsLegal-pr.pl-pl - Polish
- DocsLegal-pr.pt-br - Portuguese (Brazil)
- DocsLegal-pr.pt-pt - Portuguese (Portugal)
- DocsLegal-pr.ro-ro - Romanian
- DocsLegal-pr.ru-ru - Russian
- DocsLegal-pr.sr-cyrl-rs - Serbian-Cyrillic-Serbia
- DocsLegal-pr.sr-latn-rs - Serbian-Latin-Serbia
- DocsLegal-pr.zh-cn - Simplified Chinese
- DocsLegal-pr.sk-sk - Slovak
- DocsLegal-pr.sl-si - Slovenian
- DocsLegal-pr.es-es - Spanish
- DocsLegal-pr.es-MX - Spanish (Mexico)
- DocsLegal-pr.sv-se - Swedish
- DocsLegal-pr.th-th - Thai
- DocsLegal-pr.zh-tw - Traditional Chinese
- DocsLegal-pr.tr-tr - Turkish
- DocsLegal-pr.uk-ua - Ukrainian
- DocsLegal-pr.vi-vn - Vietnamese

How do we select content to localize

1/22/2020 • 2 minutes to read

Selecting content for localization

There is no hard and fast rule to how the localization team chooses content to localize, and into which languages to localize the content. For each project, the MSFT loc PM is carefully evaluating before making a decision. The loc PM typically works closely with the content manager/writing team, to find out which areas of the content are most important (e.g. If we cannot localize the whole content set, should we localize getting started vs. Some other content buried deeper in the TOC?).

We take many factors into consideration; the most common are:

- **Legacy.** Was the product localized before? To which locales? We typically support a certain language set per pillar (e.g. all Azure-related content is localized to 17 locales)
- **BI.** Is the content used by many international customers? Should we expand/contract the language set?
- **Importance/Visibility of content.** In discussion with content owners, which areas of content are more important?
- **Software/UI.** Into which languages is the software/UI localized? Should we match the language set, or should there be more/less languages supported for content, and why?
- **Budget.** What can we afford, what tradeoffs can/should we make? There are basically 2 levers, which are the amount of files and amount of languages, plus the localization quality (human translation vs. Machine translation).
- **Tools.** We can support markdown and yaml file localization of files living in Github repos. If you have different file types and CMS, please contact the localization PM to discuss.
- **Related projects.** Are components of our content used by partner teams, e.g. Office/Windows? Do they have a different language set we need to support?

Reference content and Intellisense

- As a rule, **reference content is localized with TM/MT**. This means, the translation memory for a project is used (translation memory always contains human translated strings) to recycle matching sentences, and the non-matching sentences are then machine translated.
- The exception is Intellisense. For most projects where Intellisense is shipped as a downloadable or the Intellisense xmls are checked into the software build we will provide **human quality translations for the intellisense-strings,** depending on schedule and budget availability.
- For **new dev languages** (e.g. Python, java, ...) which we traditionally have not localized, there may be **advanced notice** and tooling changes required before we are able to provide localized content.

Images and Videos

Image and video are very costly to localize and require additional turnaround time from our suppliers, so any decision is made on a project-by-project basis. If you believe that some of the images/videos in your content are especially important to the customers, please advise the loc PM. As a general rule, we localize images and video subtitles for all Learn content, but only for subsets of technical documentation.

Source escalation for localization issues

11/2/2020 • 2 minutes to read

What is Source Escalation

When translator finds English issue, or Community Contribution requires English content to be fixed, localization team usually escalate the issue to English writer. This process is called as Source Escalation.

How to Escalate Source Issues

We have agreed with DevRel Content managers that localization team may escalate English issue or localization query to DevRel writing vendor team, or to the English **private** repository. Escalation to the English **public** repository only applies to projects where **private** repository doesn't exist.

The following process applies to all DevRel product repositories except SelfHelpContent, TechProfile and VS Recon.

Escalating Simple Source Issues

When source issues are simple and loc team knows how to fix them, Loc PM will create a PR in the default branch of the English repository. Loc PM verifies the fix on the staging site. If no error, then Loc PM will comment **#sign-off** and mention the article author.

- If the PRMerger is enabled on that repository, then **#sign-off** will kickoff the PR merge automatically.
- If the PRMerger is not enabled on that repository, then **#sign-off** will remind the repository admin to merge the PR.

Types of issues escalated as GitHub PR are limit to:

- English Typo: always confirm the word spelling from merriam-Webster dictionary
- Syntax error that blocks localization handoff or localization rending on docs site.

Escalating Complex Source Issues or Localization Queries

When Loc PM is not sure how the source issue should be fixed, or need a clarification to translate, then Loc PM will create a work item to DevRel vendor editorial team through DevRel loc-fix template.

Types of issues escalated as DevRel localization fix work item include but not limited to:

- Correctly understand an ambiguous sentence to certain language.
- Clarify the technical concept for a term, a sentence, or a paragraph.
- Point out the markdig format syntax error repro in multiple lines or multiple articles that need to be fixed.
- Fix uncertain grammar error.

Then vendor editorial team will review the ticket and make the necessary updates to the article. If the vendor editorial team cannot confidently provide a fix or an answer, then they will contact the listed author of the article.

The response to localization ticket SLA is 10 working days except MS holiday.

DevRel Localization Fix Work Item

- Always use DevRel loc-fix template to report complex source issue or localization query.
- Don't change 'Assign To', 'State', 'Area' and 'Iteration' value in the template.

- For every new work item, always modify '<summary description>' according to the real asking in the title before save.
- The ticket owner will change the 'State' from 'Proposed' to 'Committed', then 'Completed' or 'Cut'.
  - If the 'State' becomes 'Completed', then the 'Reason' will be either 'Done' or 'Duplicated'.
- Before filing the new escalation work item, always check the duplicated asking. If there is a similar asking, link to the 'old' work item.

Writing for translation

2/18/2020 • 2 minutes to read

How you write English source content can significantly affect the quality of translation. Simplified English is easier for non-native English speakers to understand, and translates better into different languages.

Here are some general guidelines on how to write for translation and help to improve the quality of both human translation and machine translation.

Simple guidelines

Fortunately, the most fundamental rules are easy to remember:

- Use short sentences.
- Use correct spelling, capitalization, and punctuation.
- Use precise language.

More guidelines

Here are more detailed guidelines for writing content that's easy to translate into other languages (that is, 'globalized' English):

- Split long sentences into several shorter sentences whenever it's logically possible.
  - Split compound sentences into two or more simple sentences.
  - Make a 'which' clause its own sentence.
  - For long instructions, consider restructuring the content into a bulleted list.
  - Short sentences should be typically about 15-20 words.
- Use standard English word order:
  - Subject + Verb + Object
  - Avoid inverting the subject and the verb (for example, 'Never have I...', 'No sooner had they...', and 'Not only did they...').
- Use simple verb tenses (simple present, simple past, and simple future).
  - Avoid the use of progressive verb tenses ('am/is/are', 'was/were', 'will be').
  - Avoid the use of perfect verb tenses ('have/has', 'had', 'will have').
- Don't use a weak 'to be' verb when you can use a stronger verb.
  - Example: 'Work will be finished' should be 'Work will finish'.
- Use active voice instead of passive voice.
  - Example: 'Work was completed' should be 'The team completed the work'.
  - Exception: It's fine to use passive voice to avoid assigning blame to a user, or when the actor is unknown or immaterial.
- Use commas correctly.
- Use semicolons sparingly. It's usually better to create two sentences.
- Don't use `-` or `/` as puncuation. For example:
  - "Run the app and verify you can create/edit/display the page." should be written as "Run the app and verify you can create, edit, and display the page."
  - "Run the app - verify you can create a page." should be written as "Run the app and verify you can create a page."

- Avoid parenthetical clauses.
- Use abbreviations or acronyms sparingly.
- Use words and phrases consistently.
- Don't use nominalizations.
  - For example, 'The rapid increase in incident tickets is becoming a concern' should be 'Incident tickets are rapidly increasing and becoming a concern'.
- Avoid noun stacks.
  - For example, 'Install the incident tickets monitoring app' should be 'Install the app to monitor incident tickets'.
- Avoid stacking prepositional phrases on top of one another.
  - Examples of prepositional phrases are here.
- Avoid wordiness and jargon, such as English clichés and slang.
  - Technical prose translates better than marketing-style language, jokes, or wordplay.
- Include all words that will clarify a sentence.
  - Include *that* and *who*. They help to clarify the sentence structure. For example, "Select the check box of each folder *that* you want to sync with your desktop."
  - Don't omit articles and other little words from parallel structures.
- Don't use possessives for product or feature names.
  - For example, don't use 'Azure's' or 'Spell Checker's'. 'Azure's security features include' should be 'The security features of Azure include'.
- Don't append s to make types/APIs plural.

Additional resources

- Microsoft Writing Style Guide for localization

Provide Feedback to DevRel Localization Team

4/2/2021 • 2 minutes to read

This article is for anyone who wants to provide feedback to the DevRel localization team. DevRel localization team uses CEINTL DevOps to track various issues and requests with their loc vendor. Anyone who has Microsoft alias can request for access to CEINTL DevOps and open a Feedback ticket. Feedback can be about translation quality or any suggestions for localization.

> **NOTE**
>
> The **external contributor** who doesn't have MS alias can download the email template by clicking DOCS Loc Feedback, then send the feedback through email. The feedback email will be saved as an ADO ticket in CEINTL DevOps. Our team will monitor and process the feedback.

Access to CEINTL DevOps

If you don't have access to CEINTL DevOps, then you can request for access to the "CEAPEX Work Items Access" project from MyAcess.

How to Fill-In Feedback Tickets

When opening a Feedback ticket, Use DevRel localization feedback template, or type aka.ms/devrellocfeedback in URL to create a new Feedback ticket providing the following information will help us to understand your feedback better and speed up the process:

- **Assigned To**: By default, Feedback tickets are assigned to *MS International Feedback Team* account, MS Feedback PM assignes the ticket to the right loc PM.
- **Description**: Provide the following information per suggestion:
  - Language for which the issue is found
  - Which Technology which the issue is found?
  - What is the URL which the issue is found?
  - Brief Description of the issue
  - Steps to Reproduce
  - Screenshot Attached (Yes/No)
  - Current Translation (if it's a Linguistic issue)
  - Expected Translation (if it's a Linguistic issue)
  - Corresponding English Text (if it's a Linguistic issue)
  - Comments to the localization team
- **Project Type**: Select if your feedback is for *Software*, *Content*, or *Sites* localization. If your feedback affects multiple Project Types, then select *Cross-Team*.
- **Issue Found By**: Select the source of your feedback. For example, if you're a Customer Relationship PM and you received feedback from an MVP, then select **MVP**.
- **Language Origin**: Select the language your feedback was originally found in.
- **Affected Languages**: Select all the languages affected by this feedback. This will helps us understand the scope of your feedback.

Create Feedback Ticket via Email

From time to time, you may receive feedback from your partners via email. In this case, we strongly recommend for you to forward those e-mails to CEINTL e-mail alias, as it will automatically create a Feedback ticket in CEINTL DevOps. Then you should receive a confirmation email from Mail2Bug bot within 2 minutes including a link to the Feedback ticket.

You can use mnemonics (@) and pound (#) characters to set fields in the Feedback ticket to be created:

> **NOTE**
>
> - As best practice, always include an example, or attach a screenshot for the feedback.

How to Avoid Over-Localization

11/2/2020 • 3 minutes to read

What is Over-Localization?

Over-localization is a localization bug when certain term or keyword that should be left in English for localized languages is translated by error. Example of terms or keywords that shouldn't be translated are:

- Product/service/branding names (for example, "Azure DevOps", "Xamarin.Forms")
- File names or folder paths.
- Code elements or programming languages keywords such as:
  - Class/function/property names (for example, "CAccelerateDecelerateTransition")
  - Predefined macros (for example, "__cplusplus_cli")
  - C# keywords/operators (for example, "static" or "default")
  - HTML/XML tags (for example, "<element>")

Over-localization may happen as human error for human-translated (HT) topics. For machine-translated (MT) topics, there's limitation of the MT engine. Root cause can also be considered as source issue in English content. As an example, if a programming language name "Python" isn't properly capitalized like "python", then that typo could affect both HT and MT. Both human translator and MT engine may translate it as a type of snake instead of the programming language. Content writers can help to avoid these issues by using certain Markdown syntax to block terms/keywords from being over-localized.

How to Avoid or Fix Over-Localization

Here are current solutions in which content writers can help to avoid over-localization issues:

- For new product/service/branding names, or keywords that shouldn't be translated, contact your localization PM in advance before publishing the content. Loc PMs will make sure those terms are added to Term Studio with approved term to use for each language.
  - For HT topics, human translators will reference the approved terms in Term Studio.
  - For MT topics, terms added to Term Studio will be added to MT dictionary, which is then used to retrain the MT engine on quarterly basis.
- For new and existing product/service/branding names, be sure to use proper capitalization so they're correctly recognized.
- Use inline code style or code blocks for programing language keywords, file names, or folder paths.
- Use the no-loc Markdown extension inline or using metadata in YAML header.
- Add the loc-scope attribute to screenshots with a different localization scope than the article or module that contains them.
- For HTML/XML tags (for example, <element>), feature was deployed as of June 2020 to protect these escaped <element> tags from being over-localized.

> **NOTE**
>
> Due to this feature, placeholder text within angled brackets (for example, <placeholder text>) will be blocked from localization. Placeholder texts in angled brackets can be considered part of code examples, which we do not localize. However, in order to have placeholder text localized, writer can use Italic font instead of angled brackets as recommended in the style guide.

How to Report Over-Localization Issues

When you find any over-localization issues, first consider one of the options listed above on how to avoid or fix them. If you have any questions about those options, then you may directly contact your localization PM. If you receive GitHub ISSUE from customer in English repo for over-localization issue, then you may use @ mention to notify your localization PM for assistance. Otherwise, you can also submit any feedback for localized content by opening a Feedback ticket.

> **NOTE**
>
> For HT topics, you may open GitHub PR or ISSUE in the affected localized repo. However, Community Contribution (GitHub PR) or Feedback feature (GitHub ISSUE) are currently not accepted for MT content. MT topics have a disclaimer stating that part of the topic may be machine translated, and the **Edit** button won't be displayed as we have disabled it for MT topics. We would still welcome any internal feedback on MT over-localization issue, so please don't hesitate to either contact your localization PM or open Feedback ticket.

Related topics

- Get to know the Global Experiences Content Localization team
- Frequently asked questions (FAQ) about localization
- DevRel Authoring Checklists for Localization
- Provide Feedback to DevRel Localization Team

How to specify the localization scope for an image

3/10/2021 • 2 minutes to read

The localization scope is the set of locales a product or content set is localized into. For example, most Azure services and their corresponding documentation are localized into 17 languages, but Visual Studio is localized into 13 languages.

Sometimes an image in an article or Learn module shows a product or service that has a different localization scope than the main subject of its parent. For example, you might include a screenshot of Visual Studio in a module about Azure. Most of the content of the module should be localized into Azure's 17 languages, but the screenshot should only be localized into the 13 languages supported by Visual Studio. Otherwise, some international users would see misleadingly localized screenshot.

To prevent the over-localization of images, use the `loc-scope` attribute on the `:::image:::` extension for any image with a different localization scope than the content that contains it. This is particularly important for Learn modules, which often include screenshots from multiple products.

loc-scope syntax

The `loc-scope` attribute can be applied to any `:::image:::` with `type="content"` (standard conceptual images such as screenshots) or `type="complex"` (complex images that require a detailed long description, such as graphs and architectural diagrams). The `loc-scope` attribute doesn't apply to images of `type="icon"`. Icons are purely decorative and aren't localized.

The value of `loc-scope` must be one of the following:

- A valid product per the Product taxonomy allowlist.
- `third-party` for non-Microsoft products, such as Swagger.
- `other` for images that don't apply to a particular product, such as output screens for example apps.

When should I add a non-default loc-scope?

You should specify a `loc-scope` that differs from the default if your image is a screenshot of a different product. For example, if you're showing a VS Code screenshot in an Azure module, you should specify `loc-scope="vs-code"`. For conceptual images, such as a workflow diagram that shows an Azure task with steps involving VS Code, use the Azure default, `loc-scope="azure"`.

Adding loc-scope with the Docs Authoring Pack

The Docs Authoring Pack makes it easy to add images with `loc-scope`.

By default, the `loc-scope` attribute isn't included when you insert a new `:::image:::`, because it's optional for most Docs content. To add `loc-scope` to an image, follow these steps:

1. Put your cursor inside the `:::image:::` tag.

   ```
   :::image type="content" source="../../archive/contribute/archive/media/contribute-get-started-git-and-github/fork.png"
   alt-text="Illustrates forking a GitHub repo." :::
   ```

2. From the Docs Markdown menu (**Alt + M**) or the VS Code command palette (**F1**), select **Add localization scope to image**. The valid product list opens.

3. Type to filter the list if necessary, then select the product that applies to the image, or `third-party` or `other` if appropriate. The selected `loc-scope` is added:

```
:::image type="content" source="../../archive/contribute/archive/media/contribute-get-started-git-and-github/fork.png"
alt-text="Shows forking a GitHub repo." loc-scope="third-party":::
```

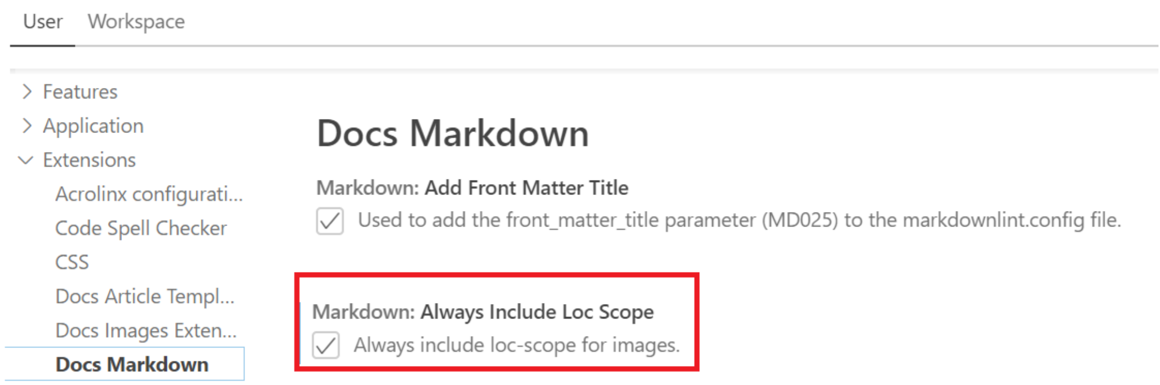
Instead, you can update your VS Code extension settings to always include `loc-scope` in images. This is recommended if you frequently work in Learn repos.

1. From the VS Code **File** menu, select **Preferences** then **Settings**.

2. On the **Settings** page, choose **User** to update your settings for all workspaces, or **Workspace** for just the current workspace (this is useful, for example, if you just want to always include `loc-scope` when working in Learn repos).

3. Under **Extensions**, click **Docs Markdown**.

4. Check the box for **Markdown: Always Include Loc Scope**.



Now, whenever you insert an `:::image:::`, you'll be prompted to select a `loc-scope` from the product list. You can still add `loc-scope` to existing images by selecting **Add localization scope to image** from the VS Code command palette (**F1**).

Request localization of video captions

11/2/2020 • 2 minutes to read

The Developer Relations Global Experiences team localizes *video captions* only. If you want to localize audio content, contact the DevRel Global Experiences Loc Pillar PM.

Video management platform

The Microsoft Docs engineering team plans to provide a new video management platform, which could potentially replace the current video player and video management platform. For now, we primarily localize videos on the *RedTiger* platform. For localizing videos on any other platform, contact the DevRel Global Experiences Loc Pillar PM.

RedTiger can upload and publish video files with video captions.

- When a video is uploaded as a new asset, it gets a new VideoID as an unique identifier. Each VideoID is associated with multiple elements including the caption files and the language setting. For example, if a RedTiger subsite is configured for ten languages, then any video uploaded to this subsite can have ten language options. One caption file can be uploaded for each language. However, the video file (.mp4) stays the same across ten languages.
- When a video is uploaded to an existing asset, then the VideoID retains its existing properties. This is commonly used when the video or video caption is updated.

Currently, our loc pipeline doesn't detect the Markdown syntax for embedded videos. So we won't know when a new video is embedded in your article. To start localization for video caption, we temporarily need the video owner to create a loc request ticket in a manual way.

Request for caption localization

For any request to localize a video caption, create an Azure DevOps ticket as described here. The loc team will use the Azure DevOps ticket to communicate with the requestor, track any issues found during production, and monitor the loc progress.

Use the Video Captioning Request Template to create a loc request ticket. If you can't access the link, then join the "DevOps_CEAPEX_Partners" security group in IDWeb.

In the ticket, fill out the following information. The first four items are mandatory to start localization.

- VIDEO ASSET ID: Enter the asset ID from RedTiger.
- LANGUAGE LIST: If it's the first time you're requesting video localization for your team, list the languages you want the captions localized in.
- COMPONENT NAME: Enter a product name, such as Azure or Power BI. If you're unsure, contact your localization PM.
- ENGLISH CAPTION: If the English caption files are available, attach them to the ticket. If the files aren't available, see Guidelines for English video captions.
- VIDEO URL: Enter the public URL from RedTiger. It will help us test the video on the video player.
- DOCS URL: Enter the article URL where the video is embedded. It will help us verify that all Docs live pages contain valid localized video.

The loc team will validate the information on RedTiger and contact you with any questions. Localization will start after all questions are answered. The loc team will also estimate the completion date and add it in the **Due Date** field in the ticket.

Caution

The VideoID doesn't indicate any change to video element, so loc team will not know when video is updated. For updated videos, create a *new loc request* for loc team to localize the updated video captions.

After a localized caption is published to the video, the loc team will attach the localized caption and resolve the ticket back to you.

Guidelines for English video captions

3/5/2021 • 2 minutes to read

The localization (loc) team localizes only video captions and not the speech audio, so English video captions are required to localize videos. The English content team should provide the caption files.

Guidelines for captions

The following guidelines are for the English content team or supplier to create English video captions. The loc team will check that these guidelines are followed before localizing the captions. The first three points are required, so the loc team will reject a caption if those aren't followed. The loc team can also help create or correct a caption if requested.

- The caption file is in .ttml or .vtt format. However, Red Tiger only supports .ttml format.
- The caption includes running time.
- There are no midsentence breaks in one time label. (Each caption needs to be a complete sentence.)
- Leave 30% space in the time label for translation wherever possible. (The shorter the string is, the more likely the translated string is to expand.)
- All technical terms are correctly spelled out.
- There are no colloquial terms or phrases.

Guidelines for video creation

The following guidelines are for creating English videos, which helps the localization of video captions:

- The narrator speaks slowly and clearly, for better speech-to-text conversion and to reduce the manual effort needed for editing captions.
- On average, leave 30% additional pauses in the spoken script to accommodate expanded length from localization.
- Avoid colloquial terms and phrases, such as "to be honest" and "frankly speaking."
- Provide scripts to the loc team when you're recording audio, if available.
- Place related videos in one folder (based on product) that the loc team can access.
- Provide a glossary file (any file format works), which contains at a minimum:
  - Technology or product of the video content, such as "GitHub" or "Intune."
  - Product or technology-specific terminology, such as service names.
  - Name of the speaker or event, such as "Scott Guthrie" or "Build."
- Remember that uploading a video takes about six times longer than the video duration. For example, a 2-minute video will take about 12 minutes to upload.

  Contact the RedTiger team before you upload a large number of videos in a short time frame.

Aliases for Azure technical content

4/6/2021 • 2 minutes to read

Frequent contributors to the Azure technical documentation need to belong to certain aliases and be aware of others for information sharing.

ALIAS NAME	ALIAS	REQUIRED?	PURPOSE
Azure technical content authors	wacom-tech-authors	Yes	All contributors to the private repo should join this alias. It is used to communicate publishing status, changes to content strategy and process, and other official updates for Azure technical content. The purpose of the alias is outbound communications only. Only authorized senders can mail to this alias.
Azure docs publishing notifications	azure-docs-publish	No	Receives automated publishing notifications for the azure-docs-pr repository. You need to create a mail rule to filter for items where the subject or body contains the text 'GitHub Branch Name: live' - these are the live publishing runs.
C&L Pull Request Review Team	techdocprs	Only for pull request reviewers.	Use this alias to contact the group of people who review pull requests for Azure tech docs.

Azure documentation feedback using GitHub issues: process for CXP forum moderators

5/7/2021 • 8 minutes to read

The Content & Learning technical content team and program management content owners will collaborate with the Azure Customer Experience (CXP) team to respond to customer feedback for Azure technical documentation.

- Azure CXP will triage all new issues in the azure-docs repo.
- Azure CXP will make their best effort to respond to and close the issues.

  When the CXP team needs help responding to the feedback, they'll assign the issue to the author who's listed in the article's metadata.

The Azure implementation aligns to the GitHub issues documentation feedback policies and guidance. However, because of the partnership with Azure CXP and because of the large scope of the Azure repo, there are specific processes for Azure. This article describes the specific implementation of the new documentation feedback experience for the Azure content set. The team will incorporate the lessons learned into the general guidance as appropriate.

Repositories covered by this process

Azure CXP manages initial triage and also provides issue response and deflection in the following Azure content repositories:

- architecture-center
- azure-cli
- azure-docs
- azure-docs-cli
- azure-docs-powershell
- azure-sphere-issues
- azure-stack-docs

Process for CXP forum moderators

1. Go to the azure-docs Issues page.
2. Run this query:
   - *https://github.com/MicrosoftDocs/azure-docs/issues?q=is%3Aopen+is%3Aissue+no%3Alabel*

   Or, use this query in the issues search:

   - *is:issue is:open no:label*
3. Review each untriaged issue:
   a. Add the **triaged** label.

   b. Assign the appropriate service label to the issue.

   c. Take the appropriate next step, as defined in the table in the next section.
4. Daily, run this query to find issues that an author has indicated can be closed, review, and if appropriate,

close:

- *https://github.com/MicrosoftDocs/azure-docs/issues?utf8=%E2%9C%93&q=is%3Aissue+is%3Aopen+%23please-close*

Or, use this query in the issues search:

- *is:issue is:open #please-close*

5. Daily, run this query to find issues that an author has marked as in process, and assign the in-process-author label:

   - *https://github.com/MicrosoftDocs/azure-docs/issues?utf8=%E2%9C%93&q=is%3Aissue+is%3Aopen+%23in-process*

   Or, use this query in the issues search:

   - *is:issue is:open #in-process*

6. Daily, run this query to find issues where an author or content team has @mentioned the Azure CXP triage team:

   - *https://github.com/MicrosoftDocs/azure-docs/issues/mentioned/azure-cxp-triage*

   Or, use this query in the issues search:

   - *is:issue is:open mentions:azure-cxp-triage*

Process and labeling table

If an issue is assigned to, answered, or resolved by CXP, add the **cxp** label. *Don't* add the **cxp** label if the CXP triager is closing an issue resolved by a PM or author.

ACTION	ASSIGN LABEL(S)	NEXT STEPS	EXAMPLE/NOTES
CXP can respond to the question related to the document or product.	**cxp**, **in-progress** Add the **product-question** label to classify the issue as a product question.	Provide the answer to the user's inquiry. If there's no further questions, close the issue	
CXP has identified the user's problem is related to an issue with the product.	**cxp**, **in-progress** Add the **product-issue** label to classify the issue as a product issue.	Work with engineering to check when they'll address the product issue or if there's a work-around available. Inform the user of the issue status and if they're satisfied with the reply, close the issue.	

ACTION	ASSIGN LABEL(S)	NEXT STEPS	EXAMPLE/NOTES
CXP can respond to the doc feedback and fix immediately.	**cxp** Add the **doc-bug** or **doc-enhancement** label to classify the doc feedback issue.	If the feedback is about something that is out of date, unclear, confusing, or broken in the article, assign the **doc-bug** label. Classify feedback about missing information that is blocking the customer's success as **doc-bug**. If the feedback suggests additions or improvements to the article, but there's no evidence the info in the article blocked the customer, assign the **doc-enhancement** label.	
CXP can fix by editing the document, but must do some work first	**in-progress**, **cxp** Add the **doc-bug** or **doc-enhancement** label to classify the doc feedback issue.	Finish the work required. When you've finished and merged the work, provide confirmation in the issue and close. Add the **cxp** labels.	Example: CXP files a pull request to address the issue. Close the issue after the PR is merged.
CXP needs help from product team to fix	**in-progress**, **cxp**, **awaiting-product-team-response**	After an initial response setting expectations with the customer, follow up with service teams similar to the process for the forums. After you've provided the answer, remove the **in-progress** label and the **awaiting-product-team-response** label, and add **cxp** label, then close.	
CXP can't fix and needs help from content author	**assigned-to-author**, and **doc-bug** OR **doc-enhancement**	In GitHub, assign the issue to the listed article author. If the feedback is about something that is out of date, unclear, confusing, or broken in the article, assign the **doc-bug** label. Classify feedback about missing information that is blocking the customer's success as **doc-bug**. If the feedback suggests additions or improvements to the article, but there's no evidence the info in the article blocked the customer, assign the **doc-enhancement** label.	

ACTION	ASSIGN LABEL(S)	NEXT STEPS	EXAMPLE/NOTES
CXP determines the issue violates Microsoft's code of conduct	**code-of-conduct**	Enter a comment indicating the issue is spam, close the issue, and then lock it to remove it from the article comments.	
CXP determines the issue is product feedback	**product-feedback**	Enter a comment for the customer indicating it's product feedback, provide a link to Azure User Voice or the appropriate feedback channel, and close the PR. **Note:** Product questions that CXP can answer will be answered immediately and not redirected to Azure User Voice	**Sample response:** "Thanks for taking the time to provide feedback on Visual Studio! This feature idea is interesting and we want to make sure it gets directly into the hands of the product team. Fortunately, we've got a dedicated channel for that very purpose! You can find it over here. We'd be happy to submit your feature request for you, if you like, but we want to give you the opportunity to submit it directly, if you'd like to receive notifications, vote, or monitor progress more closely. Let us know what you prefer. If we don't hear back from you in the next few days, we'll happily submit it on your behalf and link the created request here."
CXP reviews issue and determines that a support ticket is required to continue	**support-request**	Enter a comment for the customer to open a support ticket if they have a support plan using Azure support. If not, ask them to send an email to Azure CPX Community Connect with a link to the GitHub issue and their subscription ID. Work with the customer to file the support request, and close the issue after you get the SR number. If there's no response from the customer when you ask for the subscription ID within 48 hours, close the issue. Let the customer know they can reopen the issue at any time.	

ACTION	ASSIGN LABEL(S)	NEXT STEPS	EXAMPLE/NOTES
The issue doesn't contain enough information for CXP to triage it.	**needs-more-info**	Assign the issue to the contributor and ask for more info. What you ask for will depend on the issue contents. What service is this issue about? Is the question or problem with the documentation or with an Azure product itself? Once you have sufficient info, remove the needs-more-info label and triage per the other triage instructions.	
The issue is a request to collaborate on or contribute new documentation	**doc-idea**	Assign the issue to GitHub user **msmbaldwin** for secondary triage to content teams.	
The issue has positive feedback	**kudos**	Thank the contributor, then close the issue.	
The issue has feedback about localized content or translation quality	**loc**	Assign the issue to GitHub user **Sara-Nicolini** for triage to localization teams.	
The issue has feedback about the docs.microsoft.com experience or design	**docs-experience**	Assign the issue to GitHub user **Powerhelmsman** (Jian Dong) for secondary triage.	
CXP determines the question is best answered in a docs.microsoft.com Q&A forum or StackOverflow forum	**forum**	Enter a comment redirecting the customer to the appropriate forum for their question.	
CXP can provide a documentation or blog link that directly answers the question.	**doc-provided**	Enter a comment with a link to the existing documentation that answers the customer question.	
The customer enters a follow-up comment that says they've solved the problem, and it's clear there's no follow up necessary	**resolved-by-customer**	Thank the customer and close the issue where there's no needed fix in docs or the product.	

Miscategorized issues

If an author sends an email to Azure CXP Doc Feedback to report that an issue has been incorrectly categorized as a documentation-related issue, follow these steps:

1. Assign the **category-reassigned** label to the issues.
2. Remove the incorrect category label, and assign the correct new one.
3. Triage to the appropriate new assignee, and remove the author from the list of assignees.

Metadata

The triage team can assign issues only to the people listed in the metadata, in this priority:

1. The person listed in the **author** attribute.
2. The person whose Microsoft alias is in the **ms.author** attribute.
3. The person whose Microsoft alias is in the **ms.manager** attribute, if that person is a content publishing team manager.
4. The content publishing manager accountable for the service - see the Business approver list for Azure services.

To look up a user by their GitHub or MSFT alias, use the search feature in GitHub for Open Source - People.

Prevent assigning issues to people who are no longer employed at Microsoft

Each member of the Azure CXP triage team should install the Chrome extension. It lets you see in the GitHub UI whether a GitHub user is a Microsoft employee or not:

1. To install, go to Personal access tokens - Extension to install the extension.
2. Go to Personal access tokens - Create to generate a token.
3. Right-click the extension icon on the **Chrome** toolbar, and select **Options**.
4. Enter the token in the extension.

If the listed author of an article is no longer shown in the Chrome extension as an employee, assign the issue the content manager for the service. For names, see the Business approver list for Azure services.

Handling legal issues

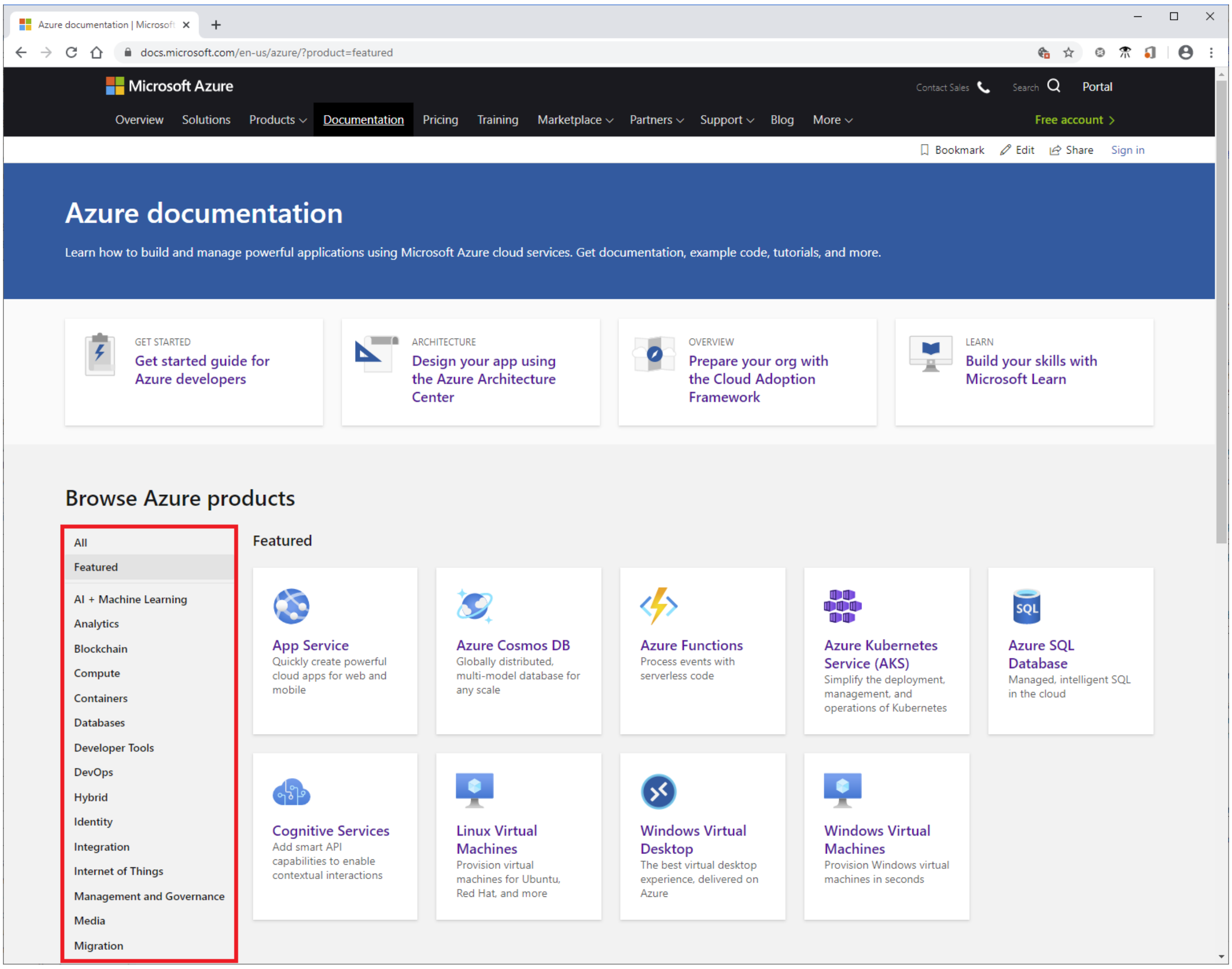
If an issue could have a potential legal implication for Microsoft, immediately escalate the issue to Azure CXP Doc Feedback. Legal issues include but aren't limited to issues like:

- Plagiarism: an issue mentions that our content is a copy of someone else's content or mentions the word "plagiarize".
- A report of any sort of misconduct, privacy issue, bias, or discrimination.

Update Azure docs hub page

4/16/2021 • 3 minutes to read

The Azure docs hub page is the entry point for all official Azure documentation. This article describes the steps to update the Azure docs hub page.



What appears on the Azure docs hub page?

The intent of the Azure docs hub page is to help customers understand the breadth and depth of documentation available for Azure. The Azure docs hub page also has a strict interlock with Azure.microsoft.com (ACOM) products to provide consistent messaging between these sites.

Here are the links that **can** appear on the Azure docs hub page:

- Product or service documentation that also appears on ACOM products
- Key programming language and tool documentation (maximum of 12)
- Sovereign cloud documentation

Here are the links that should **not** appear on the Azure docs hub page:

- Product or service documentation that is not listed on ACOM products
- Marketing content
- Links outside of the docs.microsoft.com domain (the one exception is https://code.visualstudio.com/docs)

Steps to update Azure docs hub page

1. Verify the service or product has been added to (or removed from) ACOM products.

2. If ACOM products do not reflect the needed changes, submit a request to update ACOM products.

3. Create a work item in mseng to request an update to Azure docs hub page and assign to Robert Lyon:

   https://mseng.visualstudio.com/TechnicalContent/_workitems/create/Task

   Alternatively, you can create a pull request to update index.yml in MicrosoftDocs/azure-docs-pr to match ACOM products.

   > **NOTE**
   >
   > ACOM products and Azure docs hub page must match. Before changes to the Azure docs hub page can be published, the changes must be published first on ACOM products.

The pull request reviewer will contact the designated business approver to request approval.

productDirectory cards

The Azure docs hub page is the index.yml file in MicrosoftDocs/azure-docs-pr. Each card in `productDirectory` has the following format:

- `summary` - Should match text on ACOM products.
- `imageSrc` - To keep file size small, should be a .svg file.
- `azureCategories` - One or more Azure categories. (Azure products can appear in more than one category)

```
# Card
- title: cardtitle
  summary: cardsummary
  imageSrc: svgimage
  azureCategories:
    - category1
    - category2
    - category3
  url: file OR https://docs.microsoft.com/file
```

Here is an example card:

```
# Card
- title: Azure Cosmos DB
  summary: Globally distributed, multi-model database for any scale
  imageSrc: https://static.docs.com/ui/media/product/azure/cosmos-db.svg
  azureCategories:
    - featured
    - blockchain
    - databases
    - iot
  url: cosmos-db/index.yml
```

Icons

If you need to add an icon to a card, make sure it is a .svg file. You can get icons from one of the following sources:

Docs shared Media (preferred)

- Preview page: https://static.docs.com/ui/media/index.html
- Repo: https://github.com/MicrosoftDocs/media/tree/master/src/product/azure
- Example imageSrc format: https://static.docs.com/ui/media/product/azure/file.svg

Docs.microsoft.com Shared Image Gallery

- Preview page: https://review.docs.microsoft.com/en-us/content-production-service/internal/image-gallery?branch=master#pivot=common&panel=logos
- https://github.com/MicrosoftDocs/DocsMedia/tree/master/media/logos
- Example imageSrc format: https://docs.microsoft.com/media/logos/file.svg

Local media folder

- Repo: https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/media/index
- Example imageSrc format: ./media/index/file.svg

Azure Icons Repository

- Preview page: https://cds-icons.azurewebsites.net/

Azure architecture icons

- Download page: https://docs.microsoft.com/azure/architecture/icons/

Azure category values

For `azureCategories`, you can use one of the following values. You cannot use any other values.

- ai-machine-learning
- analytics
- blockchain
- compute
- containers
- databases
- developer-tools
- devops
- featured
- hybrid
- identity
- integration
- iot
- management-and-governance
- media
- migration
- mixed-reality
- mobile
- networking
- security
- storage
- web
- windows-virtual-desktop

How is Azure docs kept in sync with ACOM products?

Approximately once a month, a script is run to detect differences between the Azure docs hub page and ACOM products. Updates are made to keep Azure docs hub page in sync with ACOM products.

Troubleshoot

When you update the Azure docs hub page, you might encounter errors. This table describes some issues and the corresponding solution.

EXAMPLE ERROR MESSAGE	SOLUTION
Unable to load file 'articles/test/index.yml' via processor 'Hub': Schema validation failed. Please validate the file and make sure it conforms to schema 'Hub' (https://static.docs.com/ui/latest/schemas/Hub.schema.json):	You are most likely trying to create a category hub page outside the predefined location (azure-docs-pr\articles\index.yml). This scenario is not supported.

Questions

If you have questions or problems related to the Azure docs hub page, send email to:

- docshub@microsoft.com

Related links

- Hub page schema
- Create or update a hub page

Azure subscriptions for Content and Learning organization (C+L)

5/10/2021 • 4 minutes to read

Employees IN C+L organization

The Azure internal registration and subscription (AIRS) setup information in this article is **only for people in Content & Learning (those who report up to Jeff Sandquist)** who don't want to or can't use the Visual Studio Subscription benefit.

To use your Visual Studio Subscription benefit for Azure account, go to https://my.visualstudio.com to activate your benefits.

Employees OUTSIDE C+L organization

If you can't use the Visual Studio Subscription benefit and you're in a group other than Content & Learning (C+L), contact your manager about setting up an Azure account through the internal registration system.

Set up Azure Subscription for a Content & Learning author

Internal Azure subscriptions are available for content developers in the Content & Learning organization. With an internal subscription, you don't need to use your Visual Studio subscription, your personal email, or your credit card in order to work with Azure features.

These subscriptions are charged back to Content & Learning. In other words they cost $$.

- DO NOT Use services which use a lot of processing power for a long period of time.
- DO NOT Leave a service running if you are **not actively** using it - Shut it down.
- DO NOT Run any production application under your test subscription. You can use your Visual Studio subscription for that.

Vendors: Have your Microsoft manager use the steps below to create a registration and subscription using their FTE alias. They should complete step 3 to add you to the subscription in the Azure portal. Due to the temporary nature of vendor assignments, this process is recommended by AIRS support to simplify tracking and management of accounts.

Step 1: Register for an Azure subscription (FTE)

Before you register for an Azure subscription, check with your manager to determine what codes you should use for cross-charging.

The following steps call out the fields that you need to update. You don't need to address all of the fields on the form.

1. Go to the Azure Internal Registration System (AIRS). You should have access, but if you don't, request access through the Azure Business Desk.
2. Click **New Registration**.
3. **Account Type**: Internal Billable Account.
4. **Account Owner ID**: Use your Microsoft account for your ID (FTE).
5. **Property/Service Name**: Azure Content Development
6. **PC Code for cross charging**: "P" + [your cost center number]

7. **PFAM for cross charging**: 000
8. **Program name**: C+AI Content & Learning Transform the Datacenter
9. **Paid support**: Developer
10. **Finance contact**: Mischelle Schimelpfenig <mischime> (Another backup contact in event of OOF: Alex Lejeune <allejeun>)
11. **Budget approver**: Alias of budget approver (your immediate manager can provide approval)
12. **Will a production application be hosted under this account**: No

Some Azure features require that you request access - you can usually do so on the preview page, or through the appropriate product team.

Step 2: Create your first subscription

1. After your initial request is approved, you receive an email message - click the activation link in the email to visit the AIRS site again.
2. Under **Purpose**, select the option for a personal subscription used for learning, support, projects, and sandbox work.
3. Under **Billing**, select **Provide a PC code that isn't showing in the list below**.
4. In the **PC Code** box, paste the profit center information you obtained from your business manager.
5. Click **Next**.
6. You'll be redirected to another internal consumption site, where you fill out another form with your personal details.

Step 3: Rename your subscription

AIRS names all new subscriptions with the same name. To make life easier for your business manager and budget team, rename your subscription so the team that owns it and the purpose of the subscription are obvious.

1. Sign in to the Account portal with your Microsoft account.
2. Click **Cost Management + Billing** in the left navigation pane.
3. Under **My subscriptions**, click the new subscription.
4. On the next page, click **Rename**.
5. In the **Subscription name** box, type a more descriptive name that makes it clear what team owns the subscription and what the subscription is used for.
6. Click **Save**.

Step 4: Onboard to 'Standard' Azure Security Center.

Ensure your subscription security setting is marked as 'Standard'; changing from default setting of 'Basic' in the Account portal. All our subscriptions must have this security setting in place to protect the C+AI fleet from unauthorized/anomalous activity. More information: (https://docs.microsoft.com/azure/security-center/security-center-get-started/)

Optional step 5: Add vendor to subscription

As an FTE, use this procedure when you've created a registration and subscription under your alias but it's specifically for vendor use.

1. Sign in to the Account portal with your Microsoft account.
2. Click **Cost Management + Billing** in the left navigation pane.
3. Select **Subscription**.
4. Under **My subscriptions**, choose the subscription you've created specifically for the vendor to use.
5. Select **Access control (IAM)** > **Add**.
6. For **Role**, select **Contributor**.
7. In **Assign access to**, select **Azure AD user, group, or application**.

8. In **Select**, type the vendor's name.
9. Click **Save**.

For more information about AIRS, see AIRS and Sponsorship FAQ.

Sovereign clouds

5/10/2021 • 2 minutes to read

This article summarizes sovereign clouds and the documentation process for them.

What sovereign clouds are

Sovereign clouds are physically isolated instances of Azure dedicated to the needs of a country, region, or government. They allow these customers to move sensitive services and apps to the cloud by addressing specific regulatory requirements. They're an important compete effort with major cloud providers with an estimated revenue of $29 billion by 2022.

Current sovereign clouds

Currently there are three sovereign clouds:

- Azure Government
- Azure Germany
- Azure China 21Vianet (***21Vianet** is the company that independently operates Azure in China.*)

Technical documentation

- **What**: *Only technical differences between the sovereigns and global Azure (Microsoft Azure) are documented in the sovereign docs*. Sovereign docs link to global Azure for info common to both. We don't add sovereign info to global Azure docs. Examples: Azure Government security and Azure Government identity.
- **Where**: There's a hub page on Docs for sovereign clouds. There are individual landing pages for Azure Government, Azure Germany, and Azure China 21Vianet.

Who does what

- **Global Ecosystems**: Owns sovereign clouds. They're part of Azure engineering. They onboard engineering service teams that have sovereign cloud offerings to relevant processes. The Global Ecosystems team creates some documentation, and they onboard Azure service PMs to do the rest.
- **Azure engineering**: After onboarding with Global Ecosystems, service PMs write or work with their content teams to write the documentation for their sovereign cloud offerings.
- **Marketing**: Owns the marketing pages for sovereigns on azure.microsoft.com. The pages include info on compliance, service availability, and pricing.
- **Content & Learning**: Content & Learning doesn't write sovereign cloud technical documentation but contributes in this way:
  - **Hosting**: Hosts sovereign technical content on Docs.
  - **Updates**: Updates engineering and Global Ecosystems on significant changes to the Docs platform and document conventions (ex. new content model).
  - **Support**: Provides ongoing support through the Docs contributor guide and the Azure Technical Content Chat alias (azcontentchat@microsoft.com), a forum for Q+A about the authoring process and tools.
  - **Contacts**: Monica Rush (monicar) and Femila Anilkumar (femila)

See also

The Azure Government marketing site on azure.microsoft.com.

> **NOTE**
>
> Azure Germany and Azure China 21Vianet don't currently have marketing pages on azure.microsoft.com.

New Azure service checklist

5/14/2021 • 3 minutes to read

If you're a technical writer and you need to create content for a new Azure service, use the following checklist to make sure you do all the required tasks:

☑	TASKS
☐	Identify the lead PM and sponsor for the new service.
☐	Keep your manager informed about the new service content progress.
☐	Identify the expected timeline for the release.
☐	Work with your product team and marketing to determine the service slug value. The service slug is the part of the docs.microsoft.com URL and the azure.microsoft.com URL that identifies your service, and the slug is used for the folder name, and metadata value `ms.service`. For example, `virtual-machines` is the service slug for Azure Virtual Machines and is consistently used between the docs site at https://docs.microsoft.com/azure/virtual-machines/ and the ACOM site https://azure.microsoft.com/services/virtual-machines/.
☐	Create a new release branch for your new content.
☐	Create a new folder for your service's docs using the new service slug value. Create the folder locally, and add your first files. Then the folders are "created" in GitHub.com when the first file is added and pushed into the online repository. For example, this folder matching the service slug is used for the Azure Virtual Machines docs: `https://github.com/MicrosoftDocs/azure-docs-pr/tree/master/articles/virtual-machines` .
☐	Create the new service metadata value. Use the new ms.service value request form to add your service metadata to the list. Make sure to select **Yes** to include your service docs metrics in the content engagement reports. The `ms.service` metadata property typically uses the same value as the service slug.
☐	Work with your product team to determine content plan and create new articles.
☐	Get the new service .svg icon from the marketing representative for your service.
☐	If available, copy a related service toc.yml file into your service folder and edit for your new service.

☑	TASKS
☐	If available, copy a related service index.yml (landing page) into your service folder and edit for new service.
☐	Determine if the service is a public preview or general availability. For a preview service, refer to the Identify public preview content article, and make the necessary changes in the TOC and landing page.
☐	Based on your content needs, list the articles that you need to create. Review the recommended article types for Docs.
☐	Based on your article list, define the TOC structure to develop the toc.yml file. If a related toc.yml file is available, copy it into your new content folder and change it based on your article list. Use the TOC checklist spreadsheet to draft your TOC.
☐	Ensure your articles align with the docs.microsoft.com writing and technical principles checklist.
☐	Understand and incorporate SEO techniques in your content.
☐	See the guidance to create or update a landing page for creating a new index.yml landing page file.For example, see the [Azure Virtual Network landing page at https://review.docs.microsoft.com/en-us/azure/virtual-network/?branch=main.
☐	Add your service to the master breadcrumb file: azure-docs-pr\bread\toc.yml.
☐	If you have a product forum URL, add it to the docset docfx.json file: `MicrosoftDocs/azure-docs-pr/blob/master/docfx.json`. Under `"feedback_product_url"` add your folder path and feedback URL (see other line items as an example).
☐	Submit a request to add the new service on your area hub page (if it applies) and include the new service icon to use.
☐	Divy Sharma (DIVYARATNA) is responsible to include your new service on the "all services" page. Contact Divy to verify everything is ready to go.
☐	When your content is ready, understand the pull-request (PR) submission recommendations and create a PR to receive feedback from SMEs.
☐	Check the build validation, and address any issues.
☐	Check the Acrolinx results, if available. See Acrolinx coverage.
☐	Invite reviewers to provide feedback on the new content.

☑	TASKS
☐	After you incorporate the reviews and feedback, sign off (#sign-off) on your PR to merge it. If you're using a release branch, you might create several PRs to bring in content from all contributors. Review each PR before merging with the release branch. Signing off is a requirement for Microsoft Docs. Talk to your repo admin for your process.
☐	Schedule your [release branch merge](contribute-content-release-planning. md#release-planning-process-for-self-managed-releases-with-a-release-branch-azure-docs-pr).
☐	Generate an fwlink to your landing page so your product team can link to the content from the portal.
☐	Consider whether you'll need reference content for any APIs, libraries, packages, CLIs. Onboarding times can be long, so take timelines into consideration.
☐	Whether your service is going into public preview or GA stage, remember to request a Product Launch Readiness(PLR) sign-off for docs. PLR is a new Azure-wide process where the product and as the docs need to be signed off. Specifically, you could look at the Azure Doc PLR process for content-specific guidelines. Private previews don't require a PLR sign-off.

What is the Content Standards report?

11/2/2020 • 5 minutes to read

The goal of the Content Standards report is to advise Azure content owners and content managers about how well their content meets content standards for their overviews, quickstarts, and tutorials, as well as their tables of content (TOCs). The report also provides specific information for which criteria are not being met in an article and links to guidance for those criteria.

Individual writers can assess content they own. Managers can aggregate based on service or team members or both.

The criteria chosen for this dashboard are those that can be evaluated by logic and in general speak to metadata, scannability, and consistent article structure. The audit does not assess the quality of the content.

Why these criteria are important:

- **Metadata**: content needs to have the right metadata for reporting purposes. This is especially important for the content performance dashboard since the calculations for article health are tied to the ms.topic value.
- **Scannability**: one of the five writing principles, it supports the quickness with which a customer can determine if the article addresses their question.
- **Consistent article structure**: ensuring that our TOC is standard and that our content shows up in the sections it belongs in helps customers find our content when browsing the TOCs.

Report scope

- Repo: azure-docs-pr
- Audit criteria

How it works

All criteria are from existing standards that are documented in the Contributor Guide.

Backend programming evaluates each article against the criteria. Each article receives a pass/fail grade for each criterion. If the article meets 80% of the criteria or more, it is considered compliant. If the article meets less than 80% of the criteria, it is considered not compliant. There is no weighting of criteria.

The report is refreshed daily.

How to use it

The dashboard has two tabs: **Charts** and **Articles**.

The **Charts** tab provides data visualizations for groups of content—a portfolio view. You can filter by article type, ms.service, author, manager. You can exclude exempt content from the charts. This report is useful for seeing and sharing aggregated views on a set of topics.

Learn more about the Charts tab.

The **Articles** tab provides article-specific audit results. It is useful for identifying articles that need to be updated to meet the 80% pass rate. For each article, it tells you which rules the article passes, and which rules the article fails. You can request exemptions for some of the criteria from this page.

Learn more about the Articles tab.

Audit criteria

RULE ID	RULE NAME	CRITERIA
Overview rules		
Ov01	ms.topic value	ms.topic must be "overview"
Ov02	H1 format	H1 format must be "What is *service*?" or "What are *services*?"
Ov03	Last H2	Last H2 must be "Next steps"
Ov04	Next steps link	Overview "Next steps" section must have 1-3 links
Ov05	Next steps link format	Overview "Next steps" link format must not use blue box link
Quickstart rules		
QS01	ms.topic value	ms.topic must be "quickstart"
QS02	meta title keyword	Meta title must include the word "quickstart"
QS03	meta description	Meta description must include the word "quickstart"
QS04	H1 keyword	Quickstart H1 must start with "Quickstart:"
QS05	First H2	First H2 must be "Prerequisites"
QS06	H2 numbering	H2s must not be numbered as steps in a procedure
QS07	CUR H2	"Clean up resources" section must come before "Next steps" section
QS08	Last H2	Last H2 must be "Next steps"
QS09	H3 numbering	H3s must not be numbered as steps in a procedure
QS10	Green checkmarks	Quickstart must not contain green checkmarks
QS11	Next steps link	Quickstart "Next steps" section may only have 1 link
QS12	Next steps link format	Quickstart "Next step" link format must be blue box link

RULE ID	RULE NAME	CRITERIA
Tutorial rules		
T01	ms.topic value	ms.topic must be "tutorial"
T02	meta title keyword	Meta title must include the word “tutorial”
T03	meta title numbering	Series number not allowed in meta title
T04	meta description	Meta description must include the word “tutorial”
T05	H1 keyword	Tutorial H1 must start with “Tutorial:”
T06	H1 numbering	Series number not allowed in Tutorial H1
T07	Green checkmarks	Intro: Tutorial must use green check marks before first H2
T08	First H2	Tutorial first H2 must be “Prerequisites”
T09	H2 numbering	H2s must not be numbered as steps in a procedure
T10	CUR H2	“Clean up resources” section must come before “Next steps” section
T11	Last H2	Last H2 must be “Next steps”
T12	H3 numbering	H3s must not be numbered as steps in a procedure
T13	Next steps link	Tutorial “Next steps” section may have only 1 link
T14	Next steps link format	Tutorial “Next steps” link format must be blue box link
TOC rules		
TOC01	Top link label	Top link in TOC must be in format "*Service* documentation"
TOC02	Top link destination	"*Service* documentation" must link to *service* index.yml
TOC03	Overview node number of topics	TOC has no more than 2 topics in Overview node, and only 1 can be ms.topic: overview
TOC04	Overview node topic type	TOC must have an Overview node and it must contain an Overview topic

RULE ID	RULE NAME	CRITERIA
TOC05	Tutorials node expanded	TOC Tutorials node must be expanded by default if no Quickstarts node
TOC06	Tutorials node required	TOC must have required Tutorials node with at least one Tutorial]
TOC07	Quickstarts node	TOC Quickstarts node must be expanded by default if there is one
TOC08	Node order + labels	TOC nodes must be in correct order. Labels must be spelled correctly.
TOC09	Node type	TOC may only contain allowed nodes
TOC11	Overview topic location	Overview topic must be in Overview node of the TOC
TOC12	Tutorial topic location	Tutorial topic must be in the Tutorials node of the TOC
TOC13	Quickstart topic location	Quickstart topic must be in the Quickstarts node of the TOC
TOC14	Quickstart in H1	Do not add “Quickstart” to the H1 of any article not in the Quickstarts node of the TOC
TOC15	Green checkmarks	Article uses green checkmarks but is not a tutorial
TOC16	Tutorial in H1	Do not add "Tutorial" to the H1 of any article not in the Tutorials node of the TOC

Next steps

- Learn more about the Charts tab.
- Learn more about the Articles tab.

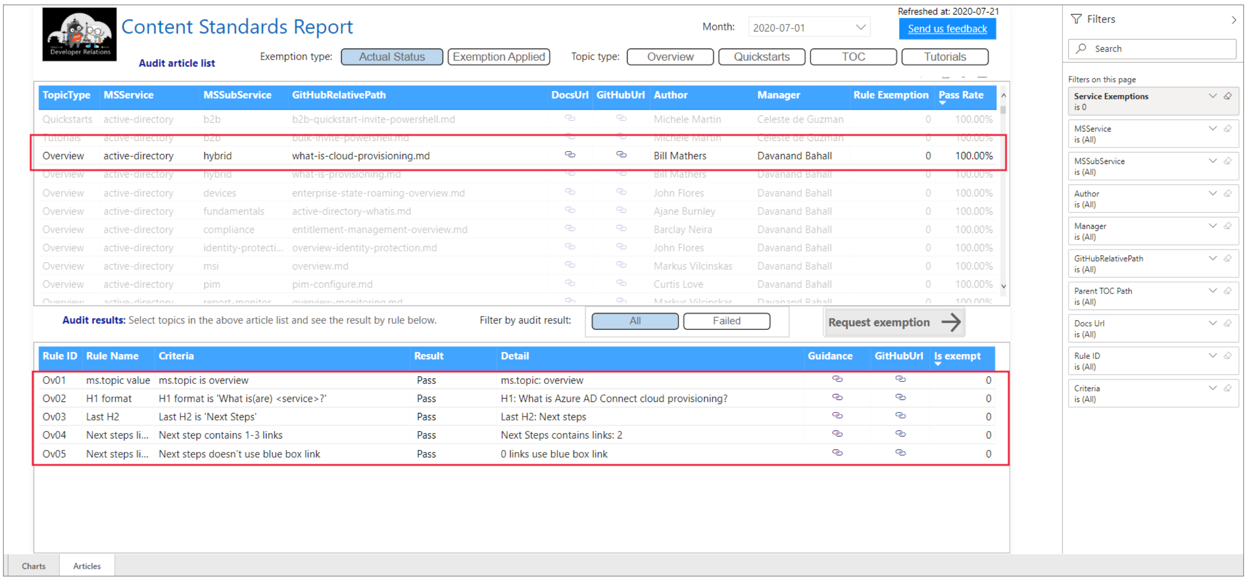
How to use the Articles tab of the Content Standards report

11/2/2020 • 7 minutes to read

The **Articles** tab of the Content Standards report has two output tables and multiple filters across each table and in the **Filter** pane on the right.

The top table is the **Audit article list** and lists all the articles that are the result of the applied filters. The bottom table, **Audit results**, lists the audit results of the articles in the top table.

If you select a single article in the **Audit article list**, it will filter the results in the **Audit results** to the rules that apply only to the selected article.



Filters set on the **Charts** page do not apply to this page.

What's in the Audit article list?

The Audit article list contains the following columns:

COLUMN	INFORMATION DISPLAYED
Topic type	Overview, Quickstarts, Tutorials, TOC
MSService	The value in the ms.service field
MSSubService	Derived from the folder hierarchy of azure-docs-pr repo
File name (GitHubRelativePath)	File name of the article
Links to the article in GitHub and on docs.microsoft.com	
Author	The value in the ms.author field
Manager	Derived from HeadTrax

COLUMN	INFORMATION DISPLAYED
Rule exemption	0 if no rule-level exemption applied to article
Service exemption	0 if no service-level-exemption applied
Pass rate	80% or higher is needed to be considered compliant

Click a column heading to sort by that column. You can only sort on one column at a time.

If the list is long, you can apply additional filters from the Filters pane or toggle the Exemption type and Topic type filter buttons at the top of the list.

To reset filters

There are several ways to reset the filters:

- The filter buttons toggle between on (blue) and off (white).
- To reset a filter in the **Filters** pane, select the **erase** icon on the filter.
- Select **Reset to default** at the top of the report to remove all applied filters.

What's in the Audit results list?

The Audit results list gives you the specific rules applied to the content in the Audit article list, including whether the content passed or failed that rule.

This list displays:

COLUMN	INFORMATION DISPLAYED
Rule ID	Unique ID for each rule
Rule name	Short name for the rule. **Rule ID** and **Rule name** are mainly used by the report and the exemption workflow.
Criteria	The criteria the content needs to meet for the rule.
Result	**Pass**: the article meets the criteria for the rule. **Fail**: the article doesn't meet the criteria for the rule.
Detail	What the audit report found in the article when testing for that rule.
Guidance	A link to the section in the Contributor Guide article that explains the rule.
GitHubUrl	A link to the article in GitHub.
Is exempt	0 if no exemption to this rule is applied in the article.

To focus on the criteria you need to address, you can filter out the passing rules by clicking the **Failed** button at the top of the list.

Note that sometimes, the only thing you need to fix is a spelling or capitalization error. For example, there's a lot of variety in how the node for how-to content is labeled. The correct label is "How-to guides".

Find articles by author

To find articles belonging to a specific writer or multiple authors:

1. In the **Filters** pane, open the **Author** filter, and click to select the author's name for each author whose articles you want to see.
2. The results of the filter will be displayed in the **Audit article list**. The results of the audit will appear in the **Audit results** list.
3. To see only audit results for a single article, select that article in the **Audit article list**. Only the rules that apply to that topic type and the pass/fail result will be displayed in the **Audit results** list.

Find articles by manager

To find articles belonging to writers reporting to a specific manager:

1. In the **Filters** pane, open the **Manager** filter, and click to select the manager's name.
2. The results of the filter will be displayed in the **Audit article list**. The results of the audit will appear in the **Audit results** list.
3. To see only audit results for a single article, select that article in the **Audit article list**. Only the rules that apply to that topic type and the pass/fail result will be displayed in the **Audit results** list.

Find TOCs

Because there is no ms.author for a TOC.yml file, TOC ownership is assigned to the manager of the writing team that writes the content for the service. To find the TOCs a manager owns, filter on the manager's name in the **Author** filter. Service and subservice values for TOCs are derived from the folder path for the TOC file.

Find articles by service or sub-service

You can filter on both service and subservice. When you select a service in the **MSService** filter, the **MSSubService** filter list is scoped to only the subservices of the selected filter.

To find articles belonging to a single service or to see articles for multiple services:

1. In the **Filters** pane, open the **MSService** filter, and click to select the service or services whose articles you want to see.
2. The results of the filter will be displayed in the **Audit article list**. The results of the audit will appear in the **Audit results** list.
3. To see only audit results for a single article, select that article in the **Audit article list**. Only the rules that apply to that topic type and the pass/fail result will be displayed in the **Audit results** list.

Request an exemption

You can request that an article or TOC be exempt from some of the rules. The allowed exemptions are:

EXEMPTION TYPES	CRITERIA IDS	FOLDER-LEVEL EXEMPTION	TOPIC-LEVEL EXEMPTION	TOC-LEVEL EXEMPTION	NOTES

EXEMPTION TYPES	CRITERIA IDS	FOLDER-LEVEL EXEMPTION	TOPIC-LEVEL EXEMPTION	TOC-LEVEL EXEMPTION	NOTES
Retiring service	n/a	Yes	No	No	An example is a service that is being replaced by or merged with a newer service, and writing and developer resources are no longer available for the content.
Legacy content	n/a	Yes	No	No	An example is content derived from a purchased service and whose content is published as-is until it can be replaced.
Content in Azure repo that doesn't follow MVC by design	n/a	Yes	No	No	An example is a mature service whose content comes to our org because of a larger re-org or migration.
TOC contains non-standard nodes	TOC09	Yes	No	Yes	Most frequent: stacked TOC, but also can be used for adding a single node, such as Troubleshoot.
TOC contains required Tutorials section	TOC06	No	No	Yes	
TOC nodes are in a non-standard order	TOC08	No	No	Yes	
Overview topic in a node other than Overview	TOC11	No	Yes	No	
Tutorial topic in a node other than Tutorials	TOC12	No	Yes	No	
“Tutorial” is in H1 of an article not in the Tutorials node	TOC16	No	Yes	No	

EXEMPTION TYPES	CRITERIA IDS	FOLDER-LEVEL EXEMPTION	TOPIC-LEVEL EXEMPTION	TOC-LEVEL EXEMPTION	NOTES
Quickstart topic in a node other than Quickstarts	TOC13	No	Yes	No	
“Quickstart” is in H1 of an article not in the Quickstarts node	TOC14	No	Yes	No	

You can only file exemptions against TOC files although they may actually affect another topic type. For example, TOC14 allows you to use the “Quickstart” H1 keyword in an article that is not linked to from the Quickstarts node.

To file an exemption in the report

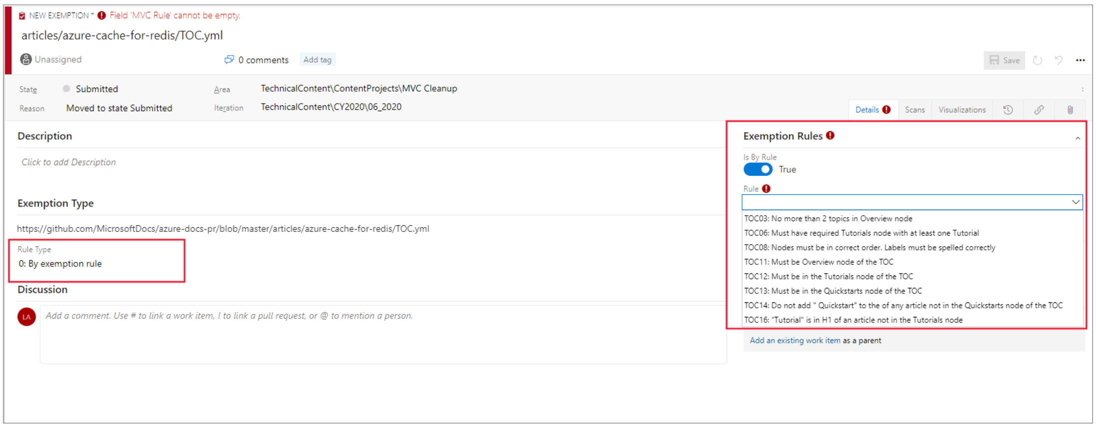
1. In the **Audit article list**, select the TOC you want to file an exemption for. The **Request exemption** button becomes active and turns blue. Note, the **Request exemption** button doesn’t turn blue, you cannot file an exemption against that article.

2. Click the **Request exemption** button. An Azure DevOps exemption request work item opens. The **Title** field is populated from the report and the article’s GitHub file path is pulled into the **Exemption Type** area.

3. In **Exemption Type** area, under **Rule Type**, select the exemption type you want. The options are:

    - **0. By exemption rule**

        - When you want a rule-level exemption, select this option, and then select the specific rule in the **Exemption Rules** area. Click under **Rule** to see the list of rules available. You can only choose one.



    - **1. Retiring service**

    - **2. Legacy content**

    - **3. Service is exempt from standards**

    - **4. Request non-standard nodes**

4. In the **Business Justification** area, provide the reason for your exemption request.

5. Save the new exemption request.

The request goes into the review backlog. The reviewers will review the request and make a decision. If the request is approved, the **State** will be set to **Approved** and the exemption will be integrated into the audit report. If the request is denied, the **State** will be set to **Denied**, and you will need to revise your content to meet the criteria.

Note: You can also file an exemption request outside of the tool. To do so, open Azure DevOps in TechnicalContent, select **Work items** > **New Work item** and, then select the **Exemption** work item type. If you log a request this way, you will need to provide the full github path to the TOC file in The **Exemption Type** area:

```
https://github.com/MicrosoftDocs/azure-docs-pr/blob/master/articles/<folder>/TOC.yml
```

Exemption request queries

- Request backlog
- Requests in review
- Approved requests
- Denied requests

Report feedback

To report a problem or provide feedback on the report, use the **Send us feedback** link at the top of the report.

Next steps

- Content Standards report overview
- Learn about the Charts tab

How to use the Charts tab of the Content Standards report

11/2/2020 • 2 minutes to read

The four charts on the **Charts** page provide an aggregate view of your portfolio.

For all charts on this page, if no filters are applied, the data applies to the entire Azure portfolio being measured. The available filters on this page are MSService, Author, and Manager, as well as the topic type filter buttons. You can also take approved exemptions into consideration by selecting the **Exemption Applied** filter button.

Filters set on the **Articles** tab do not apply to this page.

Compliance rate trend

Use this chart if you want to see a month-over-month trend chart for compliance. Data begins with FY2021.

Compliance rate by topic type

The bar chart indicates the percent compliance by topic type. This chart is also a filter for the other charts. Click a column to filter for that topic type. Ctrl+click to select more than one column.

Compliance count by typic type

The stacked bar chart indicates the number of compliant and not compliant topics by type. This chart is also a filter for the other charts. Click a column to filter for that topic type. Ctrl+click to select more than one column.

Number of passing and failing articles by rule type

The stacked bar chart indicates the number of compliant and non-compliant topics by rule type. To learn more about the specific rule, hover over the column to see the details fly-out.

Next steps

- Content Standards report overview
- Learn about the Articles tab

Write security documentation for an Azure service

3/5/2021 • 2 minutes to read

This article is for writers who are writing security documentation for their Azure service.

An Azure service that has articles about security must have the following table of contents (TOC) node:

- Concepts > Security

The service documentation can have the following security articles under the Security node:

- A security features article that provides an overview or introduction to security for the service.
- Conceptual articles to provide in-depth explanation of functionality related to the service.
- A security baseline article that contains recommendations to improve the security posture of a service.

Service documentation may also have security how-to articles. How-to articles show the customer how to complete a security task and go under the TOC node **How-to guides**.

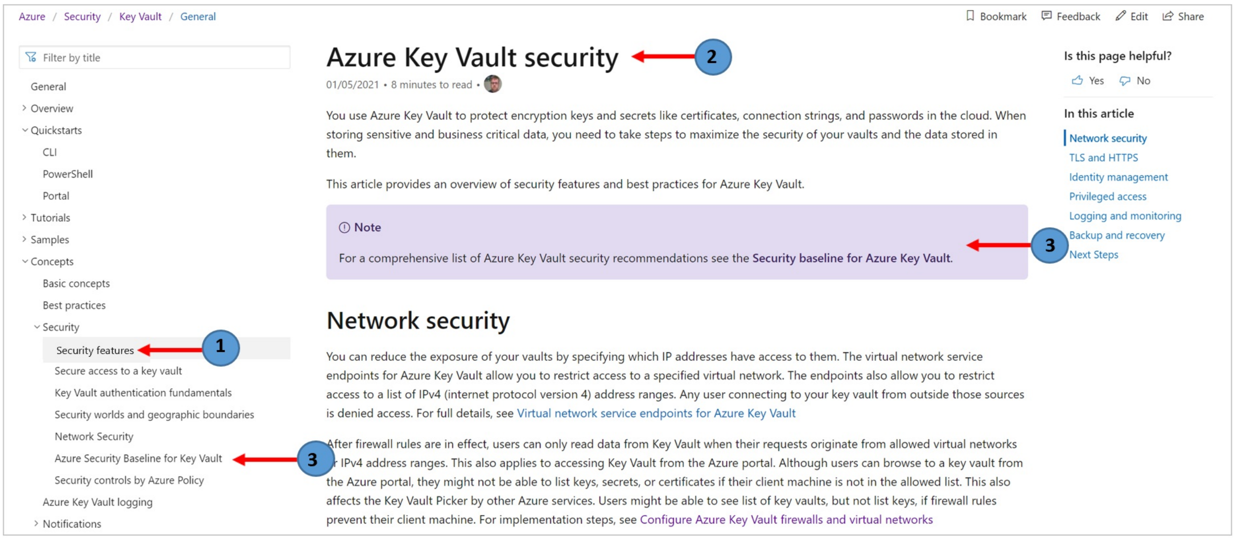
Security features

The security features article is an overview-type article that introduces the service's security capabilities. The service writer determines the capabilities to address in this article.

Security features checklist

Here's a checklist to make sure your security features article meets the key formatting guidelines.

NUMBER	REQUIREMENT
1	Use the TOC entry **Security features** as the first node under **Concepts > Security**.
2	H1 is "*Service name* security".
3	If the service has a security baseline article, include the following note in the section under the H1: "For a comprehensive list of *Azure service* security recommendations see the *service's security baseline article*." Link to the baseline article.
4	The .md file name for the security features article is security-features.md.
5	Metadata attribute ms.topic is set to conceptual.



Use this template to develop the security features article for an Azure service.

Conceptual article

Service documentation may have security conceptual articles. A conceptual article should focus on service-specific information.

Azure security fundamentals documentation provides high-level conceptual information on most security capabilities. All services that support a capability should link to security fundamentals docs for this high-level information. Linking to security fundamentals documentation for conceptual information enables the writer for a service to focus on service-specific information in their documentation.

If security fundamentals documentation doesn't document a capability that you want to link to, contact terrylan about adding the capability to fundamentals.

Conceptual article checklist

Here's a checklist to make sure your conceptual article meets the key formatting guidelines.

- Under the TOC node **Concepts > Security**.
- Address service-specific information.
- If security fundamentals documentation covers a security capability that you need to cover in your documentation, link to security fundamentals instead of creating your own.

How-to articles

Service documentation may have security how-to articles. Security how-to articles go under the TOC node **How-to guides**. Where relevant provide guidance for portal, PowerShell, CLI, and Azure Resource Manager (ARM) templates.

Next steps

- Template for writing a security features article
- Guidance for writing a security baseline article
- The Azure security fundamentals site on docs

Base template for introducing security features supported by an Azure service

3/5/2021 • 2 minutes to read

This article provides a template in markdown code that you should use when writing the security features article. The security features article introduces key security capabilities supported by your Azure service.

All articles need the metadata header along with the required values for reporting.

You can copy and paste the following markdown for your security features article:

```
---
title: Template for writing the security features article #Required; page title displayed in search results.
Include the words "security" and the Azure service name. For example, Azure Backup security. Include the
brand.
description: Template for writing the security features article #Required; article description that is
displayed in search results. Include the words "security" and the Azure service name.
author: TerryLanfear #Required; your GitHub user alias, with correct capitalization.
ms.author: terrylan #Required; Microsoft alias of author; optional team alias.
ms.service: security #Required; service per approved list. service slug assigned to your service by ACOM.
ms.topic: conceptual #Required
ms.date: 02/16/2021 #Required; mm/dd/yyyy format.
---

<!---Recommended: Remove all the comments in this template before you sign-off or merge to master.--->

<!---The security features article contains introductory information about key security capabilities
supported by your service. This article is located under the "Concepts" node and "Security" sub-node of your
service's ToC.

The security features article does not have to address every security capability supported by the service,
only key capabilities.

A goal of the security features article is to help the customer navigate to how-to information. This article
can link directly to how-to articles. If a service, though, has a conceptual article that provides more
detail about a security capability, the security features article can link to the conceptual article

--->

# <Name of Azure service> security
<!---Required:
Starts Azure service name followed by "security."
Example H1: Azure Backup security
--->

Introductory paragraph
<!---Required:
The introductory paragraph helps customers quickly determine whether an article is relevant.
Describe in customer-friendly terms what the service is and does, and why the customer should care. Keep it
short for the intro. This article is about security capabilities so focus on how the service helps to secure
data and resources.
--->

An example of an introductory paragraph in the security features article:

One of the most important steps you can take to protect your data is to have a reliable backup
infrastructure. Azure Backup provides security to your backup environment - both when your data is in
transit and at rest. This article lists security capabilities in Azure Backup that help you protect your
data and meet the security needs of your business.
```

```
<!--- Required:
If your Azure service has a security baseline article then link to the baseline article via a note.
Use the following text "For a comprehensive list of <Azure service name> security recommendations see the <title of service's baseline article>".
--->

An example of a note in the security feature article:

> [!Note]
> For a comprehensive list of Azure Key Vault security recommendations see the [Azure Security Baseline for Key Vault](https://docs.microsoft.com/azure/key-vault/general/security-baseline).

## <H2 title>
The service writer determines the format of and the information in the article's H2 sections.

## Next steps

<!--- Required:
Always have a Next steps H2.
Use regular links; do not use a blue box link. Insert links to other articles that are logical next steps or help users use the service.
Do not use a "More info section" or a "Resources section" or a "See also section".
--->
The service writer determines the conceptual or how-to articles linked to under Next steps.
```

What is the Azure Architecture Center?

3/5/2021 • 4 minutes to read

The Azure Architecture Center (AAC) helps customers design, build, and operate solutions on Azure.

Our mission statement: **Publish technical content and cloud best practices to provide the right information to the right customers at the right time. Architectural guidance is proven, authoritative, actionable, and relevant.**

The Architecture Center provides real-world guidance to solve real-world problems. It helps customers onboard to Azure and ensure long-term success. To achieve these goals, guidance on the Architecture Center follows these principles:

- **Proven:** The guidance published on the Architecture Center is based on proven practices.
- **Practical:** The Architecture Center helps customer build real solutions.
- **Comprehensive:** The Architecture Center provides guidance on all aspects of building for the cloud:
  - Operational Excellence
  - Security
  - Cost Optimization
  - Reliability
  - Performance Efficiency

The Architecture Center is governed by the patterns & practices (PnP) team within Developer Relations. This platform leverages the Microsoft Docs contribution guide and training for onboarding new users to collaborate with our platform.

This article provides the structure of the Architecture Center platform. The article AAC field contributions describes how new contributors can onboard and start publishing on this platform.

Audience

Our customers are architects, developers, and administrators who are responsible for designing, creating, and operating solutions on Azure to deliver business value.

GitHub repositories

The Azure Architecture Center content is accessible via:

- a public-facing GitHub repo, for general public wanting to make minor changes using the online editing option, and
- a private-facing GitHub repo, that allows proper staging and larger content changes. This is recommended for contributions from Microsoft employees having GitHub account.

Content types

The Architecture Center publishes some well-defined content types, distinguished by **ms.topic** and **ms.subservice** metadata values.

IMPORTANT

Article metadata are extremely important for content traffic and performance analysis, discoverability, and specific site experiences. Most articles in the Architecture Center are *conceptual*, and hence will have corresponding value in the *ms.topic* metadata. These conceptual articles are distinguished from each other by specific *ms.subservice* values. Make sure to use the right metadata as mentioned below.

Architectural content

This includes article types that provide architectural solutions for real-world customer scenarios, at varying degrees of technical depth. The **ms.topic** value for these articles must be **conceptual**.

- Technical guides for Azure categories
  - These are design guides that discuss a solution using Azure services in depth. These reside under the ***Azure categories*** TOC node, and typically appear as a series of articles. They were historically developed to cover any learnings that were not covered by a corresponding reference architecture, and so were based on tried and tested source code. These do not need to follow a set pattern, however they should cover considerable depth of the topic. For example, see the Azure Data Guide.
  - The **ms.subservice** value for these types must be **azure-guide**.
- Reference Architecture
  - These are the gold-standard architectures, backed by tried and tested engineering assets. These articles should discuss how the architecture affects the five pillars of a Well-Architected solution.
  - The **ms.subservice** value for these types must be **reference-architecture**.
- Example Scenario
  - These are typically architectures or solutions implemented by one or more customers, and hence typically coming from *field contributions*. These may or may not have engineering assets. They should attempt to discuss all five pillars of a Well-Architected solution.
  - The **ms.subservice** value for these types must be **example-scenario**.
- Solution Idea
  - These are an idea of a solution, which may or may not have been already implemented.
  - The **ms.subservice** value for these types must be **solution-idea**.

Cloud fundamentals

These include industry standard guides, focusing on fundamental patterns and practices of cloud architectures. The **ms.topic** value for these articles must be **conceptual**.

- Technology-agnostic guides
  - This is guidance that is fundamental to cloud development, and not limited to any Azure-specific architecture. This category includes articles such as *design principles*, *technology choices*, and so on.
  - The **ms.subservice** value for these types must be **guide**.
- Design Pattern
  - These are common engineering patterns used to solve a specific problem.
  - The **ms.subservice** value for these types must be **design-pattern**.
- Anti-patterns
  - These are common engineering patterns that may not be able to scale under pressure.
  - The **ms.subservice** value for these types must be **anti-pattern**.
- Best practices
  - These are some carefully-constructed best practices recommended for cloud development.
  - The **ms.subservice** value for these types must be **best-practice**.

Other topics

- Well-Architected Framework is a framework that guides architectural excellence, in the form of five pillars: Cost Optimization, Operational Excellence, Performance Efficiency, Reliability, and Security. The `ms.topic` value for these articles must be `conceptual`, and the `ms.subservice` value for these types must be `well-architected`.
- Topic Landing Page - these are a landing page that provide an opening to a series of articles, combining a navigational index with interesting visual elements. The `ms.topic` value for these articles must be `landing-page`.

Many of the content types have associated code assets, such as code samples, reference implementations, and deployment scripts. When published, they can reside in our repos or yours, as long as they are not personal repos (org repos only).

Retirement strategy

The nature of the Azure environment means that content will frequently become inaccurate, no longer apply, or need updating. Documents that are no longer valid and cannot be made valid via editing will be archived and removed regularly. The AAC follows the Docs Retirement process with minor adjustments:

- Every article is reviewed by an SME every six months. Whenever a review occurs, the ms.date metadata is updated. This metadata is the flag used for automated stale content processing.
- If an SME cannot be located or unable to perform the edits, or the content is no longer accurate/stale, it will be flagged for removal. A notice will be placed on the page and it will be removed from the TOC (if applicable)
- After an article has been flagged for 30 days, it will be archived via the process outlined by the Docs Retirement process.

Next steps

Interested in contributing to Azure Architecture Center? Read AAC Field Contributions to learn the process.

Azure Architecture Center contributions

4/5/2021 • 9 minutes to read

The Azure Architecture Center encourages contributions from both the public and Microsoft FTEs, either via GitHub issues and pulls requests (PRs), or within the context of programmatic alignment. Some general guidelines for contributing programmatically to the Architecture Center are detailed here.

Base requirements

All external contributions *must have* the following attributes:

- Describes a specific, actionable, and repeatable pattern being described where appropriate.
- Provides templates, scripts, and commands to achieve the scenario being described (not just a story).
- Covers a gap in existing articles and describes a "golden path" (not just "a path").
- Crosses individual SKU boundaries. If your proposal is specific to an individual service, perhaps contributing to the documentation for that service would be more suitable.
- Is a supported scenario (no hacks).
- Discusses the key pillars of Architecture Center material: Cost, Security, Reliability, Scalability, Resiliency, and DevOps.

All external contributions *must not have* the following attributes:

- Has limited pattern applicability (fewer than dozens of customers could benefit).
- Incurs unreasonable costs for the consumer unless we specify what type of consumer will benefit from it. For example, if you give us a pattern that only an S500 could afford, that's not much of a pattern.

All external contributions *should have* the following attributes:

- Is well-researched and validated by peers. Should also have been successfully implemented in multiple customers.
- Won't be invalidated by changes to the platform within the next engineering semester.
- Is accessible for reading and adoption by architects or developers at level 200 or higher.

All contributions that contain code samples *should have* the following attributes:

- Use the latest .NET and C# language version for all .NET code.
- Follow the recommended C# naming conventions.
- Use StyleCop to keep your code style consistent across samples.
- Keep simplicity as the main goal for code samples. Anything done in the code that doesn't respect best practices should be explained in code comments. This approach helps readers understand that while things might be done differently in production, another approach was used for simplicity. Avoid overengineering at all costs.
- If there are code assets, identify a person or team to maintain them. If the original author leaves the company, another contact will be made within their organization for assistance. Without a contact, the material might be removed if it's no longer valid or flagged based on our retirement strategy.

Azure Architecture Center contribution program (AAC Contributions)

New content aligned with the preceding content types typically should be developed in concert with the Architecture Center. There are several benefits to this approach:

- Aligns with our content publishing timeline supporting topics and requests already in our pipeline.
- Aligns with relevant product group teams about upcoming feature changes.
- Allows the Architecture Center to provide light project management and facilitation of our process for your document to ensure the accuracy and longevity of your solution.
- Our writers will be able to proactively provide you guidance throughout the publication process to ensure your content meets our content quality and accessibility bar.
- Our team can work to make your images crisp and supportable on the platform.

The Architecture Center alignment process consists of three steps:

1. **Proposal**: This simple document in draft form describes what you want to publish and why. It includes the initial content of the text but doesn't need to include supporting graphics, code samples, or other materials. After we review your draft, we'll provide you with written guidance on whether the content can meet our content needs. We'll tell you which content type it best aligns with. We'll let you know how we can support you in your development. You'll also be assigned a PM who will be your point of contact throughout the process. During this step, we'll seek to answer these main questions:

    a. Is the material technically accurate?
        a. Engineering team from Patterns and Practices (PnP) will help evaluate technical accuracy.
        b. Optionally, SMEs from the targeted Azure technological areas might be consulted. The SME might be a product team contact PM if it's a field contribution, and vice versa.
    b. Does the solution use services or features that might be deprecated soon? Will the solution be invalidated within the next engineering semester?
        a. No part of the solution should be listed on Azure Updates as getting deprecated.
    c. Does the solution use services or features in preview?
        a. Private preview features shouldn't be considered until they're either in public preview or generally available.
        b. If the solution uses public preview features, the article should have a cautionary note at the top.
    d. Is the material repeated elsewhere?
        a. The goal is to provide a consistent story between Architecture Center, Cloud Adoption Framework (CAF), and Learn platforms.
    e. Is the material a better fit elsewhere?
        a. Would this material be a better fit as a blog post or a white paper?
        b. Is this material a cloud migration scenario? If so, it will be a better fit for the CAF platform.
        c. Is this lab-focused training material? If so, it will be a better fit for Learn.
    f. What article format will it follow?
    g. What deployable assets are planned? Who will own them?
        a. Deployable source code or scripts will be required for reference architectures and optional for other article types.
    h. For example scenarios only, has more than one customer validated the material?
    i. What industry trend or content gap within the Architecture Center will this material cover?
    j. Is there a draft write-up present during the intake?
    k. Would the solution proposed by the material be supported by a product group?

2. **Drafts**: If your draft can be included in the Architecture Center, you'll receive notification from our team. Then you can begin refining your document to meet publishing standards based on the content type. During this phase, you're welcome to share drafts with the PM team for basic guidance, grammar, and voice assistance. You can also discuss ideas or ask other questions. You'll be responsible for writing the content and providing all supporting material requested by the PM teams. These items include graphics, code samples, and other artifacts. At the end of this phase, your document should be materially complete.

3. **Publication and post publication**: If your document is approved for publication, final checks will be

performed. The authors will then participate in the Docs Contributor Training and you'll convert your material to Markdown for publication. We'll merge your material, and then it will be your responsibility to maintain the content's accuracy. We might also contact you for an update if a platform change breaks or invalidates your scenario. If the material gets outdated, it might be removed or modified by us to bring it back into alignment.

Step details

Step 1: Proposal

- An external contributor has an idea they want published in the Architecture Center.
- A draft overview of the idea is created and shared with the PnP team.
- The article scope is decided on and aligned to existing content types.
- An `fcp-` prefixed branch in `architecture-center-pr` is created for the content draft.
- The contributor forks the branch into their own repo and adds their article's content to their repo.
  - If you haven't set up your GitHub account for Docs contribution yet, see GitHub account setup.
- The PM managing the content provides feedback and is available for guidance during the writing process to ensure the article meets our guidelines and quality bar.

Step 2: Drafts

- The PM tracks and follows up on the content to ensure it gets completed.
- The contributor opens a PR from their local fork to the draft `fcp-` prefixed branch on `architecture-center-pr`. This action signals that the contributor feels their contribution is ready to enter the review and publishing pipeline.
- The PM and the contributor both agree when the content is ready to move to the publication step.
- The contributor or writers make sure the publication follows the established writing voice and SEO guidance.
- The content is reviewed for technical accuracy by PnP, peers, product or FTE field SMEs (subject matter experts), or the Microsoft technical community.
  - Modifications of various natures might be requested of the contributor at this step, and the PR will be iterated on.
- The PM approves the PR and merges it to the `fcp-` prefixed branch on `architecture-center-pr`.
- A copy-edit pass is carried out by PnP or a vendor.

Step 3: Publication

- The content PM or lead writer gives final approval.
- The TOC and landing page links are created.
- A PR is created by the PM or lead writer, taking the content from the draft `fcp-` prefixed branch to `master`.
- The PR and content are reviewed by the content validation team.
- The author, PnP, or the PM implements fixes to unblock the PR.
- The PM or lead writer approves the PR and signs off.
- The content validation team merges it to the `master` branch.
- The `master` branch is merged with the `live` branch on the Docs release schedule.

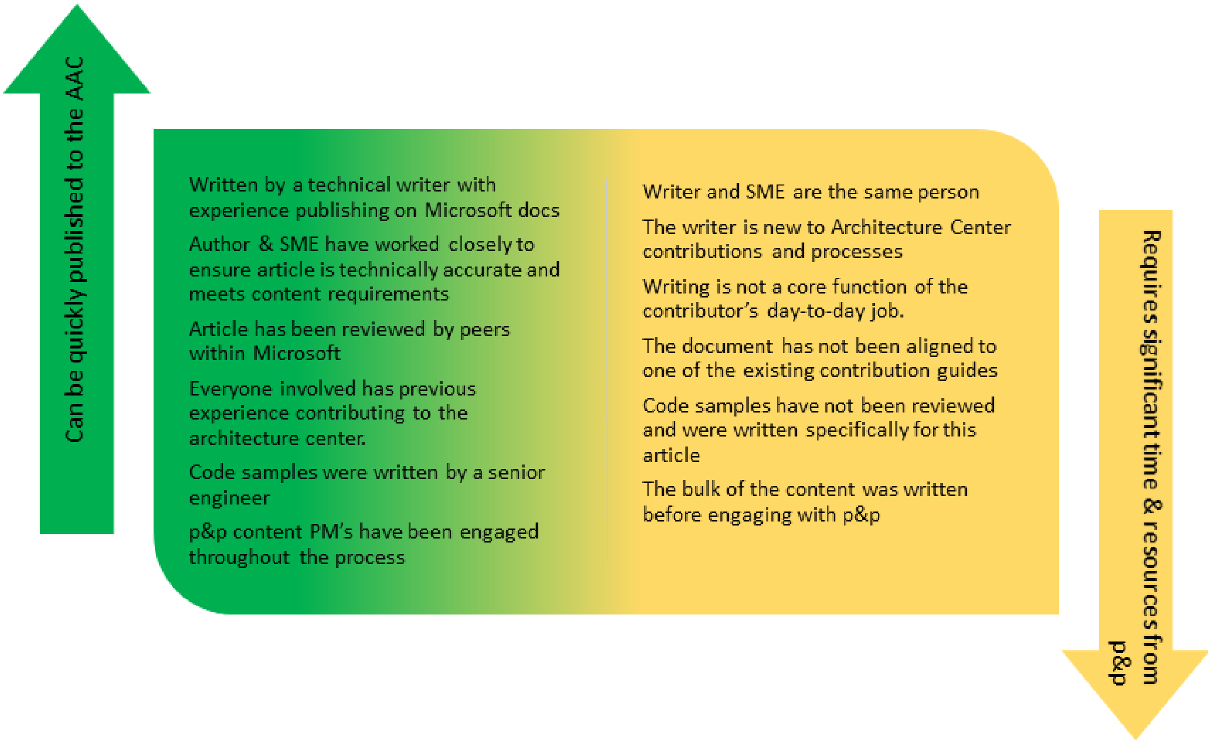
Step 4: Post publication

- **Editorial review**: The lead writer or PM might request an optional editorial review to maintain quality of the article.
- **Performance monitoring**: The PnP team will regularly review performance of AAC contributions and try to improve metrics and SEO. If, in spite of all mitigations, the article continues to perform below standards or doesn't attract traffic, the team will contact the contributors for triage.
- **Content maintenance**: Content freshness is essential for customers' confidence in our platform. For architectural articles and Azure-specific technical guidance, the PnP team will monitor if the article is *abandoned*. An abandoned article is one that hasn't been updated for more than 120 days. If so, PnP will contact the contributor to collaborate on updating the article.

Publishing timelines

New article contributions to the Architecture Center must be reviewed for quality, technical accuracy, and alignment with one of our existing content types before publication. Because of these review requirements, all contributions must be added to our backlog.

Articles that can be quickly moved through our backlog have many of the same qualities, as shown in the following diagram.



Next steps

Now that you are aware of the AAC contribution lifecycle, learn about some important guiding principles for your content:

- If you are new to GitHub, it's very easy to set up your account. Remember to link it to the Microsoft Docs organization for proper access rights.
- When you are ready to start writing, go over this quick reference for partner contributors.
- AAC contributions exclusively use release branches in GitHub to allow team collaboration. Familiarize yourself with release branches.
- Word your content keeping the writing principles in mind. Here's an easy checklist to make sure your contributions have the same voice as the rest of AAC.
- Be mindful of SEO recommendations when writing your content. Search engines provide the biggest entrypoint to our content, and they are quite easy to optimize. At the very minimum, try to:
    - write SEO-friendly titles,
    - write SEO-friendly meta descriptions, and
    - write SEO-friendly H1s. If you need more support, content your PnP PM or writer.
- Once you are ready with your markdown, invite your PnP PM or writer to review your content.
- After the article is published, you may be curious to know how well it is doing with our readers. Check out our content performance dashboard to analyze the signals from your readers.

Authoring content for the Azure Architecture Center Browser

3/5/2021 • 6 minutes to read

Patterns and Practices have adopted a new Azure Architecture Center (AAC) browser similar to the content browser experience as MS Learn and the docs code sample browser. In adopting the new browser experience, all content that is displayed in the browser must use the latest composable documentation model. This article explains how this content is composed for contribution to the AAC browser.

Browser content structure

Content presented in the AAC Browser comprises three or more files:

- A `.yml` file that contains metadata and tagging and renders into the published doc.
- One or more `.md` files that contain the document content.
- A `.png` image file that is used as the thumbnail on the browser card.

All of these are detailed in the following sections.

Article YAML file

The YAML file holds the metadata, filtering criteria, and pointers to all content files. The full schema is found here. Here's an example:

```
### YamlMime:Architecture
metadata:
  title: Manage hybrid Azure workloads using Windows Admin Center
  titleSuffix: Azure Reference Architectures
  description: Deploy Windows Admin Center to manage environments on-premises and in Azure.
  author: neilpeterson
  ms.date: 07/29/2020
  ms.topic: conceptual
  ms.service: architecture-center
  ms.subservice: reference-architecture
azureCategories:
  - hybrid
  - management-and-governance
products:
  - azure
  - azure-portal
  - windows-server
name: Manage hybrid Azure workloads using Windows Admin Center
summary: Deploy Windows Admin Center to manage environments on-premises and in Azure.
thumbnailUrl: /azure/architecture/browse/thumbs/hybrid-server-os-mgmt-wac-azure.png
content: |
  [!include[](hybrid-server-os-mgmt-content.md)]
```

The `.yml` file can be located anywhere in the repo and is picked up for indexing by the presence of the `### YamlMime:Architecture` comment on line 1. Other attributes include:

- `title` is rendered as the article's title (H1).
- `name` is rendered as the title on the browser card.
- `summary` is rendered as a description on the browser card, below the thumbnail.
- `azureCategories` contains categories used as the Categories dimension of filtering selection on the browser.

See Azure Categories for details.

- `products` contains product tags used for Products filtering on the browser. See Products for details.
- `thumbnailUrl` specifies the absolute path to the image used as the thumbnail.
- `content` contains a list of markdown files that make up the body of the article, rendered in the order listed.

Content files (Markdown)

For the AAC browser, article content is authored in Markdown much the same as other Docs content, but with the following differences:

- Do not add metadata to the content file. Metadata for the article is specified in the corresponding YAML file.
- Do not add an H1 or title to the content file. The article title is specified by the `title` metadata in the corresponding YAML file.
- The content files must be named `<yaml-filename>-content.md`. The files are intentionally excluded from the repo build process using this file naming convention.

To add one or more Markdown files into the article, create a list of content files like this:

```
content: |
  [!include[](source-one-content.md)]
  [!include[](source-two-content.md)]
```

Browser thumbnails

Each article included in the browser needs to define an image thumbnail that the AAC Browser displays in the browser card. You'll base the thumbnail image on the architecture diagram image in the article. If your article doesn't have an architecture diagram, use an image from the article that best represents it.

Existing articles already have browser cards with thumbnails. For a new article, you're responsible for creating the thumbnail image and adding that image to the repository. You create a thumbnail manually by using Paint 3D to adjust the aspect ratio so that the AAC Browser renders it correctly.

> **NOTE**
>
> There's no need to resize the image when you create the thumbnail. The docs generator dynamically scales each image to fit into the browser card.

Follow these steps to create a thumbnail image with the correct aspect ratio:

1. Open the **Paint 3D** app.

   If you don't have **Paint 3D**, use any graphics editor that allows you to set the aspect ratio of an image.

2. Select **Menu** from the upper-left corner.

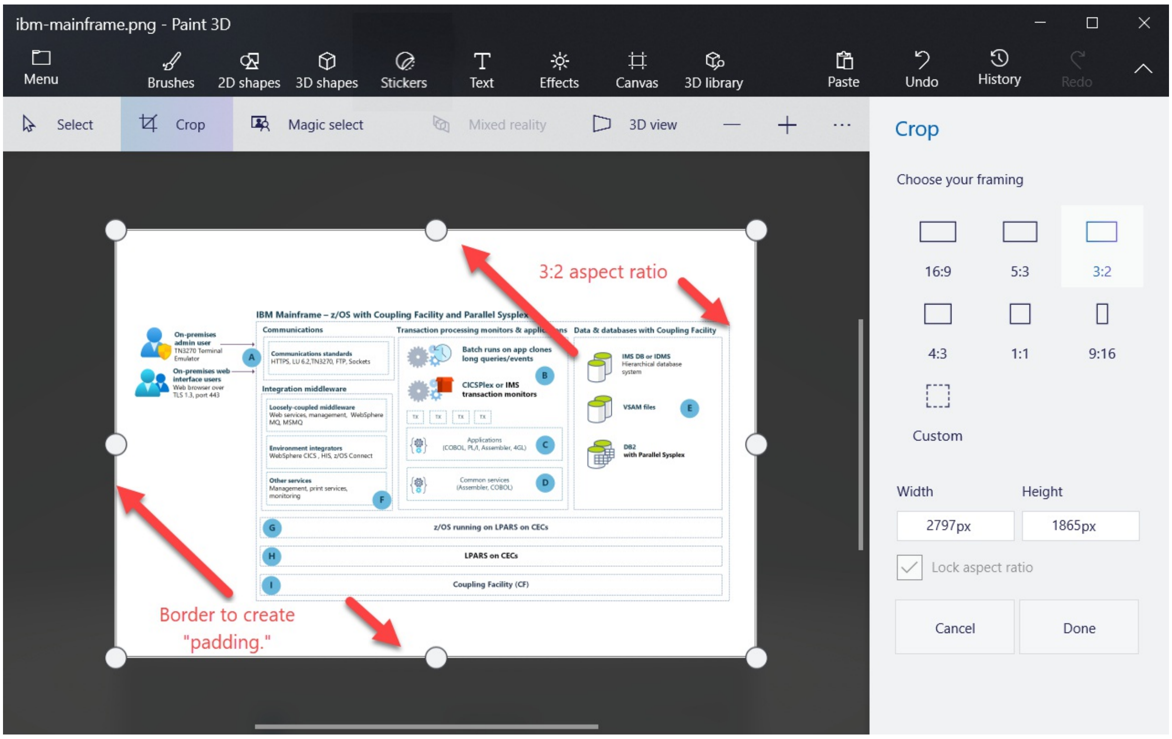
3. Select **Open** > **Browse files** and open the image you need to adjust.

4. With your image open, select **Crop**.

5. From the **Crop** sidebar, select **3:2**.

6. Drag the 3:2 ratio frame over the image until the image is centered and has a border for "padding."

   If the image doesn't fit into the frame, select **Cancel** to exit **Crop**. Then, select **Canvas** from the menu bar, make the canvas bigger, and go back to cropping the image to fit the 3:2 aspect ratio.



7. When you're ready, save the thumbnail image in the `/docs/browser/thumbs/` directory of the AAC repository.

8. Go to the **doc.yml** file that supports your article, and add the file path of your new thumbnail image to `thumbnailUrl:`.

> **IMPORTANT**
>
> As seen in the following example, the thumbnails **must be** referenced in the docs YAML file using an absolute path.
>
> ```
> thumbnailUrl: /azure/architecture/browse/thumbs/<your-thumbnail>.png
> ```

Staging browser cards

The cards found in the browser are not staged for preview when creating a pull request against the default branch. The cards are only available to view once content has been merged into the default branch and published live.

To work around this limitation, we have a staging branch that is indexed by the browser back end: https://review.docs.microsoft.com/azure/architecture/browse/?branch=wi-163353-browse After a PR has been merged into this branch, the new cards can be seen at this location.

> **NOTE**
>
> After new content is merged to the staging branch, the new card might not be visible in the staged browser for 5-10 minutes.

After the content has been verified, re-create the PR against the the default branch.

Azure Categories

The following are valid values for `azureCategories` specified in the YAML file.

```
"ai-machine-learning",
"analytics",
"blockchain",
"compute",
"containers",
"databases",
"developer-tools",
"devops",
"featured",
"hybrid",
"identity",
"integration",
"iot",
"management-and-governance",
"media",
"migration",
"mixed-reality",
"mobile",
"networking",
"security",
"storage",
"web",
"windows-virtual-desktop"
```

Valid values for `azureCategories` are maintained in the https://static.docs.com/ui/latest/schemas/Architecture.schema.json file.

Products

The `products` section of the YAML file lists tags that the browser uses for filtering by product. When specifying tags:

- Limit the list to the architecture's most important components.
- Don't include *azure* in the list unless no other tag is suitable.
- Put the list in alphabetical order.

The following are valid values for `products` specified in the YAML file.

```
"ai-builder",
"aspnet",
"aspnet-core",
```

```
"azure",
"azure-active-directory",
"azure-active-directory-b2c",
"azure-active-directory-domain",
"azure-advisor",
"azure-analysis-services",
"azure-anomaly-detector",
"azure-api-apps",
"azure-api-fhir",
"azure-api-management",
"azure-app-configuration",
"azure-app-service",
"azure-app-service-mobile",
"azure-app-service-static",
"azure-app-service-web",
"azure-application-gateway",
"azure-application-insights",
"azure-arc",
"azure-archive-storage",
"azure-artifacts",
"azure-attestation",
"azure-automation",
"azure-avere-vFXT",
"azure-backup",
"azure-bastion",
"azure-batch",
"azure-bing-autosuggest",
"azure-bing-custom",
"azure-bing-entity",
"azure-bing-image",
"azure-bing-news",
"azure-bing-spellcheck",
"azure-bing-video",
"azure-bing-visual",
"azure-bing-web",
"azure-blob-storage",
"azure-blockchain-service",
"azure-blockchain-tokens",
"azure-blockchain-workbench",
"azure-blueprints",
"azure-boards",
"azure-bot-service",
"azure-cache-redis",
"azure-cdn",
"azure-clis",
"azure-cloud-services",
"azure-cloud-shell",
"azure-cognitive-search",
"azure-cognitive-services",
"azure-communication-services",
"azure-computer-vision",
"azure-container-instances",
"azure-container-registry",
"azure-content-moderator",
"azure-content-protection",
"azure-cosmos-db",
"azure-cost-management",
"azure-custom-vision",
"azure-cyclecloud",
"azure-data-box-family",
"azure-data-catalog",
"azure-data-explorer",
"azure-data-factory",
"azure-data-lake",
"azure-data-lake-analytics",
"azure-data-lake-gen1",
"azure-data-lake-gen2",
"azure-data-lake-storage",
"azure-data-science-vm",
```

```
"azure-data-science-vm",
"azure-data-share",
"azure-database-mariadb",
"azure-database-migration",
"azure-database-mysql",
"azure-database-postgresql",
"azure-databricks",
"azure-ddos-protection",
"azure-dedicated-host",
"azure-dedicated-hsm",
"azure-dev-spaces",
"azure-dev-tool-integrations",
"azure-devops",
"azure-devops-tool-integrations",
"azure-devtest-labs",
"azure-digital-twins",
"azure-disk-encryption",
"azure-disk-storage",
"azure-dns",
"azure-encoding",
"azure-event-grid",
"azure-event-hubs",
"azure-expressroute",
"azure-face",
"azure-farmbeats",
"azure-files",
"azure-firewall",
"azure-firewall-manager",
"azure-form-recognizer",
"azure-front-door",
"azure-functions",
"azure-fxt-edge-filer",
"azure-genomics",
"azure-hdinsight",
"azure-hdinsight-rserver",
"azure-hpc-cache",
"azure-immersive-reader",
"azure-information-protection",
"azure-ink-recognizer",
"azure-internet-analyzer",
"azure-iot",
"azure-iot-central",
"azure-iot-dps",
"azure-iot-edge",
"azure-iot-hub",
"azure-iot-pnp",
"azure-iot-sdk",
"azure-iot-security-center",
"azure-iot-solution-accelerators",
"azure-key-vault",
"azure-kinect-dk",
"azure-kubernetes-service",
"azure-lab-services",
"azure-language-understanding",
"azure-lighthouse",
"azure-linux-vm",
"azure-live-ondemand-streaming",
"azure-live-video-analytics",
"azure-load-balancer",
"azure-log-analytics",
"azure-logic-apps",
"azure-machine-learning",
"azure-machine-learning-designer",
"azure-machine-learning-studio",
"azure-managed-applications",
"azure-managed-disks",
"azure-maps",
"azure-media-analytics",
"azure-media-player",
"azure-media-services",
```

```
"azure-media-services",
"azure-metrics-advisor",
"azure-migrate",
"azure-monitor",
"azure-netapp-files",
"azure-network-watcher",
"azure-notebooks",
"azure-notification-hubs",
"azure-open-datasets",
"azure-personalizer",
"azure-pipelines",
"azure-playfab",
"azure-policy",
"azure-portal",
"azure-powerbi-embedded",
"azure-private-link",
"azure-qna-maker",
"azure-quantum",
"azure-queue-storage",
"azure-rbac",
"azure-redhat-openshift",
"azure-remote-rendering",
"azure-repos",
"azure-resource-graph",
"azure-resource-manager",
"azure-rtos",
"azure-sap",
"azure-scheduler",
"azure-sdks",
"azure-search",
"azure-security-center",
"azure-sentinel",
"azure-service-bus",
"azure-service-fabric",
"azure-service-health",
"azure-signalr-service",
"azure-site-recovery",
"azure-sovereign-china",
"azure-sovereign-germany",
"azure-sovereign-us",
"azure-spatial-anchors",
"azure-speaker-recognition",
"azure-speech",
"azure-speech-text",
"azure-speech-translation",
"azure-sphere",
"azure-spring-cloud",
"azure-sql-database",
"azure-sql-edge",
"azure-sql-managed-instance",
"azure-sql-virtual-machines",
"azure-sqlserver-stretchdb",
"azure-sqlserver-vm",
"azure-stack",
"azure-stack-edge",
"azure-stack-hci",
"azure-stack-hub",
"azure-storage",
"azure-storage-accounts",
"azure-storage-explorer",
"azure-storsimple",
"azure-stream-analytics",
"azure-synapse-analytics",
"azure-table-storage",
"azure-test-plans",
"azure-text-analytics",
"azure-text-speech",
"azure-time-series-insights",
"azure-traffic-manager",
```

```
"azure-translator",
"azure-translator-speech",
"azure-translator-text",
"azure-video-indexer",
"azure-virtual-machines",
"azure-virtual-machines-windows",
"azure-virtual-network",
"azure-virtual-wan",
"azure-vm-scalesets",
"azure-vmware-solution",
"azure-vpn-gateway",
"azure-web-application-firewall",
"azure-web-apps",
"azure-webapp-containers",
"blazor",
"blazor-server",
"blazor-webassembly",
"common-data-service",
"connected-services-framework",
"consumer",
"customer-care-framework",
"customer-voice",
"dotnet",
"dotnet-core",
"dotnet-standard",
"dynamics",
"dynamics-365",
"dynamics-business-central",
"dynamics-commerce",
"dynamics-cust-insights",
"dynamics-cust-svc-insights",
"dynamics-customer-engagement",
"dynamics-customer-service",
"dynamics-field-service",
"dynamics-finance",
"dynamics-finance-operations",
"dynamics-fraud-protection",
"dynamics-guides",
"dynamics-human-resources",
"dynamics-layout",
"dynamics-market-insights",
"dynamics-marketing",
"dynamics-prod-visualize",
"dynamics-product-insights",
"dynamics-project-operations",
"dynamics-project-service",
"dynamics-remote-assist",
"dynamics-retail",
"dynamics-sales",
"dynamics-sales-insights",
"dynamics-scm",
"dynamics-talent",
"dynamics-talent-attract",
"dynamics-talent-core",
"dynamics-talent-onboard",
"ef-core",
"expression",
"expression-studio",
"github",
"hololens",
"internet-explorer",
"m365",
"m365-ems",
"m365-ems-cloud-app-security",
"m365-ems-configuration-manager",
"m365-information-protection",
"m365-myanalytics",
"m365-security-center",
"m365-security-score",
```

```
  "m365-threat-protection",
  "m365-workplace-analytics",
  "mdatp",
  "mem",
  "mem-configuration-manager",
  "mem-intune",
  "microsoft-edge",
  "microsoft-identity-web",
  "microsoft-servers",
  "mlnet",
  "mrtk",
  "ms-graph",
  "msal-android",
  "msal-angular",
  "msal-ios",
  "msal-java",
  "msal-js",
  "msal-node",
  "msal-python",
  "msc",
  "msc-operations-manager",
  "msc-service-manager",
  "office",
  "office-365",
  "office-365-atp",
  "office-access",
  "office-adaptive-cards",
  "office-add-ins",
  "office-bookings",
  "office-excel",
  "office-exchange-server",
  "office-forefront",
  "office-kaizala",
  "office-lync-server",
  "office-onedrive",
  "office-onenote",
  "office-outlook",
  "office-planner",
  "office-powerpoint",
  "office-project",
  "office-project-server",
  "office-publisher",
  "office-skype-business",
  "office-sp",
  "office-sp-designer",
  "office-sp-framework",
  "office-sp-server",
  "office-teams",
  "office-ui-fabric",
  "office-visio",
  "office-word",
  "office-yammer",
  "passport-azure-ad",
  "power-apps",
  "power-automate",
  "power-bi",
  "power-platform",
  "power-query",
  "power-virtual-agents",
  "project-acoustics",
  "qdk",
  "silverlight",
  "skype",
  "sql-server",
  "sql-server-2008",
  "surface",
  "surface-duo",
  "vs",
  "vs-app-center",
```

```
  "vs-code",
  "vs-mac",
  "vs-online",
  "windows",
  "windows-api-win32",
  "windows-azure-pack",
  "windows-forms",
  "windows-iot",
  "windows-iot-10core",
  "windows-mdop",
  "windows-mixed-reality",
  "windows-server",
  "windows-smb-server",
  "windows-system-center",
  "windows-uwp",
  "windows-virtual-desktop",
  "windows-wdk",
  "windows-wpf",
  "xamarin",
  "xbox"
```

Valid values for `products` are maintained in the https://static.docs.com/ui/latest/schemas/Architecture.schema.json file.

Azure Architecture Center refresh

3/23/2021 • 3 minutes to read

This project aims to ensure that Azure Architecture Center content (deployments, diagrams, and copy) are up to date, well-architected, and aligned. Additional goals are to ensure deployments are continuously validated and remove AZBB from the Azure Architecture Center.

This document provides an overview of the refresh process with links to examples, and other assets that can be used as additional content is refreshed.

Published components

First, let's define the components that are included in the 2021 content refresh. For each article, these items should be present.

COMPONENT (CLICK FOR MORE DETAILS)	DESCRIPTION	EXAMPLE
Code sample *	Code + details in a mspnp GitHub repository.	secure hybrid network
Validation pipeline *	A simple validation pipeline to test each ARM deployment on pull request and on a schedule.	secure hybrid network
Diagram	Visual representation of the architecture / solution.	secure hybrid network
Reference architecture or copy	Content on docs.microsoft.com.	secure hybrid network

Code sample

For Azure infrastructure and **where applicable**, include a set of ARM templates and deployment instructions. For each deployment, ensure:

- Best practices found here Azure ARM Templates QuickStart Best Practices Doc.
- Deployment is well architected (highly available, diagnostics enabled, include Azure Monitor queries, dashboarding, etc).
- Store deployment here https://github.com/mspnp/samples.
- Include a validation pipeline for each deployment (detailed later in this doc).

Code samples browser

The readme for each deployment should include YAML front matter, which allows the sample to be indexed and published on the docs.microsoft.com code samples browser. More information can be found here.

The YAML front matter looks like this:

```
---
page_type: sample
languages:
- azurepowershell
- azurecli
products:
  - azure-app-service
  - azure-log-analytics
  - azure-key-vault
  - azure-sql-database
description: description goes here
---
```

Link to reference architecture

The readme for each deployment should include a link back to the related reference architecture. Use the docs.microsoft.com markdown button for the link. The markdown looks like this:

```
For detailed information, see the Implement a secure hybrid network:

> [!div class="nextstepaction"]
> [Implement a secure hybrid network](https://docs.microsoft.com/azure/architecture/reference-architectures/dmz/secure-vnet-dmz)
```

Build badge

The readme for each deployment should include the Azure Pipeline build badge associated with the automated validation (reference next section of this doc). The markdown used for the build badge can be found on the settings of the Azure Pipeline and will look like this:

```
[![Build Status](https://nepeters-devops.visualstudio.com/arm-template-validation-pipelines/_apis/build/status/secure-hybrid-network?branchName=master)](https://nepeters-devops.visualstudio.com/arm-template-validation-pipelines/_build/latest?definitionId=135&branchName=master)
```

Validation pipeline

Where applicable, create a simple set of automated validation tasks using Azure Pipelines. At a minimum, complete these things:

- Run ARM TTK tests and publish test results.
- Validate template against validation API.
- Deploy template twice (ensure frictionless idempotency).
- Delete deployment assets.

A sample pipeline with annotations can be found here - secure hybrid network.

Pipeline YAML

Store the validation pipeline YAML in the same directory as the main ARM template.

Azure DevOps project

This Azure DevOps project can be used to run the validation pipeline - arm-template-validation-pipelines. If you need access, reach out to Neil Peterson.

Validation Dashboard

Add a section to this Azure DevOps dashboard with a validation badge. The markdown used for the build badge can be found on the settings of the Azure Pipeline and will look like this:

```
[![Build Status](https://nepeters-devops.visualstudio.com/arm-template-validation-pipelines/_apis/build/status/secure-hybrid-network?branchName=master)](https://nepeters-devops.visualstudio.com/arm-template-validation-pipelines/_build/latest?definitionId=135&branchName=master)
```

Diagram

Update each diagram so that it is technically aligned with the deployment. Include things beyond resources such as network flow, diagnostic flow, and availability configurations. Some other things to consider:

- Make the main diagram short and wide. This size allows us to present it on top of the article without taking up the first page with art.
- Use updated icons where applicable.
- Consider using a shadow box for inserting a larger diagram or more detailed diagram. For more information on using shadow box, see, Create and expandable screenshot.

Copy update

Finally, update the copy of the article to match any changes to the deployment. This would be a good time to give the completed article a refresh pass. While doing so, check that:

- Everything in the copy up to date.
- Any additional screenshots to update.
- Can any extra copy be removed?

End to end examples

These articles have gone through this refresh process and can be used as examples.

- Hub and Spoke
- Basic Web Application
- Secure hybrid network

AAC refresh feedback

Raise any feedback and / or suggest options that you feel would improve the Azure Architecture Center better to the pnp team.

Create a reference architecture

3/24/2021 • 5 minutes to read

A **reference architecture** shows a proven architecture that a customer can deploy today on Azure.

It gives detailed recommendations on how to design and deploy the architecture to Azure. Also it includes guidance on the quality pillars (availability, scalability, security, operations) as they relate to the architecture. Finally, a reference architecture should (in most cases) include a deployment artifact such as an Azure Resource Manager template that incorporates the recommendations and best practices.

Target Audience: TDM, Architect, Infra/DevOps

Audience Level: 300-400

Examples: Azure reference architectures

Purpose

A reference architecture gives customers a "golden path" for a particular scenario or workload. If you follow this path, you'll avoid common pitfalls.

It also gives customers a jump-start to design their own production-ready deployments, using the provided scripts and templates as a starting point.

Reference architectures aren't quickstarts. They're intended to help customers plan real-world deployments, so reading time can be 10-30 minutes as needed to cover the details.

Reference architectures are also not tutorials. First, a reference architecture is oriented around a scenario or solution, rather than a task. Second, a reference architecture will generally point to existing how-to topics for individual tasks.

Contributing

In order to contribute a Reference Architecture, please follow the below outline and keep in mind these base requirements:

- Is it generalized and canonical? (A reference architecture represents the most recommended settings per each architecture style.)
- Does it include a reference implementation?
- Was the reference implementation perf tested
- Best practices for security, scalability, etc.

TOC and URL

Reference architectures are located in the Reference Architectures node of the Architecture Center TOC.

Related architectures are grouped into subnodes under Reference Architectures node.

The URL structure follows the same pattern as the TOC nodes:

```
/azure/architecture/reference-architectures/<group>/name-of-architecture
```

Document structure

We use a consistent structure for reference architecture documents.

H1 title - State the scenario

> Example: "Connect an on-premises network to Azure using ExpressRoute"

After the title, there's an introductory section, consisting of:

Intro paragraph (no heading)

A short paragraph that describes the scenario and the solution that is deployed. This paragraph should be concise, so that the diagram (next item) appears close to the top of the page.

Example:

> This reference architecture shows a set of proven practices for running multiple Linux virtual machines (VMs) behind a load balancer, to improve availability and scalability. This architecture can be used for any stateless workload, such as a web server, and is a foundation for deploying n-tier applications.

Architecture diagram

The diagram shows the Azure resources that get deployed. The level of detail will depend on the scenario.

> Example: For an IaaS deployment, the diagram *should* include the network topology (VNet, subnets, gateways, public IP addresses), VMs or VM scale sets, plus load balancers, Traffic Manager, and App Gateway, as appropriate. It *might* include other elements such as NICs, storage accounts, or VHDs, but only if the document contains specific guidance related to those elements. For example, if some VMs need multiple NICs, include those in the diagram. Otherwise, it can be assumed that each VM has a NIC.

H2 - Architecture

This section is a bullet list that describes the elements in the diagram. In most cases, there should be a 1:1 correspondence. Everything in the diagram should be described here.

For each element, describe its purpose within the context of the overall solution.

- **Weak:** "A subnet is a way to segment a VNet into multiple address spaces."**

- **Better:** "Put each application tier into a separate subnet."

It's OK to do both, describing *what* something is and *why* it's there. But avoid repeating information that's already in the product documentation for that service/feature. Instead, link to the product documentation.

H2 - Recommendations

This section gives more detailed recommendations for the elements in the previous section. Put all of the messy details and here.

This section can have subheadings (H3) to organize the recommendations. For example, "VM recommendations", "SQL Server recommendations", "Networking recommendations"

Examples of recommendations:

> - For each tier, put two or more VMs in an availability set.
> - Create a separate storage account for diagnostic logs.
> - For SQL Server high availability, we recommend using Always On Availability Groups.

As much as possible, provide the justification or rationale for each recommendation. Examples:

> - Put two or more VMs in an availability set. This makes the VMs eligible for a higher SLA.

> - Create a separate storage account for diagnostic logs to avoid hitting IOPS limits.

Recommendations should be definite when possible. In practice, sometimes you'll need a qualifier like "depending on your workload..."

H2 - Considerations (pillars)

After the recommendations section, include sections (H2) that describe non-functional characteristics of the architecture.

- Performance considerations
- Scalability considerations
- Availability considerations
- Manageability considerations
- Security considerations
- Cost considerations

Only include the sections that are relevant for the reference architecture. In some cases, there won't be anything to say for a particular pillar, especially if a reference architecture builds on a previous reference architecture, as part of a series.

These sections may include whatever discussion points are relevant for that pillar, with respect to the architecture. Including:

- What the architecture gives you.

  > "Scale out by adding more VMs to the load balancer pool." (Scalability) ← **A capability that this architecture gives you.**

- What the architecture doesn't give you. When possible, point to a possible alternative.

  > "If you need higher availability, replicate the application across two regions and use Traffic Manager for failover." (Availability) ← **What this architecture does *not* give you (availability during a regional outage), along with a pointer to something actionable.**

- Things to keep in mind to achieve the pillar, when using this architecture, including design and operational considerations.

  > "If your VMs run an HTTP server, create an HTTP health probe. Otherwise, create a TCP health probe." (Availability) ← **A best practice that depends on the specific application workload**

  > "The web front end should be stateless, to avoid the need to maintain client affinity." (Scalability) ← **An app dev consideration.**

- Additional options to consider, which may not be part of the golden path.

  > "If you need additional throughput, consider creating additional Event Hubs and sharding the messages." (Scalability) ← **Advanced option, not shown in the reference architecture.**

H2 - Deploy the solution

If there's a deployable asset (Azure Resource Manager template or script), include instructions for how to deploy.

Markdown templates

The markdown templates for a reference architecture are at Reference architecture markdown templates.

Create an example workload

3/24/2021 • 4 minutes to read

Example workloads guide customers through the design process of solving specific problems Azure. They provide actionable architecture guidance based on real customer examples. Their goal is to shorten a customer's learning curve by telling them the story of another customer who has been there before.

Each example workload has an architecture diagram, discusses decisions that went into the architecture, explores some alternatives, and includes cost estimates based on the pricing calculator. It may include a Resource Manager template to deploy.

Target Audience: TDM, Architect, Lead Developer

Adoption Stage: Planning

Audience Level: 100-200

Example of this content type: Baseball ML workload

Tenets

Example Workloads adhere to the following tenets:

- **Accelerate Customer Adoption**: Example Workloads accelerate customer adoption by providing guidance and lessons learned from others who have been there before.
- **Easy to Consume**: Workloads should be easy to consume for audiences of all technical levels.
- **Solve business problems**: Workloads use technology to solve problems facing the business, they aren't focused on the technology itself.
- **Based on real deployments**: They are either built from a customer deployment or set of common patterns derived from customer deployments.
- **Teaching to fish**: Example Workloads talk about the decisions that were made, what their constraints were, and the thought process that lead to the solution. They don't give guidance without explaining why.

What a workload isn't

- An Example Workload is not a step-by-step implementation guide or a tutorial
- Workloads aren't built to solve technology problems.
  - Things like setting up a load balancer, building a VPN, or running micro-services belong in other documents.
- They are not focused on a single specific technology or Azure service.

Purpose

Example workloads are similar in many ways to reference architectures, but have a different customer intent:

- Example workloads help a customer to understand the different types of architectures that are possible on Azure. After reading an example workload, the next step is to go deeper into the architectural guidance or go to the product documentation to learn more about the services used.
- Workloads are specific to an industry, vertical, or application domain. Reference architectures are more general ("N-tier application")
- Workloads are "bite sized"; they can be consumed more quickly than a reference architecture.

- Workloads aim for level 100-200, where reference architectures are built for a more technical audience.
- A reference architecture is Microsoft's recommendation for that architecture. It's a "golden path" that customers can follow. An example workload shows one way that a customer has solved their specific domain problem.

TOC and URL

All articles are located in the "Example Workloads" node of the Architecture Center TOC.

Related workloads are grouped into subnodes. The URL structure follows the same pattern as the TOC nodes:

```
/azure/architecture/example-workload/<group>/name-of-workload
```

Document structure

Example workloads have a consistent structure.

Markdown templates

The markdown templates for an example workload, example scenario, or sample solution are at Sample solution markdown templates.

Title

The title is a noun phrase that describes the workload.

> Example: "Insurance claim image classification on Azure"

Avoid naming the workload after the Azure technologies that are used.

Introductory section (no heading)

The introduction contains:

- A paragraph that describes what the solution does (the domain)
- A paragraph that contains a brief description of the main Azure services that make up the solution. This paragraph should convey the Azure value proposition, not a complete description of the architecture (which comes later).

Potential use cases

A bullet list of example use cases for this solution. The purpose is to show that an example architecture may apply to more than one application domain.

Architecture

This section starts with an architecture diagram. The diagram shows the Azure resources in the solution plus the data flow. It may include other elements such as a user, a mobile device, or an external data source.

After the architecture diagram, include a numbered list that describes the data flow through the solution. Start from the user or external data source and then through rest of the solution.

Components

A bullet list of components in the architecture (including all relevant Azure services) with links to the product documentation.

Alternatives

Use this section to list alternative Azure services or architectures that you might consider for this solution. Include the reasons why you might choose these alternatives.

Considerations (pillars)

Within this section, describe the non-functional characteristics of the architecture, under the following H3 headers:

- Scalability considerations
- Availability considerations
- Security considerations
- Resiliency considerations

Deploy the solution or See it in Action

If there's a deployment (Azure Resource Manager template or script), include instructions for how to deploy. This is greatly encouraged (even a small sample would be valuable), but it is not required.

Pricing

For this section, you should create three cost estimates for small, medium, and large deployments, using the Azure pricing calculator. Save and share the estimates, and include links to the three estimates in this section.

The exact definition of small, medium, and large will depend on the workload. Describe what the sizes mean for this workload. Size is defined by throughput or scale.

Next steps

This section walks customers through their next steps when they want to start building this out. It can link to additional training that may be required, if there are any reference architectures already built that cover this solution.

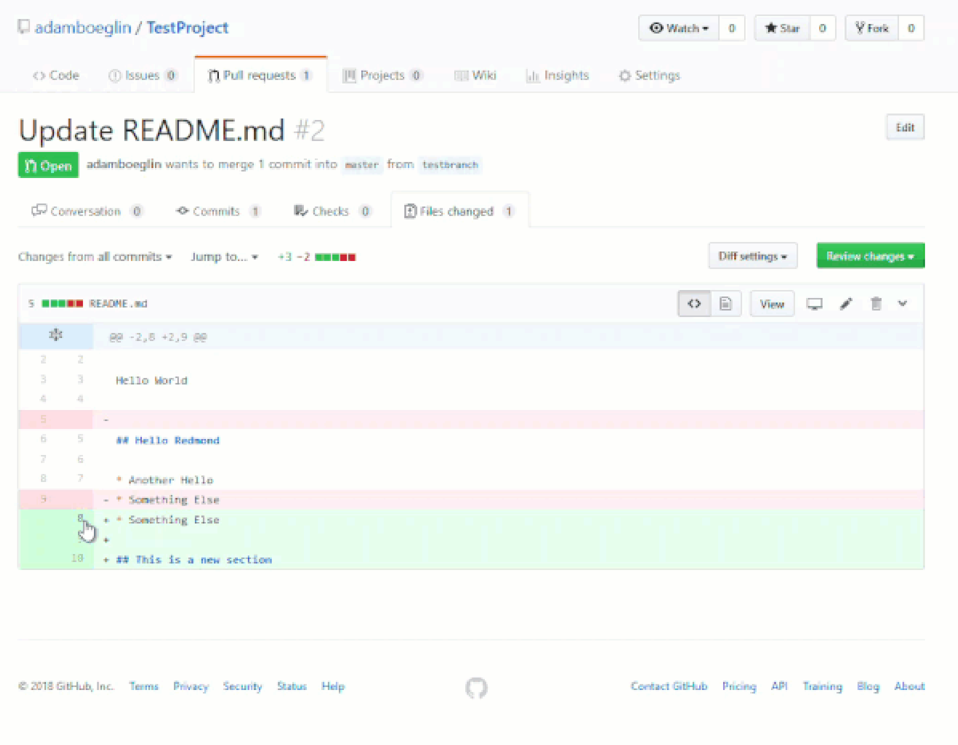
Related resources

Links to other resources that are relevant such as product documentation, case studies, or other workloads.

Article reviews

Before an article is published, it goes through both an editorial review and a technical review to validate the article and look for any technical inaccuracies or areas that should have been covered.

You'll need a GitHub account if you don't have one already. Using your personal account is fine. Once your account is setup, reviewing is easy. Navigate to the "Files" section of the pull request you're reviewing. Then click the "+" by the line number you'd like to comment on, add your notes, and click "Start Review". You can then add multiple comment lines and click "Finish Review" when complete.


adamboeglin / TestProject
Watch 0
Star 0
Fork 0
Code
Issues 0
Pull requests 1
Projects 0
Wiki
Insights
Settings
Update README.md #2
Edit
Open
adamboeglin wants to merge 1 commit into master from testbranch
Conversation 0
Commits 1
Checks 0
Files changed 1
Changes from all commits
Jump to...
+3 −2
Diff settings
Review changes
5 README.md
View
@@ -2,8 +2,9 @@
Hello World
Hello Redmond
* Another Hello
- * Something Else
+ * Something Else
+
+ ## This is a new section
© 2018 GitHub, Inc.
Terms
Privacy
Security
Status
Help
Contact GitHub
Pricing
API
Training
Blog
About

Create a solution idea

3/24/2021 • 2 minutes to read

Solution ideas provide brief overviews of solutions that customers can implement using Azure services.

Each solution idea has an architecture diagram, data flow, overview of the services used, and next step links.

Target Audience: CXO, TDM, Architect, Developer

Adoption Stage: Planning

Audience Level: 100-200

Examples of this content type: Solution Ideas

Document structure

Below you'll find the structure of a typical Solution idea

Title

The title is a noun phrase that describes the scenario.

> Example: "Back up on-premises applications and data to Azure"

Azure technologies can be used in the name if the goal is to highlight solving a specific problem with a specific Azure technology.

Introductory section (no heading)

The introduction contains:

- A paragraph that describes what the solution does (the domain)
- A paragraph that contains a brief description of the main Azure services that make up the solution. This paragraph should convey the Azure value proposition, not a complete description of the architecture (which comes later).

Potential use cases

A bullet list of example use cases for this solution. The purpose is to show that an example architecture may apply to more than one application domain.

Architecture

This section starts with an architecture diagram. The diagram shows the Azure resources in the solution plus the data flow. It may include other elements such as a user, a mobile device, or an external data source.

After the architecture diagram, include a numbered list that describes the data flow through the solution. Start from the user or external data source and then through rest of the solution.

Components

A bullet list of components in the architecture (including all relevant Azure services) with links to the product documentation.

Next steps

This section walks customers through their next steps when they want to start building this out. It can link to additional training that may be required, if there are any reference architectures already built that cover this solution.

Related resources

Links to other resources that are relevant such as product documentation, case studies, or other scenarios.

Markdown templates

The markdown templates for a solution idea are at Solution idea markdown templates.

Create a design pattern

3/24/2021 • 3 minutes to read

A software design pattern is a "reusable solution to a commonly occurring problem ... in software design" (Wikipedia). Cloud design patterns address specific challenges in distributed systems. They often address a quality pillar such as availability, or resiliency.

The qualities of a design pattern include:

- Proven. A design pattern emerges from actual implementations that have been shown to work. Often, a pattern is identified after the fact, by observing a common approach that appears repeatedly.
- Abstract. A design pattern is not the same thing as an implementation of that pattern. Design patterns are described in terms of objects or functional blocks that communicate or exchange data. An implementation may use a particular language or framework
- Generalized. A design pattern solves a problem that appears across multiple application domains. Design patterns are the building blocks of application architectures.
- Technology agnostic. A design pattern does not depend on a specific technology. For example, *publish-subcribe* is a design pattern, and Azure Service Bus is one way to implement the pattern.

Target Audience: Lead developer

Audience Level: 200-400

Find the current catalog of design patterns here.

Purpose

Provide customers with a catalog of patterns that they can use like a toolkit.

- A design pattern distills a set of real-world proven practices. They're technology agnostic. For example, "publish/subscribe" is a pattern, and using Azure Service Bus is one way to implement this pattern.
- The catalog of design patterns supports other architectural guidance, such as design guides and best practices articles. They allow us to refer to the patterns without explaining them each time. (The reason they're called "patterns" is because they show up in many places!)
- Design patterns help customers reason about their architectures from a functional/logical viewpoint, before worrying about the specifics of individual technologies.
- Design patterns aren't specific to a particular scenario, technology, or programming language. Usually there's more than one way to implement a pattern.

TOC and URL

Cloud Design Patterns are located in the Cloud Design Patterns node of the Architecture Center TOC.

The URL for a Cloud Design Pattern article is:

```
/azure/architecture/patterns/name-of-pattern
```

Categories

The design patterns are grouped into the following categories:

- Availability
- Data management
- Design and implementation
- Messaging
- Management and monitoring
- Performance and scalability
- Resiliency
- Security

A pattern will generally fall under at least one of these categories, and sometimes more than one. Under the Design Patterns landing page, there's a Categories node that has subpages for each of the categories. These allow customers to browse by category.

Document structure

A Design Pattern article follows a fixed structure that's inspired by the original book on software design patterns, Design Patterns: Elements of Reusable Object-Oriented Software.

Title

The title is the name of the pattern. If the pattern is already established within the developer community, use the standard name for the pattern.

After the title, include a short paragraph that summarizes the pattern. This paragraph should briefly describe the pattern and the reasons to use it. Don't include a lot of detail. The remainder of the article will go into the details. The purpose of the summary is for the reader to quickly decide whether the pattern is relevant to their problem.

Context and problem

This section describes the problem that the design pattern solves. Provide enough context to motivate the discussion of the problem. Often, the context involves an application's non-functional requirements, such as throughput, maximum latency, or concurrency.

Solution

This section describes the pattern. Describe how the pattern works, including the objects or functional blocks and their interactions. Also describe how the pattern solves the problem. When possible, include a diagram of the pattern. Don't use this section to describe any specific implementation of the pattern.

Issues and considerations

Use this section to describe considerations such as:

- Considerations for implementing the pattern.
- Challenges that you might face when implementing or using the pattern.
- Tradeoffs or possible negative implications of using a pattern.

When to use this pattern

Describe, in general terms, the situations when you would use this pattern. If possible, also list examples of when the pattern is not useful.

Example

Use this section to provide a concrete example of how to implement the pattern. In some cases, the example is a code sample. In other cases, the example might just describe how to use an Azure service to implement the pattern.

Markdown template

The markdown template for a design pattern is at Design pattern markdown template.

Create a topic landing page

5/5/2021 • 3 minutes to read

A topic landing page should be thought of as a hub that links to articles that already exist across documentation, blogs, GitHub, MSLearn, and the broader internet (if MS content is not available). There should be little to no content created specifically for this page.

The format was inspired by a recent user study that we've completed around how customers consume documentation. These pages were built to address the following high-level learnings from that study:

- Provide pathways that allow developers to learn through example projects and videos.
- Ensure that documentation has clear pathways for next steps.
- Participants desire case studies, tutorials, and logical links to other resources in the documentation so that they can learn how to effectively deploy to the cloud.

Topic focused pages should be a sort of virtual whitepaper helping customers to see and understand a path through our documentation and be a "home base" for learning about a specific technology on Azure. Think of everything required to be successful when implementing complex topics such as machine learning, IOT, microservices, and HPC. These require knowledge that spans products and services, relies on a foundation of knowledge around specific technologies, and are complex enough that there's no one single way of doing it. The topic landing page should help current and prospective customers feel confident that they're heading the right direction and not missing anything.

Example Page: https://docs.microsoft.com/azure/architecture/topics/high-performance-computing/index

Building the page

The topic pages should follow these guidelines:

- Assume the audience has no knowledge of the topic and provide a quick introduction to it
- Be visually interesting
  - Have a video outlining the topic & solutions at the very top
  - Include images and links to architecture diagrams and solutions
- Walk the customer through everything they would need to know about a given topic
  - Start at the basics and continue to add depth
  - Provide lots of examples
- Provide opportunities to go hands on with code & walk-throughs
- Outline clear next steps such as contacting sales, launching a demo, and viewing pricing information
- Include customer case studies and examples of others successfully implementing this solution

Start with an outline of the page before linking to any content. First outline your h2 headers covering the important high-level concepts and topics that are necessary for customers to fully understand this technology area. Then, locate the documentation required to fill in those areas and add explaining text to describe why each of these areas is important and what they're used for. When everything is complete, a Contextual TOC will be built to provide a cohesive experience when navigating between pages and breadcrumbs will be set up to help customers understand where they are across pages. This will be created by the Architecture Center team.

Markdown template

Use this Markdown template when creating a topic page:

```
---
title: TOPIC NAME on Azure
description: A guide to building running TOPIC NAME on Azure
author:
ms.date: 2/4/2019
---
<!-- markdownlint-disable MD033 -->
<!-- markdownlint-disable MD026 -->

# TOPIC NAME

## Introduction to TOPIC

<!-- markdownlint-disable MD034 -->

> [!VIDEO https://www.youtube.com/embed/]

<!-- markdownlint-enable MD034 -->

<description of Topic>

## Implementation checklist

As you're looking to implement TOPIC, ensure you're reviewed the following topics:

<!-- markdownlint-disable MD032 -->

> [!div class="checklist"]
> - Important [SECTION1](#section1)
> - Important [SECTION2](#section2)

<!-- markdownlint-enable MD032 -->

## SECTION1


### SUB-SECTION

This is a template that allows for the card layout on topic pages.  Use it for links to scenarios, reference
architectures, and solutions.  Only the highlighted sections (link, image, title, & description) will need
to be changed.

<ul class="columns is-multiline has-margin-left-none has-margin-bottom-none has-padding-top-medium">
    <li class="column is-one-third has-padding-top-small-mobile has-padding-bottom-small">
        <a class="is-undecorated is-full-height is-block"
            href="/azure/architecture/example-scenario/apps/hpc-saas?
context=/azure/architecture/topics/high-performance-computing/context/hpc-context">
            <article class="card has-outline-hover is-relative is-fullheight">
                    <figure class="image has-margin-right-none has-margin-left-none has-margin-top-none has-
margin-bottom-none">
                        <img role="presentation" alt="" src="../../example-scenario/apps/media/architecture-
hpc-saas.png">
                    </figure>
                <div class="card-content has-text-overflow-ellipsis">
                    <div class="has-padding-bottom-none">
                        <h3 class="is-size-4 has-margin-top-none has-margin-bottom-none has-text-
primary">Computer-aided engineering services on Azure</h3>
                    </div>
                    <div class="is-size-7 has-margin-top-small has-line-height-reset">
                        <p>Provide a software-as-a-service (SaaS) platform for computer-aided engineering
(CAE) on Azure.</p>
                    </div>
                </div>
            </article>
        </a>
    </li>
    <li class="column is-one-third has-padding-top-small-mobile has-padding-bottom-small">
        <a class="is-undecorated is-full-height is-block"
```

```
            href="/azure/architecture/example-scenario/infrastructure/hpc-cfd?context=/azure/architecture/topics/high-performance-computing/context/hpc-context">
            <article class="card has-outline-hover is-relative is-fullheight">
                    <figure class="image has-margin-right-none has-margin-left-none has-margin-top-none has-margin-bottom-none">
                        <img role="presentation" alt="" src="../../example-scenario/infrastructure/media/architecture-hpc-cfd.png">
                    </figure>
                <div class="card-content has-text-overflow-ellipsis">
                    <div class="has-padding-bottom-none">
                        <h3 class="is-size-4 has-margin-top-none has-margin-bottom-none has-text-primary">Computational fluid dynamics (CFD) simulations on Azure</h3>
                    </div>
                    <div class="is-size-7 has-margin-top-small has-line-height-reset">
                        <p>Execute computational fluid dynamics (CFD) simulations on Azure.</p>
                    </div>
                </div>
            </article>
        </a>
    </li>
 </ul>


## Cost or pricing

Details of how much this will cost and what effects it

## Security

For an overview of security best practices on Azure, review the [Azure Security Documentation](/azure/security/azure-security?context=/azure/architecture/topics/high-performance-computing/context/hpc-context).


Anything else that would relate to security


## Customer stories

Links to case studies or customer stories running this workload

## Other important information


## Next steps
```

Markdown template for design patterns

6/11/2021 • 4 minutes to read

Use this Markdown template to create a design pattern.

For guidance, see Create a design pattern.

```
---
title: Replace this text based on the guidance in Completing the metadata under the Article Title section.
description: Replace this text based on the guidance in Completing the metadata under the Article Title
section.
keywords: Replace this text based on the guidance in Completing the metadata under the Article Title
section.
ms.date: mm/dd/yyyy
ms.service = architecture-center
ms.topic = conceptual
ms.subservice = design-pattern
pnp.series.title: Cloud Design Patterns
pnp.pattern.categories: Replace this text based on the guidance in Completing the metadata under the Article
Title section.
---

# Article title

_Your article can have only one **H1 heading (#)**, which is the article title. The H1 heading is always
followed by a succinct descriptive paragraph that informs the reader what the article is about and how it
can help them. Do not start the article with a note or tip. Descriptive text always appears in italics.
Descriptive text should be deleted or replaced with your text prior to publication._

_This template is specific to **Cloud Design Patterns**. Your article should describe the problem that the
pattern addresses, identify considerations for applying the pattern, and provide an example based on
Microsoft Azure. Most of the patterns include code samples or snippets that show how to implement the
pattern on Azure. Your article should be broken down into six subheadings (H2, ## in markdown)--Context and
problem, Solution, Issues and considerations, When to use this pattern, Example, and Next steps. The H2
headings and descriptions are included in this template. If you need to create a new heading under one of
the H2 headings, use an H3 heading (###)._

_**Completing the metadata:**_
_This section provides guidance on completing the metadata section at the top of this template. Update the
placeholder text based on the following guidance:_

- _**title:** Use the H1 (#) title of your article from the top of this section. Both titles should be
identical. Maximum recommended length is 60 characters._
- _**description:** Provide a brief summary of your article. This is the description that appears in search
engine results, so ensure the summary is clear and concise and attracts your intended audience. Maximum
recommended length is 150-160 characters._
- _**keywords:** Add a comma-separated list of key concepts and terms from your article. These are the words
your intended audience will submit in a search engine._
- _**ms.date:** Enter the date in mm/dd/yyyy format, as shown in the metadata field. Initially this should
be the date your article is published. After publication, this field should be refreshed whenever the
article is updated so readers can see that the content is fresh._
- _**pnp.series.title:** Cloud Design Patterns is always the correct patterns and practices series title for
this template. Do not change._
- _**pnp.pattern.categories:** Review the bracketed list that follows and delete all categories that don't
apply to your article: [availability, data-management, design-implementation, messaging, management-
monitoring, performance-scalability, resiliency, security]. Copy and paste the final bracketed list into the
metadata section._

## Context and problem

_Provide a brief background on the specific pattern and the problem that the pattern addresses._
```

```markdown
## Solution

_Describe the solution for the problem identified in the **Context and problem** section. If there are
multiple solutions, put them in order of complexity and provide instructions on how the reader can choose
the best solution for their given problem. Best practices are to provide a step-by-step approach to
implement the solution, and to include screenshots to help guide the reader._

## Issues and considerations

_Highlight pertinent concerns relevant to implementing the solution provided in the previous section. Best
practices are to start this section with the following boilerplate sentence, followed by a bulleted list of
topics for the reader to contemplate prior to implementing the given solution._

Consider the following points when deciding how to implement this pattern:

- Item 1
- Item 2
- Item 3

## When to use this pattern

_Deliver specific guidance on when this pattern should and should not be used. Use the following boilerplate
text, followed by bullet points, to help the reader determine if the solution is applicable to their
specific scenario._

Use this pattern when:

- Item 1
- Item 2
- Item 3

This pattern might not be suitable:

- Item 1
- Item 2
- Item 3

## Example

_Include a working sample that shows the reader how the pattern solution is used in a real-world situation.
The sample should be specific and provide code snippets when appropriate._

## Next steps

_Provide links to other topics that provide additional information about the pattern covered in the article.
Topics can include links to pages that provide additional context for the pattern discussed in the article
or links to pages that may be useful in a next-steps context. Use the following boilerplate sentence
followed by a bulleted list._

The following information may be relevant when implementing this pattern:

- Item 1
- Item 2
- Item 3

## Related guidance

_Bulleted list that links to one or more related patterns._
```

Markdown and YML templates for reference architectures

5/5/2021 • 2 minutes to read

Use these Markdown and YML templates to create a reference architecture.

For guidance, see Create a reference architecture.

Each article must have two files. Follow the instructions for browse format authoring at Authoring content for the Azure Architecture Center Browser.

Name this file *<article-title>.yml*:

```
### YamlMime:Architecture
metadata:
  title: <H1 title of your article. Maximum recommended length is 60 characters>
  titleSuffix: Azure Reference Architectures
  description: <A brief summary of your article. This is the description that appears in search engine
results. Maximum recommended length is 150-160 characters.>
  author: <github username>
  ms.date: <publish or major update date - mm/dd/yyyy>
  ms.topic: conceptual
  ms.service: architecture-center
  ms.subservice: reference-architecture
azureCategories:
  - <choose at least one category from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#azure-categories>
  - <there can be more than one category>
products:
  - <choose 1-5 products from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#products>
  - <1-5 products>
  - <1-5 products>
name: <H1 title>
summary: <Write a summary. Can be same as description.>
thumbnailUrl: /azure/architecture/browse/thumbs/<browse-png-filename>.png
content: |
  [!include[](<article-filename>-content.md)]
```

Name this file *<article-title-content>.md*:

```
> The H1 title is the same as the title metadata. Don't enter it here, but as the **name** value in the
corresponding YAML file.

_Brief introduction goes here._ [**Deploy this solution**.](#deploy-the-solution)

![alt text](./media/folder_name/architecture-diagram.png)

_Download a [Visio file](https://arch-center.azureedge.net/architecture.vsdx) that contains this
architecture diagram. This file must be uploaded to `https://arch-center.azureedge.net/`_

## Architecture

The architecture consists of the following components:

- **Thing 1**. Description

- **Thing 2**. Description
```

```markdown
## Recommendations

The following recommendations apply for most scenarios. Follow these recommendations unless you have a
specific requirement that overrides them.

_Include considerations for deploying or configuring the elements of this architecture._

## Scalability considerations

_Identify and address scalability concerns relevant to the architecture in this scenario._

## Availability considerations

_Identify and address availability concerns relevant to the architecture in this scenario._

## Manageability considerations

_Identify and address manageability concerns relevant to the architecture in this scenario._

## Security considerations

_Identify and address security concerns relevant to the architecture in this scenario._

## Deploy the solution

_Describe a step-by-step process for implementing the reference architecture solution. Best practices are to
add the solution to GitHub, provide a link (use boilerplate text below), and explain how to roll out the
solution._

A deployment for a reference architecture that implements these recommendations and considerations is
available on [GitHub](https://www.github.com/path-to-repo).

1. First step
2. Second step
3. Third step ...

## Next steps

Links to articles on the AAC, and other Docs articles. Could also be to appropriate sources outside of Docs,
such as GitHub repos or an official technical blog post.

Examples:
* [Artificial intelligence (AI) - Architectural overview](/azure/architecture/data-guide/big-data/ai-
overview)
* [Choosing a Microsoft cognitive services technology](/azure/architecture/data-guide/technology-
choices/cognitive-services)
* [What are Azure Cognitive Services?](/azure/cognitive-services/what-are-cognitive-services)

## Related resources

Another optional link list, in bulleted form. If you have several links to other articles in Docs, include
those in "Next steps" and use "Related resources" for related architecture guides and architectures.

Here is an example section:

Fully deployable architectures:

* [Chatbot for hotel reservations](/azure/architecture/example-scenario/ai/commerce-chatbot)
* [Build an enterprise-grade conversational bot](/azure/architecture/reference-
architectures/ai/conversational-bot)
* [Speech-to-text conversion](/azure/architecture/reference-architectures/ai/speech-ai-ingestion)
```

Markdown and YML templates for example workloads

5/21/2021 • 5 minutes to read

Use these Markdown and YML templates to create an example workload, previously called an example scenario or sample solution.

For guidance, see Create an example workload.

Each article must have two files. Follow the instructions for browse format authoring at Authoring content for the Azure Architecture Center Browser.

Name this file *<sample-solution>.yml*:

```
### YamlMime:Architecture
metadata:
  title: <Article title, which becomes the title metadata>
  titleSuffix: Azure Example Scenarios
  description: <Write a 100-160 character description that ends with a period and ideally starts with a call
to action. This becomes the browse card description.>
  author: <contributor's GitHub username. If no GitHub account, use EdPrice-MSFT>
  ms.author: <contributor's Microsoft alias. If no alias, leave out this metadata as it will be
automatically set to pnp by the docfx.json.>
  ms.date: <publish or major update date - mm/dd/yyyy>
  ms.topic: conceptual
  ms.service: architecture-center
  ms.subservice: example-scenario
azureCategories:
  - <choose at least one category from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#azure-categories>
  - <there can be more than one category>
products:
  - <choose 1-5 products from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#products>
  - <1-5 products>
  - <1-5 products>
name: <The H1 title is a noun phrase that describes the scenario. Avoid naming the scenario after the Azure
technologies that are used. Example: "Insurance claim image classification on Azure">
summary: <Write a summary. Can be same as description.>
thumbnailUrl: /azure/architecture/browse/thumbs/<browse-png-filename>.png
content: |
  [!include[](<sample-solution>-content.md)]
```

Name this file *<sample-solution-content>.md*:

```
> The H1 title is a noun phrase that describes the scenario. Don't enter it here, but as the **name** value
in the corresponding YAML file.

Introductory section - no heading

> This should be an introduction of the business problem and why this scenario was built to solve it.
>> What industry is the customer in?
>> What prompted them to solve the problem?
>> What services were used in building out this solution?
>> What does this example scenario show? What are the customer's goals?

> What were the benefits of implementing the solution described below?
```

```
## Potential use cases

> Are there any other use cases or industries where this would be a fit?
> How similar or different are they to what's in this article?

These other uses cases have similar design patterns:

- List of example use cases

## Architecture

_Architecture diagram goes here_

> What does the solution look like at a high level?
> Why did we build the solution this way?
> What will the customer need to bring to this?  (Software, skills, etc?)

Under the diagram, include a numbered list that describes the data flow or workflow.

### Components

A bulleted list of components in the architecture (including all relevant Azure services) with links to the
service pages.

> Why is each component there?
> What does it do and why was it necessary?
> Link the name of the service (via embedded link) to the service's product service page. Be sure to exclude
the localization part of the URL (such as "en-US/").

- Examples:
  - [Azure App Service](https://azure.microsoft.com/services/app-service)
  - [Azure Bot Service](https://azure.microsoft.com/services/bot-service)
  - [Azure Cognitive Services Language Understanding](https://azure.microsoft.com/services/cognitive-
services/language-understanding-intelligent-service)
  - [Azure Cognitive Services Speech Services](https://azure.microsoft.com/services/cognitive-
services/speech-services)
  - [Azure SQL Database](https://azure.microsoft.com/services/sql-database)
  - [Azure Monitor](https://azure.microsoft.com/services/monitor): Application Insights is a feature of
Azure Monitor.
  - [Resource Groups][resource-groups] is a logical container for Azure resources.  We use resource groups
to organize everything related to this project in the Azure console.

### Alternatives

Use this section to talk about alternative Azure services or architectures that you might consider for this
solution. Include the reasons why you might choose these alternatives.

> What alternative technologies were considered and why didn't we use them?

## Considerations

> Are there any lessons learned from running this that would be helpful for new customers?  What went wrong
when building it out?  What went right?
> How do I need to think about managing, maintaining, and monitoring this long term?
> Note that you should have at least two of the H3 sub-sections.

### Availability

> How do I need to think about managing, maintaining, and monitoring this long term?

### Operations

> How do I need to think about operating this solution?

### Performance

> Are there any key performance considerations (past the typical)?

### Scalability
```

```
> Are there any size considerations around this specific solution?
> What scale does this work at?
> At what point do things break or not make sense for this architecture?

### Security

> Are there any security considerations (past the typical) that I should know about this?

### Resiliency

> Are there any key resiliency considerations (past the typical)?

### DevOps

> Are there any key DevOps considerations (past the typical)?

## Deploy this scenario

> (Optional, but greatly encouraged)
>
> Is there an example deployment that can show me this in action?  What would I need to change to run this in production?

## Pricing

> How much will this cost to run?
> Are there ways I could save cost?
> If it scales linearly, than we should break it down by cost/unit. If it does not, why?
> What are the components that make up the cost?
> How does scale affect the cost?
>
> Link to the pricing calculator with all of the components in the architecture included, even if they're a $0 or $1 usage.
> If it makes sense, include small/medium/large configurations. Describe what needs to be changed as you move to larger sizes.

## Next steps

> Where should I go next if I want to start building this?
> Are there any reference architectures that help me build this?
> Be sure to link to the Architecture Center, to related architecture guides and architectures.

- Examples:
  - [Artificial intelligence (AI) - Architectural overview](/azure/architecture/data-guide/big-data/ai-overview)
  - [Choosing a Microsoft cognitive services technology](/azure/architecture/data-guide/technology-choices/cognitive-services)
  - [What are Azure Cognitive Services?](/azure/cognitive-services/what-are-cognitive-services)
  - [What is Language Understanding (LUIS)?](/azure/cognitive-services/luis/what-is-luis)
  - [What is the Speech service?](/azure/cognitive-services/speech-service/overview)
  - [What is Azure Active Directory B2C?](/azure/active-directory-b2c/overview)
  - [Introduction to Bot Framework Composer](/composer/introduction)
  - [What is Application Insights](/azure/azure-monitor/app/app-insights-overview)
  - [Chatbot for hotel reservations](/azure/architecture/example-scenario/ai/commerce-chatbot)
  - [Build an enterprise-grade conversational bot](/azure/architecture/reference-architectures/ai/conversational-bot)
  - [Speech-to-text conversion](/azure/architecture/reference-architectures/ai/speech-ai-ingestion)

## Related resources

> Are there any relevant case studies or customers doing something similar?
> Is there any other documentation that might be useful?
> Are there product documents that go into more detail on specific technologies that are not already linked?

<!-- links -->

[calculator]: https://azure.com/e/
```

Markdown and YML templates for solution ideas

4/26/2021 • 3 minutes to read

Use these Markdown and YML templates to create a solution idea.

For guidance, see Create a solution idea.

Each article must have two files. Follow the instructions for browse format authoring at Authoring content for the Azure Architecture Center Browser.

Call this file *<solution-idea>.yml*:

```
### YamlMime:Architecture
metadata:
  title: <Article title, which becomes the title metadata>
  titleSuffix: Azure Solution Ideas
  description: <Write a 100-160 character description that ends with a period and ideally starts with a call
to action. This becomes the browse card description.>
  author: <contributor's GitHub username. If no GitHub account, use EdPrice-MSFT>
  ms.author: <contributor's Microsoft alias. If no alias, leave out this metadata as it will be
automatically set to pnp by the docfx.json.>
  ms.date: <publish or major update date - mm/dd/yyyy>
  ms.topic: conceptual
  ms.service: architecture-center
  ms.subservice: solution-idea
azureCategories:
  - <choose at least one category from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#azure-categories>
  - <there can be more than one category>
products:
  - <choose 1-5 products from the list at https://review.docs.microsoft.com/en-
us/help/contribute/architecture-center/aac-browser-authoring?branch=master#products>
  - <don't use "azure" unless there is nothing else>
  - <1-5 products>
name: <The H1 title is a noun phrase that describes the scenario. Avoid naming the scenario after the Azure
technologies that are used. Example: "Insurance claim image classification on Azure">
summary: <Write a summary. Can be same as description.>
thumbnailUrl: /azure/architecture/browse/thumbs/<browse-png-filename>.png
content: |
  [!include[](<solution-idea>-content.md)]
```

Call this file *<solution-idea-content>.md*:

```
> The H1 title is a noun phrase that describes the scenario. Don't enter it here, but as the **name** value
in the corresponding YML file.>
> Include the solution idea header note at the top of the solution idea. This adds clarification why this is
a scaled-back architecture (and provides consistency with our other SIs)...

[!INCLUDE [header_file](../../../includes/sol-idea-header.md)]

Introductory section (no heading)
The introduction contains:

- A paragraph that describes what the solution does (the domain)
- A paragraph that contains a brief description of the main Azure services that make up the solution. This
paragraph should convey the Azure value proposition, not a complete description of the architecture.

## Potential use cases

> Are there any other use cases or industries where this would be a fit?
```

```
> How similar or different are they to what's in this article?

These other uses cases have similar design patterns:

- List of example use cases

## Architecture

_Architecture diagram goes here_

After the architecture diagram, include a numbered list that describes the data flow or workflow through the
solution. Explain what each step does. Start from the user or external data source, and then follow the flow
through the rest of the solution.

### Components

A bullet list of components in the architecture (including all relevant Azure services) with links to the
product service pages. This is for lead generation (what business, marketing, and PG want). It helps drive
revenue.

> Why is each component there?
> What does it do and why was it necessary?

- Example: [Resource Groups][resource-groups] is a logical container for Azure resources.  We use resource
groups to organize everything related to this project in the Azure console.

## Next steps

Links to articles on the AAC, and other Docs articles. Could also be to appropriate sources outside of Docs,
such as GitHub repos or an official technical blog post.

Examples:
* [Artificial intelligence (AI) - Architectural overview](/azure/architecture/data-guide/big-data/ai-
overview)
* [Choosing a Microsoft cognitive services technology](/azure/architecture/data-guide/technology-
choices/cognitive-services)
* [What are Azure Cognitive Services?](/azure/cognitive-services/what-are-cognitive-services)

## Related resources

Another optional link list, in bulleted form. If you have several links to other articles in Docs, include
those in "Next steps" and use "Related resources" for related architecture guides and architectures.

Here is an example section:

Fully deployable architectures:

* [Chatbot for hotel reservations](/azure/architecture/example-scenario/ai/commerce-chatbot)
* [Build an enterprise-grade conversational bot](/azure/architecture/reference-
architectures/ai/conversational-bot)
* [Speech-to-text conversion](/azure/architecture/reference-architectures/ai/speech-ai-ingestion)![image]
(https://user-images.githubusercontent.com/13895622/116135227-b73cac00-a685-11eb-92d3-003350ba6604.png)
```

Old markdown templates

3/24/2021 • 4 minutes to read

These are old templates moved from the AAC repo. Do not use them. For more information, see What is the Azure Architecture Center?

```
---
title: <Use the H1 title of your article. Maximum recommended length is 60 characters>
titleSuffix: Azure Reference Architectures
description: <A brief summary of your article. This is the description that appears in search engine results. Maximum recommended length is 150-160 characters.>
author: <github username>
ms.date: <publish or update date - mm/dd/yyyy>
ms.service: architecture-center
ms.topic: conceptual
ms.subservice: reference-architecture
---

[NOTE: You can find detailed guidelines about creating reference architectures [here](../reference-architectures.md).]

# Article title

_Brief introduction goes here._ [**Deploy this solution**.](#deploy-the-solution)

![alt text](./media/folder_name/architecture-diagram.png)

_Download a [Visio file](https://arch-center.azureedge.net/architecture.vsdx) that contains this architecture diagram. This file must be uploaded to `https://arch-center.azureedge.net/`_

## Architecture

The architecture consists of the following components:

- **Thing 1**. Description

- **Thing 2**. Description

## Recommendations

The following recommendations apply for most scenarios. Follow these recommendations unless you have a specific requirement that overrides them.

_Include considerations for deploying or configuring the elements of this architecture._

## Scalability considerations

_Identify and address scalability concerns relevant to the architecture in this scenario._

## Availability considerations

_Identify and address availability concerns relevant to the architecture in this scenario._

## Manageability considerations

_Identify and address manageability concerns relevant to the architecture in this scenario._

## Security considerations

_Identify and address security concerns relevant to the architecture in this scenario._
```

```
## Deploy the solution

_Describe a step-by-step process for implementing the reference architecture solution. Best practices are to add the solution to GitHub, provide a link (use boilerplate text below), and explain how to roll out the solution._

A deployment for a reference architecture that implements these recommendations and considerations is available on [GitHub](https://www.github.com/path-to-repo).

1. First step
2. Second step
3. Third step ...
```

```
---
title: <Article title>
titleSuffix: Azure Example Scenarios
description: <Write a 100-160 character description that ends with a period and ideally starts with a call to action.>
author: <github username>
ms.date: <publish or update date - mm/dd/yyyy>
ms.service: architecture-center
ms.topic: conceptual
ms.subservice: example-scenario
---

# Article title

The title is a noun phrase that describes the scenario.

> Example: "Insurance claim image classification on Azure"

Avoid naming the scenario after the Azure technologies that are used.

(Introductory section - no heading)

> This should be an introduction of the business problem and why this scenario was built to solve it.
>> What industry is the customer in?
>> What prompted them to solve the problem?
>> What services were used in building out this solution?
>> What does this example scenario show? What are the customer's goals?

> What were the benefits of implementing the solution described below?

## Potential use cases

> Are there any other use cases or industries where this would be a fit?
> How similar or different are they to what's in this article?

These other uses cases have similar design patterns:

- List of example use cases

## Architecture

_Architecture diagram goes here_

> What does the solution look like at a high level?
> Why did we build the solution this way?
> What will the customer need to bring to this? (Software, skills, etc?)
> Is there a data flow that should be described?

### Components

A bullet list of components in the architecture (including all relevant Azure services) with links to the product documentation.

> Why is each component there?
```

```
> What does it do and why was it necessary?

- Example: [Resource Groups][resource-groups] is a logical container for Azure resources. We use resource groups to organize everything related to this project in the Azure console.

### Alternatives

Use this section to talk about alternative Azure services or architectures that you might consider for this solution. Include the reasons why you might choose these alternatives.

> What alternative technologies were considered and why didn't we use them?

## Considerations

> Are there any lessons learned from running this that would be helpful for new customers?  What went wrong when building it out?  What went right?
> How do I need to think about managing, maintaining, and monitoring this long term?

### Availability

### Scalability

> Are there any size considerations around this specific solution?
> What scale does this work at?
> At what point do things break or not make sense for this architecture?

### Security

> Are there any security considerations (past the typical) that I should know about this?

## Deploy this scenario

> (Optional, but greatly encouraged)
>
> Is there an example deployment that can show me this in action? What would I need to change to run this in production?

## Pricing

> How much will this cost to run?
> Are there ways I could save cost?
> If it scales linearly, then we should break it down by cost/unit. If it does not, why?
> What are the components that make up the cost?
> How does scale affect the cost?
>
> Link to the pricing calculator with all of the components in the architecture included, even if they're a $0 or $1 usage.
> If it makes sense, include small/medium/large configurations. Describe what needs to be changed as you move to larger sizes

## Next steps

> Where should I go next if I want to start building this?
> Are there any reference architectures that help me build this?

## Related resources

> Are there any relevant case studies or customers doing something similar?
> Is there any other documentation that might be useful?
> Are there product documents that go into more detail on specific technologies not already linked

<!-- links -->

[calculator]: https://azure.com/e/
```

Cloud Adoption Framework content review process

3/5/2021 • 2 minutes to read

Prerequisite requirements

The following steps are required.

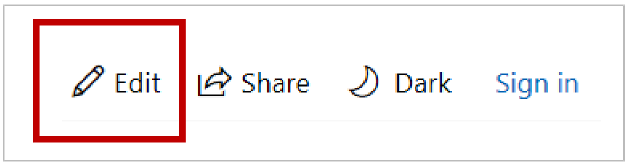
1. If you don't already have a GitHub account, create one.
2. Link your GitHub account to your Microsoft account (@microsoft.com):
   a. Go to Microsoft Open Source organizations, and sign in.
   b. Complete the steps to link your accounts. If your accounts are already linked, you see **Link OK**.
3. Join the **MicrosoftDocs** organization:
   a. Go to Microsoft Open Source organizations.
   b. In the **Available Microsoft GitHub organizations** section (right-hand side), scroll down until you see **MicrosoftDocs**.
   c. Select **Join**.

Review content and submit suggested changes that the author can review

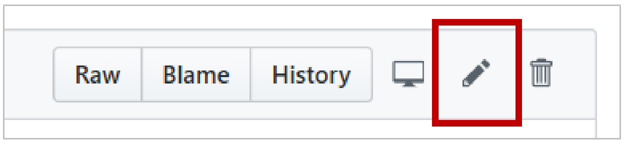
Watch this 2-minute video to see how to edit staged content.

Or follow the steps outlined below.

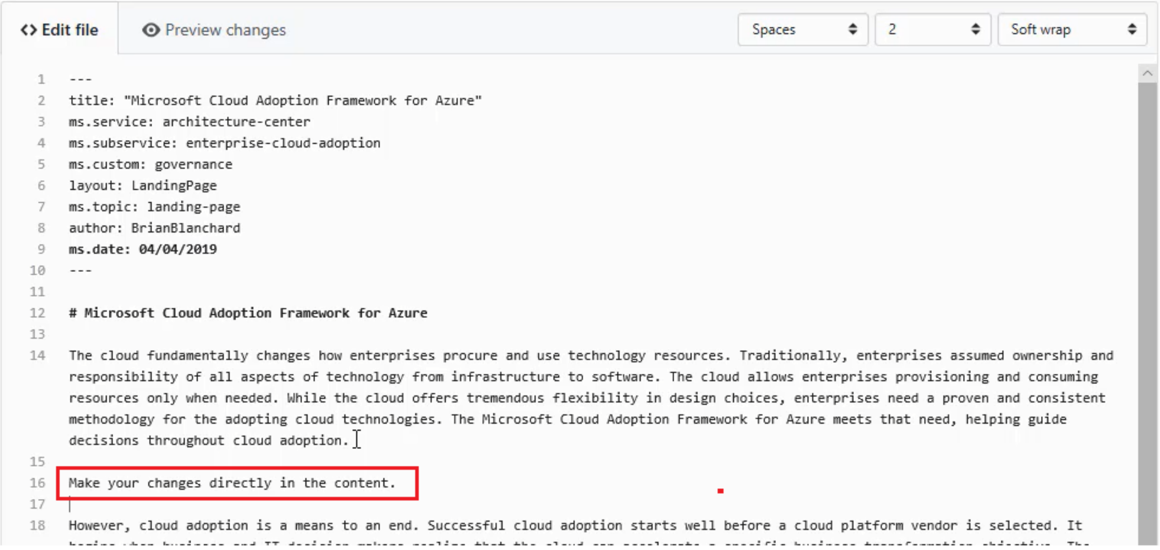
1. If you were provided with direct links, go to the links provided. If you weren't provided a direct link go to https://aka.ms/CAF/StagingSite. NOTE: Don't bookmark this URL as the staging site changes with each review cycle – use the aka link to get the latest.
2. Click the **Edit** link at the top of the page. This takes you to the GitHub source of the article.



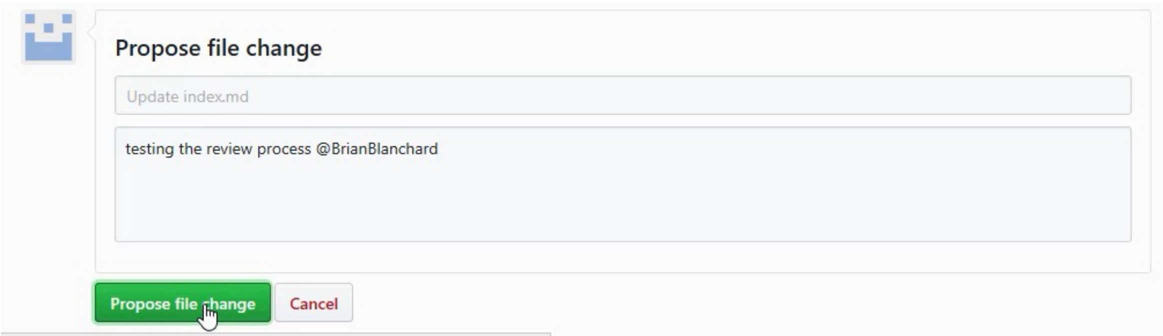
3. Click the Edit button (a pencil) in the GitHub source.



4. Make your changes directly in the content. You can add, remove or change text. But you must make an alteration to the text before you can leave a comment and propose a change.



5. After you make your change, title the change request and leave a comment describing your changes in the text boxes at the bottom of the page. Then click **Propose file change** to start a pull request.



6. On the **Comparing changes** page, review the changes that you have made, then, click **Create pull request**.

7. On the **Open a pull request** page, click **Create pull request** again.

Watch for GitHub notifications. The author may @mention you in the pull request to have a conversation about the changes.

It's OK to propose changes even if you are not 100% sure of the change. Your pull request is the start of a conversation.

See also

- Adding line comments to a pull request.
- Collaboration with Git and GitHub.

Troubleshooting

If you click a **Edit** in the review topic and you get a 404 message, you should verify two things:

- Are you actually signed into GitHub with the account you linked to your Microsoft identity?
- Are you a member of the MicrosoftDocs organization in GitHub. Visit the Open Source portal to verify your memberships.

Work tracking for Azure DevOps content

5/10/2021 • 2 minutes to read

The Azure DevOps content team plans and tracks their work in Azure Boards on the following backlogs and boards:

- **Azure DevOps Features backlog**: `https://dev.azure.com/mseng/TechnicalContent/_backlogs/backlog/Azure%20DevOps/Features?showParents=true`
- **Azure DevOps User Stories backlog**: `https://dev.azure.com/mseng/TechnicalContent/_backlogs/backlog/Azure%20DevOps/Stories?showParents=true)`
- **Azure DevOps User Stories board**: `https://dev.azure.com/mseng/TechnicalContent/_boards/board/t/Azure%20DevOps/Stories`

We track significant work for documenting new scenarios and tasks, and changes to the UI and feature support. We also review and update content to support accessibility, SEO, and general content freshness.

> **TIP**
>
> If you are adding new content or making significant changes, please include a member of the content team on this work to avoid duplication and to align with ongoing content strategies. You can find the member of the content team who oversees a specific content area in Azure DevOps repositories, content areas, content owners. Or email the Azure DevOps content team.

Add a User Story or Doc Defect Bug

For new content to document a new scenario or address a feature or UI change, use the following template:

- Azure DevOps User story

For a change to content to address an inaccuracy or doc defect, use the following template:

- Azure DevOps Content Bug

Some specific fields to provide:

- Area path: `TechnicalContent\Dev Tools and Services\Jill Reinauer Team\Azure DevOps`
- Iteration: `TechnicalContent\3WeekSprints\Future`
- Description: Explain the issue or requirement. Provide a link when possible. If there is a due date, explain why.

Additional resources

- Azure Boards (work tracking) documentation

Azure DevOps content review checklist

6/3/2021 • 3 minutes to read

Contributors and content owners can use this checklist when writing or reviewing content contributed to Azure DevOps repositories.

Other resources include:

- Writing checklist
- Microsoft Cloud Style Guide
- Term Studio
- Accessibility guidelines

Structure

- [ ] Is the content appropriate for a technical article, as opposed to a blog post or white paper?
- [ ] Does the introduction make clear what the article is about?
- [ ] Is it clear who the target audience is?
- [ ] Is the goal of the article clear? How does it help customers? What problem does it solve?
- [ ] Is there any unnecessary content that doesn't support the intent or help solve the problem?
- [ ] Does the article follow the standards for its content type, is it the right content type?
- [ ] Does the flow of the article logical?
- [ ] Does the article file name meet our conventions? Is it located in the right folder?

Metadata

- [ ] All required metadata filled out.
- [ ] Review metadata for SEO.
- [ ] Update the `ms.date` for:

- New articles.
- Articles that you've updated in a significant way.

- [ ] `monikerRange` check.

Text

- [ ] Review of staged content for any poorly formatted text.
- [ ] You provided Header/Platform/Version.
- [ ] Headers - No gerunds, sentence capitalization, short to cause least amount of **In this article** wrapping.
- [ ] Check all headers for sentence capitalization.

- [ ] Check all headers for use of active verbs in Quickstarts, Tutorials, and How-to guides.
- [ ] Use of **Prerequisites** section where needed.
- [ ] Modern voice

- Are the voice and tone of the text following modern voice guidelines?
- Is the content concise, conversational, and engaging--but not colloquial?

- [ ] A Tutorial has a checklist.
- [ ] You used Fictitious names as needed.

Terms and acronyms

- [ ] Check Term studio as needed around common use terms.
- [ ] Check suspect terms against the Microsoft Cloud Style Guide.
- [ ] Terminology check: are terms used and spelled correctly?
- [ ] Are technical terms and concepts introduced and explained?
- [ ] Check that you've spelled out acronyms on first usage.

SEO

- [ ] See Guidance for using the SEO cheat sheet.
- [ ] Do these parts of the article reflect the words the customer will use when searching for this content:

- Metadata title
- Metadata description
- Page title (H1)
- Summary (first paragraph)

- [ ] Do those parts of the article clearly capture the relevance of the article?
- [ ] Are the expected search keywords and phrases used in the content?

Art and images

- [ ] See Create and format screenshots in documentation.
- [ ] Review staged content for any poorly formatted images.
- [ ] Check that screenshot size is within height (> 750 pixels) and width (> 725 pixels) limits.
- [ ] You've provided well-formed Alt text for screenshots.
- [ ] Call outs in screenshots use established color and format.
- [ ] All screenshots have a gray border.
- [ ] Screenshots use fictitious names and don't display sensitive information.

Tags: Moniker Tags and Conceptual

- [ ] Have you applied moniker tags correctly?
- [ ] Review staged content for each version to check format and that no moniker labels appear in the text.

- [ ] Are tabbed conceptual tabs applied correctly using the existing tags defined for Azure DevOps?

Cross linking

- [ ] Are there helpful links to related resources?
- [ ] Quickstarts and tutorials should contain **Next steps** section.
- [ ] Do all external links resolve correctly? Do any need update?
- [ ] Are internal links formatted correctly?
- [ ] Should you create links from anywhere else or other product docsets to this article?
- [ ] Disambiguation – Is there a need to refer the user to some other article to disambiguate terms? For example Work Item Tags versus Git Tags, Azure DevOps Projects and Azure DevOps Project.

TOC

- [ ] Article added to the main TOC YAML file.
- [ ] Article added to secondary TOC YAML files where appropriate with shared TOC tagging.
- [ ] Review the TOC entires check that you didn't use gerunds.
- [ ] TOC entry makes sense even if shortened from the article title.
- [ ] Filtering TOC yields results when using article-specific terms.

Aliases for Azure DevOps technical content

5/10/2021 • 2 minutes to read

Azure DevOps contributors to technical documentation need to belong to certain aliases and be aware of others for information sharing.

ALIAS NAME	ALIAS	REQUIRED?	PURPOSE
Content & Learning Azure DevOps Content Devs	AzDevContentDev	For FTE content developers	FTE communications for content developers (CPub discipline) who oversee major content areas and docsets for Azure DevOps
Azure DevOps Content Contributors	vsocontc		For PMs and Devs who contribute to Azure DevOps Content
Azure DevOps	azuredevops		Email notifications about Azure DevOps

Azure DevOps azure-devops-docs-pr repo and branch policies

The following groups are added to the Branch Policies for one or more folders in the **azure-devops-docs-pr** repository. By joining a group, you'll be able to approve your PRs so that they are merged into the master branch. You join these groups by going to IDWEB(`https://idweb/IdentityManagement/default.aspx`).

SECURITY GROUP	WEB ID ALIAS	OWNER	PURPOSE
Azure Boards Content Approvers	ts-work-cnt-approve	kaelli	For content contributors to **Azure Boards** and the **/boards**, **/organizations/settings/work** and **/reference** folders
Azure DevOps Artifacts Approvers	az-artifacts-approve	phwilson	For content contributors to the **/artifacts** folder
Azure DevOps Collaborate Content Approvers	ts-collbrt-cnt-appro	kaelli	For content contributors to the **/project** and **/notifications** folders
Azure DevOps Release Notes	AzDevRelNotesApprove	alexn	For content contributors to the **/release-notes** folder
Azure DevOps Report Content Approvers	ts-repor-cnt-approve	kaelli	For content contributors to the **/report** folder
Azure DevOps Setup-admin Approvers	vsts-setup-approvers	kaelli	For content contributors to the **/organizations** folder

SECURITY GROUP	WEB ID ALIAS	OWNER	PURPOSE
Team Services CI/CD Content Approvers	ts-cicd-doc-approve	Julia.KullaMader	For all contributors to **Azure Pipelines** content and the **/pipelines** and **/deploy-azure** folders
Team Services Integrate Approvers	ts-integrate-approve	chcomley	For all contributors to content to the following folders **/integrate**, **/extend**, **/service-hooks**, and **/marketplace**
[TechnicalContent]/Azure DevOps Test Content Approvers	N/A	sdanie	For content contributors to the **/test** folder
DevOps Doc Approvers	vsts-vc-docapprovers	sdanie	For content contributors to the **/repos** folder

These folders are missing from Branch policies: /demo-gen, /java,

Azure DevOps azure-devops-server-docs-pr repo and branch policies

The following groups are added to the Branch Policies for one or more folders in the **azure-devops-server-docs-pr** repository. By joining a group, you'll be able to approve your PRs so that they are merged into the master branch.

SECURITY GROUP	WEB ID ALIAS	OWNER	PURPOSE
Azure DevOps TFS Content Approvers	tfs-setup-approvers	elbatk	For content contributors to the tfs-docs-pr repository

Azure DevOps repositories, content areas, content owners

11/2/2020 • 2 minutes to read

This article provides specific guidance for Microsoft employees to contribute to Azure DevOps documentation accessible from here:

Git and GitHub repositories

DOCUMENTATION	REPOSITORY (PRIVATE)	GITHUB REPOSITORY (PUBLIC)
Azure DevOps Documentation	azure-devops-docs-pr	MicrosoftDocs/vsts-docs
Azure DevOps Server Documentation	azure-devops-server-docs-pr	Not set up
DevOps Resource Center	devops-learn-pr	Not set up
Azure DevOps Project Documentation	MicrosoftDocs/azure-docs-pr/articles/devops-project	MicrosoftDocs/azure-docs/articles/devops-project/
Azure DevOps CLI	MicrosoftDocs/vsts-docs-cli-python	Not applicable
REST APIS	TBD (Owned by the Product team)	Not applicable

Main product docsets

DOCSET	FOLDER	CONTENT OWNER
New User Guide	/user-guide	Chrystal Comley (chcomley);
Azure Artifacts (formerly Package Management	/artifacts	Rami Bououni (ramiMSFT)
Azure Boards **Reporting & Analytics**	/report	Kathryn Elliott (kaelli)
Azure Repos	/repos	Tier 3 support
Azure Pipelines	/pipelines	Steve Danielson (sdanie) Julia Kulla-Mader (Julia.KullaMader)
Azure Test Plans	/test	Tier 3 support

Analytics and social tools docsets

DOCSET	FOLDER	CONTENT OWNER
Reporting	/report	Kathryn Elliott (kaelli)
Notifications	/notifications	Chrystal Comley (chcomley)
Wikis	/project/wiki	Chrystal Comley (chcomley)
Search	/project/search	Kathryn Elliott (kaelli)
Feedback	/project/feedback	Kathryn Elliott (kaelli)

Settings and administration docset

DOCSET	FOLDER	CONTENT OWNER
Manage organizations	/organizations/accounts/	Chrystal Comley (chcomley)
Billing	/organizations/billing	Chrystal Comley (chcomley)
Manage projects	/organizations/projects/	Chrystal Comley (chcomley)
Security, Access, & Billing	/organizations/security	Kathryn Elliott (kaelli)
All Settings	/organizations/settings	Chrystal Comley (chcomley)
Scale and configure teams	/organizations/settings/scale/	Kathryn Elliott (kaelli)
Work Item Customization	/organizations/settings/work/reference	Kathryn Elliott (kaelli)
Azure DevOps Server Administration	azure-devops-server-docs-pr repo	Kathryn Elliott (kaelli)
Data Warehouse & SQL Reports	/report/sql-reports	Kathryn Elliott (kaelli)

Developer docsets

DOCSET	FOLDER	CONTENT OWNER
Custom Extensions	/extend	Chrystal Comley (chcomley)
Marketplace	/marketplace	Chrystal Comley (chcomley)
Service Hooks	/service-hooks	Chrystal Comley (chcomley)
REST APIs	TBD	TBD

Deprecated resources

Azure DevOps Archive documentation

DOCSET	FOLDER	CONTENT OWNER
SharePoint Integration	/report/sharepoint-dashboards	Kathryn Elliott (kaelli)
TFS-Project Server integration	/reference/tfs-ps-sync	Kathryn Elliott (kaelli)

Azure DevOps metadata guidance

6/16/2021 • 4 minutes to read

We invest a lot to create content for users, decision-makers, influencers, and potential customers. Metadata multiplies that investment by:

- Helping search engines display our content in search results to help our customers.
- Guiding customers to the information they seek, so they can finish tasks with our services and products.
- Enabling us to track and measure how well our content performs and what needs improvement.

General guidance provided here:

- Metadata overview
- Metadata attributes

Sample Azure DevOps metadata

All articles include metadata at the top. You delimit metadata with three dashes `---` as shown in the following example:

```
---
title: Add built-in charts to a team dashboard
titleSuffix: Azure DevOps & TFS
description: Add system-generated charts or query-based charts to a team dashboard for Azure DevOps or Team Foundation Server
ms.prod: devops
ms.technology: devops-analytics
ms.reviewer: greggboe
ms.author: kaelli
author: KathrynEE
ms.topic: quickstart
monikerRange: '>= tfs-2013'
ms.date: 09/20/2018
---
```

For an `/includes` file, provide this metadata:

```
---
ms.topic: include
---
```

Required tags

TAG	USAGE	EXAMPLE
`title`	SEO focused title, 60 total char max, including spaces Docs Build adds the site identifier, Microsoft Docs, to the end of each title.	`Add built-in charts to a team dashboard.`

TAG	USAGE	EXAMPLE
`description`	SEO focused description. It's a minimum 115 characters long and a maximum 145 characters, including spaces. Use important keywords in the first 100 characters.	`Add system-generated charts or query-based charts to a team dashboard.`
`ms.prod`	Enter the value associated with the product.	`devops`
`ms.technology`	Enter a value associated with the content area.	See **ms.technology** list below.
`ms.topic`	Enter a value that represents the article type.	See **ms.topic** list below.
`manager`	Microsoft alias of Content team manager. `ms.manager` works as well.	`douge`
`ms.author`	Microsoft alias of the primary writer of the content area.	`alewis`
`author`	GitHub alias of the primary writer of the content area.	`alewis`
`ms.date`	Date of first publish or with a significant update or change in content. Follow the date pattern shown in the example. Enter the `ms.date` value without time. Docs Build interprets the time as 0:00 in the UTC time zone. Docs converts the date so it's displayed to users in their own time zone.	`09/20/2018`

For more SEO guidance on titles and description, see the Guidance for using the SEO cheat sheet.

Optional tags

TAG	USAGE	EXAMPLE

`titleSuffix`	Use to specify another site identifier associated with your product brand. Product name that will appear in the browser tab followed by **| Microsoft Docs**.	`Azure DevOps`
		or
		`Azure DevOps Services`

`ms.reviewer`	Enter the alias of the primary PM or Dev who reviews or helps author the content.	`greggboe`
`ms.assetid`	GUID, not required.	`B080CEFA-4D94-44B2-99E3-0E3E85616D04`

TAG	USAGE	EXAMPLE
`ROBOTS`	Crawl instruction for the page for web crawlers. The default value is `INDEX, FOLLOW`. All possible values are: • `INDEX, FOLLOW` • `NOINDEX, NOFOLLOW` • `INDEX, NOFOLLOW` • `NOINDEX, FOLLOW`	
`ms.custom`	Text field. You're allowed to add multiple values separated by commas. (**NOTE**: Don't change any `devx-track-*` tags. They're used by the Developer Experience team to track groups of articles across services and repositories. Contact B. Kess with any questions.)	
`ms.devlang`	You can find a list of values in the Taxonomies for Docs article.	`csharp, javascript`
`monikerRange`	Specifies the versions that apply to the article. For the home docset, the article will appear in the TOC only for the versions specified in the range.	`monikerRange: 'vsts'`

ms.technology (for ms.prod: devops)

MS.TECHNOLOGY	TECHNOLOGY NAME STRINGS
`devops-analytics`	Analytics
`devops-accounts`	Accounts
`devops-agile`	Agile
`devops-artifacts`	Artifacts
`devops-billing`	Billing
`devops-code-git`	Repos (Git)
`devops-code-tfvc`	Repos (TFVC)
`devops-collab`	Search and Collaboration, Wiki
`devops-cicd`	Pipelines
`devops-ecosystem`	Extensibility and Integration

MS.TECHNOLOGY	TECHNOLOGY NAME STRINGS
`devops-learn`	Azure DevOps Learn
`devops-marketplace`	Marketplace
`devops-new-user`	New User Guide and Navigation
`devops-public-projects`	Public Projects (pending)
`devops-ref`	Azure DevOps API Reference
`devops-security`	Security and Permissions
`devops-settings`	Settings
`devops-test`	Test
`devops-whitepapers`	DevOps Whitepapers

ms.topic values

MS.TOPIC VALUE	USAGE
`article`	Useful content that isn't specific to a product scenario, like migration or data security. For example, whitepapers like Azure DevOps Technical Articles.
`conceptual`	Any content linked in a TOC that isn't marked in metadata. For example: • `Quickstart` • `Tutorial` • `Sample` • `Reference` • `Overview`
`contributor-guide`	Any content authored specifically for inclusion in a content contributor guide.
`include`	Use in **[!INCLUDE]** files only. It's required so Docs Build, validation, and other tools to work correctly.
`interactive-tutorial`	Interactive content.
`overview`	Overview or User Guide articles. Typically live under an **Overview** node in a TOC, or high-level general article in a New User Guide or Onboarding guide.
`quickstart`	Any authored content placed under a **Quickstart** node in a TOC *and* that follows the specific guidelines and template for a Quickstart article.

MS.TOPIC VALUE	USAGE
`reference`	Any authored (not autogenerated) content placed under a **Reference** node in a TOC.
`sample`	Any authored (not autogenerated) content placed under a **Samples** or **Examples** node in a TOC.
`troubleshooting`	Any authored content placed under a **Troubleshooting** node in a TOC AND that follows the specific guidelines and template for a Troubleshooting article.
`tutorial`	Any authored content placed under a **Tutorials** node in a TOC *and* that follows the specific guidelines and template for a Tutorial article.

Azure DevOps Open Publishing workflow

12/13/2019 • 2 minutes to read

Unlike the majority of Microsoft Docs content that is maintained in the MicrosoftDocs GitHub repo, Azure DevOps content is maintained in several Git repos in Azure Repos. For a list of repos, see Azure DevOps repositories, content areas, content owners.

Workflow to author and publish content

> **TIP**
> If you are adding new content or making significant changes, please include a member of the content team on this work to avoid duplication and to align with ongoing content strategies. You can find the member of the content team who oversees a specific content area in Azure DevOps repositories, content areas, content owners. Or email the Azure DevOps content team.

The current process for authoring and publishing content is:

1. Make sure you your GitHub account is joined to the Microsoft and MicrosoftDocs organizations.
2. Clone the azure-devops-docs-pr, in Visual Studio or Visual Studio Code.
3. Create a new local branch from the `/master` branch. You can do this from Visual Studio or Visual Studio Code. The new working branch will be for your content additions and updates.
4. Make your changes to the content and commit them to this working branch.
   a. If you are working from a clone repo, push them to the remote repo for your working branch
   b. Create a Pull Request from your branch to the master branch.
   c. Once you have committed your content to the (remote) repo, a build will kick off. Builds can take 10-15 min, so if you're not sure if you're in the build queue, you can check the build queue portal - You need to login with your GitHub alias to reach this page. You will receive an email – probably two emails – that will tell you where the content is staged and if there are broken links – a build report. Link to build report is in the email.
   d. Check that your content displays as expected on the staging site.
   e. Once your content displays correctly, and there are no build errors, it is ready for review. Create a comment with the text "#sign-off" to trigger the PR Merger review.
   f. PR Merger approves some PRs automatically and requests reviews for others. After the PR is approved, PR Merger will merge the PR.
5. Twice a day or more often a PR from master to live is made, at which time your changes will get published.

Contributor's guide resources

- **Follow content standards**
  - Write an overview
  - Write a quickstart
  - Write a tutorial
  - Write a sample
  - Edit a TOC

- Improve your writing
  - Follow writing principles
  - Review the Microsoft style guide

Additional notes

Staged content

- Any time you push your working branch to the cloud repo, the branch builds. However, you get email when the build completes along with a link to an error report.
- You get staging with a working TOC and links at:

  ```
  https://review.docs.microsoft.comazure/devops/File-path-and-name?branch=branchname
  ```

  For example:

  ```
  https://review.docs.microsoft.comazure/devops/boards/boards/index?branch=master
  ```

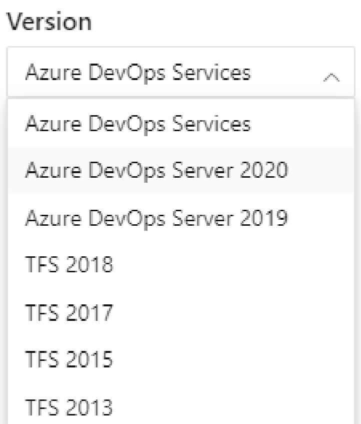
- When you push your branch, you can see where you are in the OPS Portal build queue. You'll receive one or two emails when the build completes for a push or PR. Dev is working on better build filtering to avoid the two unnecessary builds when you PR.
- Each area has its own TOC that uses markdown syntax within a yml file (toc.yml), and the TOC on the published site only shows one area's TOC at a time.
- The UI around navigating across hubs changes to a "breadcrumb" top nav bar where each element in the "breadcrumb" path is a drop-down to other sub nodes at that level.

Azure DevOps versioning guidance and scenarios

11/2/2020 • 4 minutes to read

Many Azure DevOps articles require the application of moniker tags to support versioning. Users can select between Azure DevOps Services and the version of TFS (will become Azure DevOps Server) they are using. The content displayed will only show that content which applies to their selected version, based on the monikers defined within the article.

Users select the Azure DevOps Services or Azure DevOps/TFS version from the left-hand TOC to scope the content that is displayed. The **azure-devops-docs-pr** repository contains content for the cloud product, Azure DevOps Services, and the on-premises product, Azure DevOps Server (previously named Team Foundation Server). Versions for TFS 2013 and all later versions are supported.



As much as possible, we want to minimize topic versioning. That is, we want each article to address whatever platform/versions that support the task or scenario covered by the content.

Features and scenarios introduced during an update, such as TFS 2017.2, are handled through notes. These notes are encapsulated within monikers specific to the main version.

As much as possible, we want to maintain the same article (with URL juice) and allow it to cover features introduced first on Team Services and then later into a TFS Update.

- Articles should be scenario and task-oriented, features are secondary
- Only when the tasks diverge fully (or lets say 60% or more) between platforms or versions should you consider branching an article to address the differences.

Azure DevOps monikers

Azure DevOps and product versions are associated with monikers. The following table indicates the current monikers that you can use in Azure DevOps articles to control versioning:

MONIKER	PRODUCT/VERSION
azure-devops	Azure DevOps product version
azure-devops-2020	Azure DevOps Server 2020
azure-devops-2019	Azure DevOps Server 2019

MONIKER	PRODUCT/VERSION
tfs-2018	Team Foundation Server 2018
tfs-2017	Team Foundation Server 2017
tfs-2015	Team Foundation Server 2015
tfs-2013	Team Foundation Server 2013

Versioning is implemented at three levels: Zone, Article, and Folder.

Only major level versioning is supported. If new features are added to a version update, for example TFS 2017.2, then you need to make that explicit in the content.

Zones within articles

Inside an article, you can version specific **zones** of text. Consider the following taken from the Add and manage dashboards. Zone monikers uses the same moniker range concepts, but they apply to zones inside the article.

```
::: moniker range="tfs-2015"
> [!NOTE]
> Multiple team dashboards and the [widget catalog](widget-catalog.md) are available from TFS 2015.1 or
later versions. For TFS 2015 and earlier versions, you don't have access to multiple team dashboards.
Instead, your home page serves as a [single team dashboard](team-dashboard.md). For information on
SharePoint dashboards, see [Project portal dashboards](../sharepoint-dashboards/project-portal-
dashboards.md).
::: moniker-end

::: moniker range="tfs-2017"
> [!NOTE]
> For information on SharePoint dashboards, see [Project portal dashboards](../sharepoint-
dashboards/project-portal-dashboards.md).
::: moniker-end


::: moniker range="azure-devops"
[!INCLUDE [temp](../includes/dashboard-prerequisites-vsts.md)]
::: moniker-end

::: moniker range=">= tfs-2017 <= tfs-2018"
[!INCLUDE [temp](../includes/dashboard-prerequisites-tfs-2017-18.md)]
::: moniker-end

::: moniker range="tfs-2015"
[!INCLUDE [temp](../includes/dashboard-prerequisites-tfs-2015.md)]
::: moniker-end
```

Article level metadata

Article-level versioning is applied when the article applies to select versions. You add a **monikerRanger** metadata value to the article. This overrides the folder-level defaults. Consider the following portion of metadata:

```
ms.topic: quickstart
monikerRange: '>= tfs-2015'
ms.date: 07/21/2018
---
```

In this example, the article applies to all versions of TFS 2015 and greater (`>= tfs-2015`) including Azure DevOps Services

In the following example, the article applies to only versions TFS 2013 and TFS 2015.

```
monikerRange: '>= tfs-2013 <= tfs-2015'
ms.date: 08/11/2016
---
```

Folder level defaults

Each folder in the repository has a versioning default assigned in the **vsts-docs-pr/docs/docfx.json** file in the repository. This means that new articles are automatically versioned just by their location.

Folder-level monikers have been applied to the following folders:

```
    "fileMetadata": {
      "monikerRange": {
        "report/admin/**": "< azure-devops",
        "report/extend-analytics/**": ">= azure-devops-2019",
        "report/powerbi/**": ">= azure-devops-2019",
        "report/sql-reports/**": "< azure-devops",
        "reference/witadmin/**": "< azure-devops",
        "reference/mapping/**": "<= tfs-2018"
```

Platform version header

On many articles, you'll see a platform-version header right below the article title. As the version selector is often ignored by our customers, we want to give them another visual cue as to which platforms and versions an article applies to. Add this type of header to your article right below the title.

You should use the INCLUDE files located under /docs/includes folder or in the /includes folder for your specific vertical.

For articles that apply only to Azure DevOps Services:
Azure DevOps Services
`\docs\includes\version-vsts-only.md`

For articles that apply only to Azure DevOps Services and Azure DevOps Server 2020:
Azure DevOps Services | Azure DevOps Server 2020
`\docs\includes\version-cloud-plus-2020.md`

For articles that span all product versions:
Azure DevOps Services | Azure DevOps Server 2020 | Azure DevOps Server 2019 | TFS 2018 - TFS 2013
`\docs\includes\version-vsts-tfs-all-versions.md`

For articles that are specific to select versions of TFS:
TFS 2017 | TFS 2015 | TFS 2013 `\docs\includes\version-tfs-2013-2017.md`

For articles that are specific to a product/service, such as Azure Boards:

Azure Boards | Azure DevOps Server 2020 | Azure DevOps Server 2019 | TFS 2018 - TFS 2013

`\docs\boards\includes\version-vsts-tfs-all-versions.md`

General guidance for applying monikers

- Do apply article monikers where needed
- Do apply zone monikers where needed
- Don't nest monikers
- Don't apply monikers around conceptual tabs

When a new feature is introduced, it is usually on the cloud version first, so you apply monikers accordingly.

When a new version of Azure DevOps Server is released, you need to review the moniker applications and update accordingly.

Related articles

- Conceptual Versioning
- SQL versioning guidance and scenarios for writers

Azure DevOps tabbed conceptual tags usage and guidance

10/17/2019 • 2 minutes to read

Tabbed conceptual tags are used to provide different versions of content depending on selections made by the user. That selection might cover:

- Different features supported by the product, such as **New navigation** and **Previous navigation**
- Git repo versus TFVC repo
- Browser view versus a client view, such as Visual Studio or Team Explorer Everywhere
- OS differences
- New work item form versus old form

> **NOTE**
>
> Tabbed conceptual tags differ from versioning moniker tags which are used to support differences in content among different product versions.
>
> The zone pivot feature is often a competitive alternative to tabbed conceptual.

Syntax for usage

The syntax for using tabbed conceptual tags is simple, as shown in the following example. This syntax has been updated to support markdig build engine.

```
#### [Browser](#tab/browser)

<!--- Content here  -->

#### [Visual Studio](#tab/visual-studio)

<!--- Content here  -->

---
```

The last three dashes close out the tabbed conceptual entry.

Examples used in articles

- Uses **Browser** and **Visual Studio**:
  - Use work item templates, Markdown file
- Uses **New form**, **Old form**, **Visual Studio**, and **Team Explorer Everywhere**:
  - Link work items to support traceability and manage dependencies

Guidance on using tabbed conceptual tags

- If you use Tabbed Conceptual tags, you need to use the same set throughout the article. That is, you can't use different sets of tags or two tags in one place and three tags in another place within the same article.

Guidance on using Tabbed Conceptual tags and Conceptual Versioning Moniker tags

- Can use moniker tags within a Tabbed Conceptual tag, but not vice-versa; that is, you can't surround a Tabbed Conceptual set of tags with monikers

Tags to use

Here are the tags that are in use. Use these tags whenever possible. Wherever possible, we need to use the same tags everywhere. If you have to add new tags, add them to this list.

LABEL	TAG	NOTES, ARTICLE WHERE USED
New web form	new-web-form	/boards content
Old web form	old-web-form	/boards content
Designer	designer	/pipelines/build content
YAML	yaml	/pipelines content
Web	web	/pipelines/agents content
Windows	windows	/pipelines/agents content
Linux	linux	/pipelines/agents content
macOS	unix	/pipelines/agents content
Azure DevOps CLI	azure-devops-cli	/organizations/settings/set-area-path
Azure Repos or TFS repo	vsts	/pipelines/apps content
Azure Repos	gitvsts	/pipelines/apps content
GitHub repo	github	/pipelines/apps content
Install them during the build	apple-install-during-build	\pipelines
Preinstall them on a macOS build agent	apple-preinstall	\pipelines
Team Explorer Everywhere	tee	Used in a few /boards content
Visual Studio	visual-studio	/boards and /repos content
Command Line	command-line	/repos content
MSTest	mstest	/pipelines content
NUnit	nunit	/pipelines content
Batch	batch	/pipelines content

LABEL	TAG	NOTES, ARTICLE WHERE USED
Bash	bash	/pipelines content
PowerShell	powershell	/pipelines content
Shell	shell	/pipelines content
Browser	browser	Use to support the web portal view
HTTP	http	/extend content
C# (client library)	csharpclient	/extend content
C# (generic)	csharpgeneric	/extend content
Node.js (generic)	nodejsgeneric	/extend content
Legacy URLs	vsts	DUPLICATE USAGE - /artifacts/quickstarts
New URLs	azuredevops	/artifacts/quickstarts

Tags to not use

LABEL	TAG	NOTES
Web	web	Use **Browser** tab instead
Web Portal	vsts-tfs-web-portal	Use **Browser** tab instead

Azure DevOps terminology and branding guidance

3/15/2021 • 12 minutes to read

This article provides usage examples and guidance for using the brand names and specific terms for Azure DevOps content.

Our umbrella name is **Azure DevOps**. For the full official naming guidance, see Azure DevOps in the cloud style guide. But generally:

1. Prefer use of the service names, i.e. 'Azure Pipelines' when talking about the product
2. When talking about the collection of services, prefer 'Azure DevOps'.
3. When required, we will use Azure DevOps Services & Azure DevOps Server to disambiguate between the cloud & on-premises offerings (i.e. formerly VSTS & TFS) but where possible just use Azure DevOps as the umbrella term.

No abbreviations should be used (no ADS, ADOS etc), internally our shortened team name is AzDev but this should not be used externally or in social hashtags etc. Azure DevOps or #AzureDevOps are the only permitted forms. DevOps should always be CamelCase.

There is no need to prefix with Microsoft for the name, (i.e. you do not need to say Microsoft Azure Pipelines). There are no trademark symbols required in licenses or console output. However 'Azure' should always be preceded unless there are naming / repetition constraints (please ping Ori Zohar and Martin Woodward if you think you might have a reason to not precede with Azure)

Product/service names and usage

IMPORTANT

For some period of time, we need to address both the **New Navigation** elements and the **Previous Navigation** elements (previous names). So, when adding or editing content, think through the names used in the user interface based on the navigation steps you're documenting.

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS:	NOTES ON USAGE

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS:	NOTES ON USAGE
Azure DevOps		Umbrella name of Microsoft's DevOps solutions, which include 5 cloud services (collectively called Azure DevOps Services) and an on-premises product (currently called TFS, will become Azure DevOps Server in March 2019.	*To create an Azure DevOps organization with a personal Microsoft account or a work or school account, go to the Azure DevOps Services site.*	Limit usage as follows: - Azure DevOps organization, Azure DevOps project, Azure DevOps group. - Procedural steps documenting a user interface that uses Azure DevOps. - Otherwise, **DO NOT USE in docs until after TFS is rebranded to Azure DevOps Server in March 2019 unless you are documenting a user interface with "Azure DevOps" in the user interface string.**
Azure DevOps Services	Replaces: Visual Studio Team Services, VSTS, Team Services, Visual Studio ALM, Visual Studio Team Services (VSTS), Visual Studio Online, VSO	Umbrella suite name for the hosted or cloud offering.	*In this quickstart, you learn how to view your or another person's permissions defined in Azure DevOps Services or Team Foundation Server (TFS).*	- Use only to refer to *the collection of 5 services*. When referring to functionality or features of a specific service, use the name of that service. Do not find and replace previous names with Azure DevOps Services; it's important to identify the correct individual service. - Use to disambiguate the cloud offering from the on-premises offering. - Do not shorten or abbreviate (for example, "DevOps Services", "ADS", "ADOS", etc.)

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS:	NOTES ON USAGE
Azure DevOps Server	Team Foundation Server and TFS	Name of the on-premises Azure DevOps product.	Example 1 – You have documentation that references on-premises in general (no specific version). Continue to call on-premises "TFS" until March 2019. - Example 2 – After 9/5, you blog about a new feature on the service that will be in the next on-premises version. Say it will be available in Azure DevOps Server. - Example 3 – You have an error message that is specific to "TFS 2018 and earlier". Always keep this as "TFS" since we will not rebrand these shipped versions of TFS.	**We won't be making that change yet in the product or in the docs until we are close to launching Azure DevOps Server 2019.** - We will not go back and update TFS 2018 or earlier. TFS branding will remain for all versions from TFS 2013 through TFS 2018 (including TFS 2018.3). - **For things that reference TFS on-premises in general (clients, tools, documentation, blogs, etc.), we should keep them as "TFS" until Azure DevOps Server 2019 ships in March. Until then, the latest on-premises is still TFS 2018, so we should call it "TFS". For references to the next version of on-premises, you can use Azure DevOps Server.**

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS:	NOTES ON USAGE
Azure Pipelines	Replaces: Build & Release hub, build and release hub, Visual Studio Team Services Build Service, Visual Studio Team Services Build & Deployment Service	Name of the service/page that hosts the CI/CD functionality	*Go to the Azure Pipelines service in the web UI...*	- Always use "Azure Pipelines"; do not shorten to "Pipelines". - If you encounter phrasing like "CI/CD with VSTS", replace with "CI/CD with Azure Pipelines" rather than "CI/CD with Azure DevOps". The goal is to be Azure DevOps service specific now whenever possible based on the context of the doc. For more terminology specific to Azure Pipelines, see Azure Pipelines terms.
Azure DevOps Project(s)		A specific Azure service released long before the Azure DevOps brand.	*Configure continuous integration (CI) and continuous delivery (CD) for your .NET core or ASP.NET application with Azure DevOps Project.*	First reference: Azure DevOps Project(s) Subsequent references: DevOps Project(s) - Always capitalize Project. - Singular or plural Project depends on context. - Only use Azure DevOps Project(s) when referring to the named Azure service. Do not capitalize when referring to the concept formerly known as "team project" in VSTS. - When referring to a single user-created Azure DevOps Projects project, use "project" unqualified by the service name.

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS:	NOTES ON USAGE
Azure Boards	Work	Work management for teams with work item visualization such as Kanban boards.	*To manage your backlog, go to "Boards"...*	- Always use "Azure Boards"; do not shorten to "Boards", except when referring to the user interface. - If you encounter phrasing like "Work item tracking with VSTS", replace with "Work item tracking with Azure Boards" rather than "Work item tracking with Azure DevOps". The goal is to be Azure DevOps service specific now whenever possible based on the context of the doc.
Azure Repos	Code	Code Repositories for Git and TFVC.		- Always use "Azure Repos"; do not shorten to "Repos". - Awareness: If you encounter phrasing like "Get a VSTS Git repo", replace with "Get a Git repo with Azure Repos" rather than "Get an Azure DevOps Git Repo". The goal is to be Azure DevOps service specific now whenever possible based on the context of the doc.
Azure Test Plans	Test	Capabilities for manual testing of apps.		- Always use "Azure Test Plans"; do not shorten to "Test Plans". - The goal is to be Azure DevOps service-specific where possible.
Azure Artifacts	Package, Package management, Package extension	Package management and distribution.		- Always use "Azure Artifacts"; do not shorten to "Artifacts". - The goal is to be Azure DevOps service-specific now whenever possible based on the context of the doc.

NAME	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS:	NOTES ON USAGE
Team Foundation Version Control (TFVC)				- Always capitalize Version Control. Do not use "Team Foundation version control."

UI element terminology

The intent of this table is to capture the current state of terminology and plan for the next few Sprints as we transition to the new branding. This is to ensure consistency between design, product, docs, and marketing.

TERM	PREVIOUS TERM	DEFINITION AND NOTES ON USAGE
breadcrumb		Navigation in global header **Sample usage in docs:** *Global breadcrumb* or *breadcrumb in the global header*
flyout		A panel that appears in the right to take further inputs or provide additional information about the selected item. **Sample usage in docs:** *My Work flyout is available at the upper right of every page, providing quick access to the items you care about, no matter where you are in Azure DevOps services.*
global header		Header present on every page, shows location context and provides access to global controls. Includes homepage, breadcrumb, search, my work, marketplace, user profile
homepage		https://dev.azure.com - Use lower case "project" to refer to a project in Azure DevOps. Use Azure DevOps Project(s) to refer to the named Azure service. - Don't prepend Azure DevOps unless you really need to disambiguate. For example, use "project" rather than "Azure DevOps project" or "Azure DevOps Services project". - If you do need to disambiguate, use "Azure DevOps project".
lists		Secondary navigation used to search and filter within “pages”

TERM	PREVIOUS TERM	DEFINITION AND NOTES ON USAGE
organization	account, VSTS account, Team Services account	A shared space where users can collaborate on projects. See https://blogs.msdn.microsoft.com/devops/2018/07/19/adopting-the-word-organization/. **Sample usage in docs:** *This article is about adding a project to an Azure DevOps organization or a Team Foundation Service project collection. If instead you want to create a project in Azure DevOps Projects, see Azure DevOps Projects.* - Don't prepend Azure DevOps unless you really need to disambiguate. For example, use "organization" rather than "Azure DevOps organization" or "Azure DevOps Services project". - If you need to disambiguate, use "Azure DevOps organization".
pages	tabs, hubs, hub groups	Sections below services + anything else below
pivot	tabs	Navigation between two or more content panes and rely on text headers to label the different sections of content.
project	Team Project, VSTS project, Team Services project	A project provides a repository for source code and a place for a group of people to plan, track progress, and collaborate on building software solutions. **Sample usage in docs:** *This article is about adding a project to an Azure DevOps organization or a Team Foundation Service project collection. If instead you want to create a project in Azure DevOps Projects, see Azure DevOps Projects.* Since project is such a generic term, make sure that the context is clear. For example, avoid confusion with Visual Studio Project, etc. when needed
project page		`https://dev.azure.com/your-org/your-project`
selectors		Drop down menus provided to select from a favorite, filter a list of options, or browse all options. **Sample usage in docs:** *To quickly navigate to a feature or artifact—such as a dashboard, repository, product backlog, Kanban board, build pipeline—you can use breadcrumbs or selectors.*

TERM	PREVIOUS TERM	DEFINITION AND NOTES ON USAGE
services	Verticals, hub groups, products **Sample usage in docs:** *Go to the Services page and enable Azure Boards.*	Top-level services we sell
settings		There are several levels of settings: personal, team, project settings, organization (and possibly enterprise)
sidebar		Vertical pane that hosts the left navigation menu

Pipelines terminology changes

The following table captures some of the changes made in terminology for pipelines, builds, and releases. Terminology of unified pipelines is not addressed.

CONCEPT	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
build pipeline	build definition	A process describing how your app should be built and tested	(a) To create a pipeline, go to ... (b) When you are done, save the pipeline and queue a build. (c) Add the following snippet of code to your .azure-pipeline.yml file and push the changes to your Git repository.	The concept of build pipeline is more prominent when you use the web designer in order to author it. When using YAML files, you just author and push a YAML file (for example, .azure-pipeline.yml), and users do not think about "creating" a pipeline. Avoid repeatedly calling out *build pipeline* and just say *pipeline* when the context is clear. The verb that is used the most with this noun is *create* or *edit*.
build	build	The result of running a build pipeline once	(a) Queue a build ... (b) View the build logs ... (c) If the build fails, check if you made any changes to the pipeline (or to the YAML file) ...	This term can be used as a verb in a more general context, for example, build your app. When referring to gestures in Azure pipelines, it is typically used as a noun, for example, queue a build. The verbs that are used the most with this noun are *view* or *queue*.

CONCEPT	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
artifact	artifact	A build publishes zero or more artifacts.	(a) Add a *Publish artifacts* task to your build pipeline... (b) Your build published an artifact... (c) Download the artifacts in a release...	A build publishes artifacts, a release downloads and consumes them. Common verbs: *publish*, *download*.
job	phase or Job	A collection of tasks that run as one unit on an agent in a build or deployment	(a) Define a job in your pipeline ... (b) You can configure dependencies between jobs ... (c) Configure a matrix for your job ... (d) Each job in the build ...	Earlier, we used the term "Phase" as a definition-time concept, and "Job" as a run time concept. Now, we use the term job to mean both. Common verbs: *define*, *run*.
agent	agent	Software installed on a machine to run build or deployment jobs	(a) Use the Microsoft-hosted agent ... (b) Install an agent ...	Wherever ambiguous, we used to call this Build and Release agent. We can continue to do so.
self-hosted agents	private agents	Agents that you set up and manage to run build and deployment jobs	(a) You can use self-hosted agents to ...	You can drop 'self-hosted' if it is clear from the context. We are moving away from 'Private agent' since sentences such as 'Use a private agent in a public project' are awkward.
Microsoft-hosted agents	hosted agents	Agents that are set up and managed by Microsoft to run customers' build and deployment jobs	(a) The following software is installed on Hosted VS2017 agents ...	Microsoft runs four types of Microsoft-hosted agents - Hosted VS2017, Hosted, Hosted Linux, and Hosted Mac. Use Microsoft-hosted agents when referring to any of these agents. Otherwise, refer to the specific type. This change has been made to be consistent with the change to self-hosted agents.
parallel jobs	concurrent pipelines	A unit of licensing in pipelines	(a) Ensure that you have enough concurrent jobs ...	We changed the licensing recently from concurrent pipelines to concurrent jobs.

CONCEPT	PREVIOUS NAME	DEFINITION	SAMPLE USAGE IN DOCS	NOTES ON USAGE
release pipeline	release definition	A process describing how your app should be deployed to multiple stages	(a) To create a pipeline, go to ... (b) When you are done, save the pipeline, and create a release.	The verb that is used the most with this noun is *create* or *edit*.
release	release	The result of running a release pipeline once (or) A collection of artifacts that are is being deployed.	(a) Create a release ... (b) Deploy the release to a stage ...	This term is used as a noun. Common verbs: *create*, *deploy*
stage	environment	A logical break in the pipeline needed to separate concerns	(a) Add a stage to the release pipeline ... (b) Deploy the release to the QA stage	
deployment	deployment	The result of running a release targeting a single stage	(a) Deploy a release to a stage ... (b) Check the deployment logs (c) If the deployment fails, redeploy the release ...	We often mix up release and deployment when the context is clear. But, it would be better to keep them straight.
service connection	service endpoint	The credentials used to connect to a service outside of Azure DevOps services	(a) Configure an Azure service connection ... (b) Configure a GitHub service connection ...	

Docs URLs

While all links to VSTS docs should redirect, update URLs to the new Azure DevOps urls.

Project/Service URLs

OLD URL	NEW URL
{organization}.visualstudio.com/{project}	dev.azure.com/{organization}/{project}
{account}.visualstudio.com/{project}	dev.azure.com/{organization}/{project}

Other URLs

OLD URL	NEW URL	LINK TEXT	NOTES
https://twitter.com/vsts	https://twitter.com/AzureDevOps	@AzureDevOps	
https://azure.microsoft.com/case-studies/	https://azure.microsoft.com/case-studies/	Azure DevOps Case Studies	

OLD URL	NEW URL	LINK TEXT	NOTES
http://visualstudio.marketplace.com/vsts	http://visualstudio.marketplace.com/azuredevops	Azure DevOps extensions	Marketplace for both cloud and on-premises so ok to use Azure DevOps without qualifier
https://azure.microsoft.com/support/devops/	https://azure.microsoft.com/support/devops/	Azure DevOps support	en-us can be replaced with any locale
https://social.msdn.microsoft.com/Forumshome?forum=TFService	n/a	Visual Studio Team Services	No change!
https://visualstudio.uservoice.com/forums/330519-team-services	n/a	Visual Studio Team Services	No change!
https://stackoverflow.com/questions/tagged/vs-team-services	https://stackoverflow.com/questions/tagged/azure-devops	https://stackoverflow.com/questions/tagged/azure-devops	

Azure DevOps navigation guidelines

11/2/2020 • 2 minutes to read

With Azure DevOps, users will have a choice of using Horizontal (traditional) navigational model and the New navigation (with a vertical sidebar). To support users in navigating the UI, here are the recommended guidance elements to provide.

Provide explicit guidance for Quickstarts and Tutorials

For Quickstarts and Tutorials and other high page view articles, you should provide instructions on how to navigate to the starting page. These procedures should address both the new navigation experience and the previous navigation experience exemplified by the horizontal blue bar. For TFS 2013 through TFS 2018, there is ONLY the horizontal navigation experience. Azure DevOps Server 2019 (previously TFS) and later versions will only support the New Navigation experience.

For an example of usage, see the following Navigation basics (MD file).

Also, provide a link to Navigation basics for more information. For example: *To learn more, see Navigation basics.*

Provide a link to Navigation basics

For all topics that aren't Quickstarts or Tutorials, you can provide a link to the Navigation basics in place of explicit instructions.

Use INCLUDE files as much as possible

To cut down on redundancy and increase maintainability, use INCLUDE files as much as possible. In particular, use INCLUDE files when providing introductory content that is the same across several articles.

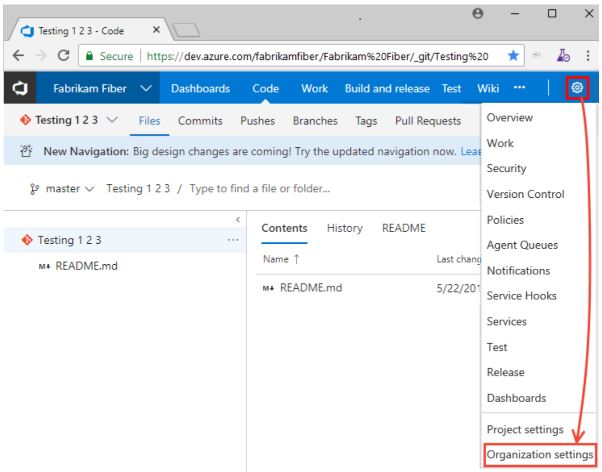
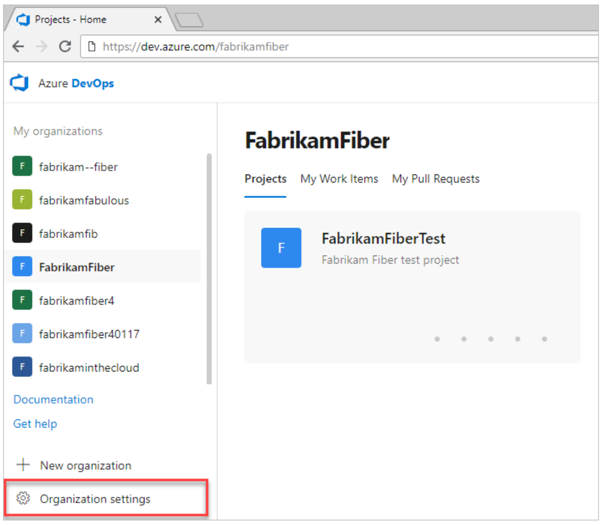
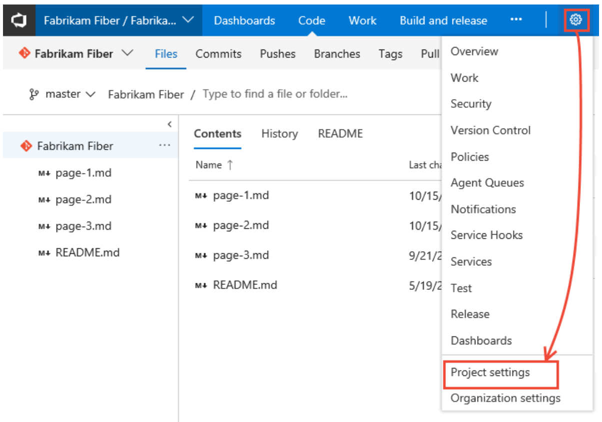
Use shared /includes image files as much as possible

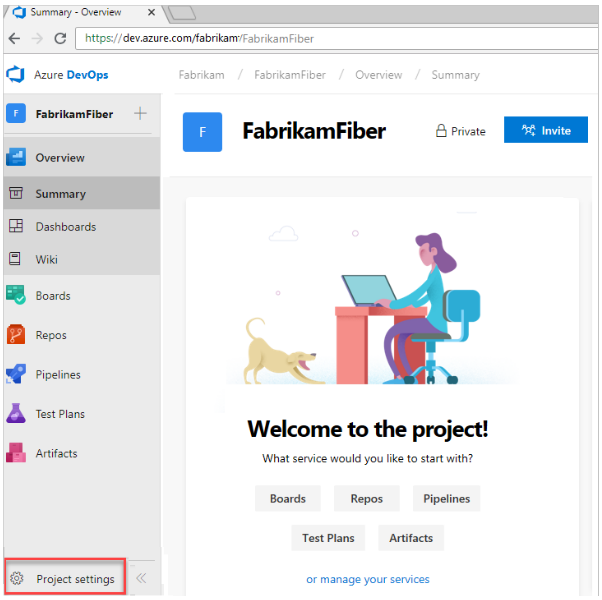
To reduce content redundancy, it's important to reuse images and content where it makes sense.

The **/docs/includes/media/settings** folder contains a number of common images used to open Organization and Project settings.

Examples are:

NAME	IMAGE

<table>
<tr><th>NAME</th><th>IMAGE</th></tr>
<tr><td>open-account-settings-horz-brn.png</td><td>
</td></tr>
<tr><td>open-admin-settings-vert.png</td><td>
</td></tr>
<tr><td>open-project-settings-horz.png</td><td>
</td></tr>
</table>

NAME	IMAGE
open-project-settings-vert-brn.png	

Publish Azure Databricks content

3/5/2021 • 6 minutes to read

Azure Databricks is a first-party Azure service that is primarily developed by Databricks Inc. The content for Azure Databricks is maintained by the Databricks Inc. documentation team in a private repository they own. Aside from a few cloud-specific differences, the documentation for Azure and AWS is the same and is reused on both documentation sites. The Databricks docs team updates content in a single source repo, and that repo is stripped of all AWS content before it is published on Microsoft Docs.

Workflow

1. The Databricks docs team makes updates in the source repo they own - `Databricks/docs`.
2. When the Databricks docs team is ready to publish, a CircleCI job strips the articles of all AWS content and puts the sanitized content in a new repo - `Databricks/azure-databricks-docs`.
3. Run the publishing script in the next section to move the new/updated Databricks content from `Databricks/azure-databricks-docs` to `MicrosoftDocs/databricks-pr`.
4. Open a pull request to the default branch using the output link from the publishing script.
5. Address any warnings by asking the Databricks team to update the content in the source repo and repeat the previous publishing steps. If there are no warnings, open a PR from the default branch to live.
6. Merge to publish.

Prerequisites

- Install PowerShell.
- Install Git.
- Access to the Databricks organization on Git Hub. Contact Leif Brown if you need access to publish.
- Clone the following repos to the same directory (save directory path for later – it's needed to run the script)
  - databricks/azure-databricks-docs
  - Your fork of MicrosoftDocs/databricks-pr
  - After your clone your fork, run the following command:

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

Script

Save the script as `Publish-Databricks.ps1`. Use the following command to run the script. Use the directory path that contains both prerequisite repos as the `-gitpath`.

```
Publish-Databricks.ps1 -gitpath "c:\docs"
```

- Windows
- Mac

```
#requires -version 5
<#
.SYNOPSIS
```

```
  Prepares Databricks content for publishing.
.DESCRIPTION
  Gets the latest changes from the azure-databricks-docs repo, moves them to the databricks-pr repo, cleans
up unwanted files, and commits the changes.
.NOTES
  Version:        1.0
  Author:         Mary McCready (mamccrea)
  Creation Date:  06/30/2020
  Purpose/Change: Initial script development
.EXAMPLE
  Publish-Databricks.ps1 -gitpath "c:\docs"
#>

#------------------------------------------------------------[Parameters]------------------------------------
----------------------
param(
    [Parameter(Mandatory=$true)][string] $gitpath
)

#----------------------------------------------------------[Initializations]---------------------------------
----------------------

#Set Error Action to Silently Continue
$ErrorActionPreference = "Stop"

#-----------------------------------------------------------[Declarations]-----------------------------------
----------------------

# Script Path/Directories
$azuredbksdocs = $gitpath + '\azure-databricks-docs'
$databrickspr = $gitpath + '\databricks-pr'
$azuredbksdocsarticles = $azuredbksdocs + '\docs'
$azuredbksdocskb = $azuredbksdocs + '\kb'
$azuredbksdocsscenarios = $azuredbksdocs + '\scenarios'
$azuredbksdocstoc = $azuredbksdocs + '\databricks-pr\TOC.yml'
$databricksprarticles = $databrickspr + '\databricks'
# Date/Time format
$date = Get-Date -UFormat "%m-%d-%Y"
$time = Get-Date -Format "HHmmss"
# Git variables
$commitmessage = 'merge from source ' + $date
$branchname = 'merge-' + $date + $time

#------------------------------------------------------------[Execution]-------------------------------------
----------------------

# Get latest changes to azure-databricks-doc repo (owned by Databricks)
Write-Host "Getting latest changes to azure-databricks-doc repo"
Set-Location $azuredbksdocs
git checkout master
git pull
Write-Host "Finished getting changes to azure-databricks-docs"


# Sync databricks-pr repo (owned by Microsoft) with master and create new branch
Write-Host "Syncing databricks-pr with master and creating new branch"
Set-Location $databrickspr
git checkout master
git pull upstream master
git checkout -b $branchname
Write-Host "Finished Syncing databricks-pr with master and creating new branch"

Write-Host "Removing old content from databricks-pr"
# Get list of all items in the databricks-pr\databricks directory
[System.Collections.ArrayList]$oldtberemoved = Get-ChildItem $databricksprarticles | ForEach-Object {
$_.FullName }

# Remove items in the excludes list from the items to be removed
[System.Collections.ArrayList]$excludes = ($databricksprarticles + '\docfx.json'), `
```

```
    ($databricksprarticles + '\breadcrumb'),`
    ($databricksprarticles + '\index.yml')

Write-Host $excludes

foreach($e in $excludes){
    $oldtberemoved.Remove($e);
}

# Remove all items in the to be removed list
foreach($item in $oldtberemoved){
    Remove-Item $item -Force -Recurse
}

Write-Host "Finished removing old content from databricks-pr"

Write-Host "Copying new content to databricks-pr"
# Copy the docs directory to databricks-pr
Get-ChildItem -Path $azuredbksdocsarticles | Copy-Item -destination $databricksprarticles -Recurse -
Container

# Create a new kb directory in databricks-pr and copy the kb directory contents to databricks-pr
$newkb = New-Item -Path $databricksprarticles -Name "kb" -ItemType "directory"
Get-ChildItem -Path $azuredbksdocskb | Copy-Item -Destination $newkb -Recurse -Container

# Create a new scenarios directory in databricks-pr and copy the kb directory contents to databricks-pr
$newscenarios = New-Item -Path $databricksprarticles -Name "scenarios" -ItemType "directory"
Get-ChildItem -Path $azuredbksdocsscenarios | Copy-Item -Destination $newscenarios -Recurse -Container

# Copy the TOC.yml to databricks-pr
Get-ChildItem -Path $azuredbksdocstoc | Copy-Item -Destination $databricksprarticles -Recurse -Container

Write-Host "Finished copying new content from databricks-pr"

# Delete all AWS/S3 references, the sitemap, the index.md, and the /spark/latest/faq directory
Write-Host "Deleting unwanted content"
Get-ChildItem -Path $databricksprarticles -Include *aws*.* -Recurse | ForEach-Object { $_.Delete()}
Get-ChildItem -Path $databricksprarticles -Include *s3*.* -Recurse | ForEach-Object { $_.Delete()}
Get-ChildItem -Path $databricksprarticles -Include *sitemap*.* -Recurse | ForEach-Object { $_.Delete()}
Remove-Item -LiteralPath $databricksprarticles\index.md
Remove-Item -LiteralPath $databricksprarticles\spark\latest\faq -Force -Recurse
Write-Host "Finished deleting unwanted content"

# Commit changes and delete local branch
Write-Host "Committing changes"
git add .
git commit -m $commitmessage
git push origin $branchname
git checkout master
git branch -D $branchname
Write-Host $branchname " committed. Go to GitHub to open a PR: https://github.com/MicrosoftDocs/databricks-
pr"
```

Points of contact

- Leif Brown leif.brown@databricks.com - Databricks Director of Documentation
- Stephanie Bodoff stephanie.bodoff@databricks.com - Databricks Staff Technical Writer
- Santosh Perla Santosh.Perla@microsoft.com - Azure Databricks Program Manager

Update Azure Databricks content

3/5/2021 • 2 minutes to read

Azure Databricks content is updated by a one-way pipeline from a Databricks-owned source repo to Microsoft Docs. Because of this, you can't make updates directly in the MicrosoftDocs/databricks-pr repo. To make updates, reach out to one of the following points of contact:

- Leif Brown leif.brown@databricks.com - Databricks Director of Documentation
- Stephanie Bodoff stephanie.bodoff@databricks.com - Databricks Staff Technical Writer
- Santosh Perla Santosh.Perla@microsoft.com - Azure Databricks Program Manager

Metadata

Metadata is automatically generated/updated by the CircleCI job. Because of this, custom metadata is not supported and all topic types are `conceptual`.

Exceptions

The following folders in the databricks/azure-databricks-docs repo do not exist in the source repo and must be updated outside of the CircleCI process.

- databricks-pr: contains the TOC.
- scenarios: contains all top-of-funnel content (overviews, quickstarts, tutorials, and workspace and SQL Analytics landing pages).

The following files in the MicrosoftDocs/databricks-pr repo do not exist in the source repo or the databricks/azure-databricks-docs repo and must be updated directly in MicrosoftDocs/databricks-pr.

- databricks/docfx.json
- databricks/index.yml: overall Databricks service landing page

Repository and publishing information for the Partner Center business

5/26/2021 • 4 minutes to read

There are several separate GitHub repositories for the Partner Center docs. The following table describes the purpose of each repo, and shows the public/private pairing if any. The public docs URL is also shown where applicable.

REPO NAME	PURPOSE	PRIVATE REPO	PUBLIC REPO	DOCS URL
Partner Center	Conceptual docs for Partner Center and customer docs for the commercial Marketplace.	partner-center-pr	None	Docs URL
Azure Docs	Conceptual docs for Azure commercial Marketplace and Billing.	azure-docs-pr	azure-docs	Docs URL
Office store Docs	Conceptual docs for commercial Marketplace offering in the Office store.	NA	office-store-docs	Docs URL
Partner Center SDK	Conceptual docs and hand written reference docs for Partner Center. Overlaps with some of the autogenerated reference docs as well.	partner-center-sdk-pr	partner-center-sdk	Docs URL
Partner Center Rest	Partner Center Rest API Reference	TBD	TDB	Docs URL
Partner Rest	Partner REST API (different from Partner-center)	partner-rest-pr	partner-rest	Docs URL
Partner Center Swagger Specs	Swagger specs to publish automated REST documentation. There is a partner-center-swagger-admin group to join to get access to publish the swagger specs.	partner-center-swagger	NA	NA

REPO NAME	PURPOSE	PRIVATE REPO	PUBLIC REPO	DOCS URL
Partner .NET SDK	Partner Center .NET SDK Reference	Windows-partner-center-dotnet	None	Docs URL
Partner Java SDK	Partner Center Java SDK Reference	None	partner-center-java-docs	Docs URL
Partner PowerShell SDK	PowerShell reference docs for Partner Center	None	partner-center-docs-powershell	Docs URL
Partner Center China Redirect	Hosts several redirects from URLS to old MSDN URL that points to 21Vianet, such as this url	partner-china-redirect	None	redirects only
Partner Center Red Tiger	Hosts .pdf and .docx files for downloads. Several agreements are hosted here that are not widely available on docs. The URL is a pattern such as `https://query.prod.cms.rt.microsoft.com/*` . Though one contributor can post and approve an asset, a second person has to publish it.	RedTiger	None	Varies - Example

Pull Request Reviews

1. **Azure-docs-pr** pull request reviews are handled by the Aquent managed team. Type `#sign-off` in the PR, and the PRMerger bot will add the `ready-to-merge` label, which the reviewers monitor for.

2. **Partner-center-pr**, **Partner-center-sdk**, and **Partner-center-sdk-pr** pull requests are also handled by the Aquent managed team.

   You can request a technical review before signing off. In the PR comments, type @partnerdocs or mentioning an individual, and assign a review at the upper right of the pull request.

   Type `#sign-off` in the PR, and the PRMerger bot will add the `ready-to-merge` label, which the reviewers monitor for. They will review within an hour usually, but worst case 1 day.

   If the reviewers give you blocking feedback that prevents them from merging, they will type `#hold-off` and it resets the `ready-to-merge` label. You should adjust the source markdown, and `#sign-off` again to get their attention to request the merge.

3. The remaining repos listed above are not monitored for `#sign-off` regularly. Follow the instructions in the Pull Request opening template, and @ mention the repo admins if in doubt in the comments to get their attention.

Publishing schedules

The busiest repos are published live on a schedule. The other are on-demand.

1. **Azure-docs-pr** publishing is handled by CPS. Generally happens three times a day. Schedule here
2. **Partner-center-pr** and **Partner-center-sdk-pr** docs publish twice a day.

   There is a set of custom Azure DevOps Pipeline jobs that run on a schedule. The scheduled times are 9am and 3pm PST. The job pulls from main branch and merge into live branch. Occasionally the publication fails if there is a merge conflict or a build error.

   Repo admins can also merge from main branch to live branch manually if needed.

3. The other docs are all published on demand only by the repo admin (merge main branch to live branch).
4. Public repos that are paired to a private repo use a **repo_sync_working_branch** branch provided by OPS to keep the two repos in sync. The repo-sync branch is used to forward the public commits from main branch in the public repo in a pull request into the main branch of the private repo. A pull request shows up in the private repo when the sync mechanism has forwarded public changes. However, that PR does not automatically merge, so an administrator needs to be monitor and sign-off or merge it.

Repo security

The members of the group @MicrosoftDocs/partnerdocs help with the content in the Partner Center Docs.

The group @MicrosoftDocs/partner-center-docs-admins can help with repo administration, such as setting up branches, resolving build issues, and publishing out of band. Content production Service (CPS) helps with Azure-docs-pr, but does not assist with branches and administration for the other partner repos.

Use the Docs Support general channel in Microsoft Teams if there is a problem with the repo that requires advanced troubleshooting.

Autogenerated REST docs

1. Swagger specs (.json files) go into this repo:

   https://github.com/MicrosoftDocs/partner-center-sdk-pr/blob/master/rest-swagger-specs/

   Anyone in MicrosoftDocs org can submit a PR, but only repo admins or PR Reviewers can merge the PR.

   Developers who want to publish new swagger specs for Partner Center can discuss with the folks in this team: @MicrosoftDocs/partner-center-docs-admins

2. Run a CI/CD pipeline to pull the Swagger specs in and build the autogenerated YAML page output.

   Join the security group on ID Web: pcswaggerdoc

3. Publication

   Rest API Docs are staged here: https://review.docs.microsoft.com/rest/api/partner-center/audit?branch=main

   Once public, the docs render here: https://docs.microsoft.com/rest/api/partner-center/audit.

Use applies to and appropriate roles in Partner Center

5/27/2021 • 2 minutes to read

Specific attributes in the meta-data

Assign values for both ms.service and ms.subservice to your article. Generally, the values for the articles in Partner Center include the following (there are a few exceptions for topics related to Marketplace, etc.):

ms.service:

- partner-dashboard

ms.subservice:

- partnercenter-mpn
- partnercenter-csp
- partnercenter-incentives
- partnercenter-sdk (for developer docs only)

Requirements for every article

An article may contain an **Applies to** section. However, for articles that just apply to the general Partner Center, you do not need **Applies to**. Use **Applies to** only if you are calling out cloud instances that are different from the standard Partner Center. In this case, also include Partner Center as a value (for example: **Applies to**: Partner Center | Partner Center for Microsoft Cloud Germany). Otherwise, you don't need to tell readers that the article applies to Partner Center. If **Applies to** is needed it should be at the top, directly following the H1.

Next, with few exceptions, each Partner Center article needs an **Appropriate roles** section at the top, either directly following **Applies to**, if needed, or the H1.

Values for both **Appropriate roles** and **Applies to** should be on a single line and separated by a pipe (|).

Special instructions for the SDK

In the SDK, we currently do not use **Appropriate roles**. We do use the **Applies to** section, although the values used may be different than the values described in this guideline article. You should format the values for **Applies to** in the same way as for the Partner Center documentation (values on a single line and separated by a pipe).

Possible values for Applies to

Applies to refers to websites, not Partner programs such as CSP. Accepted terms used here include (but are not limited to):

- Partner Center for Federal partners
- PartnerCenterforMicrosoftCloud for US Government
- PartnerCenterforMicrosoftCloudGermany
- Partner Center (when used with another value)

(Remember, for articles that just apply to the standard Partner Center, there is no need to include an **Applies to** section.)

Possible values for Appropriate roles

Appropriate roles refers to the Partner Center user role that can accomplish the task being explained in the article or that can see the information in the dashboard. The author of the content should be aware of which type of user will be able to make use of the content. We don't want partners to waste their time trying to do tasks they don't have the needed credentials to do.

Roles can overlap--an employee may have more than one role.

There are a few articles, such as the overview article **What is Partner Center?**, that are appropriate for all partners. The correct terminology for this is:

- All partners interested in Partner Center *or*
- All partners interested in learning more about

Otherwise, valid roles include (but are not limited to) the following values:

Manages commercial transactions and Azure AD:

- Global admin
- Account admin
- User management admin

CSP specific:

- Billing admin
- Admin agent
- Sales agent
- Helpdesk agent
- Referrals admin
- Incentives admin
- Incentives user
- Executive report viewer
- Report viewer

MPN specific:

- MPN partner admin
- Account admin

PSC specific:

- Partner Sales Connect (PSC) seller
- Partner Sales Connect (PSC) admin
- Partner Sales Connect (PSC) deal manager

Additional roles:

- Guest user
- CPV admin
- Advisor
- System admin or system customizer on the CRM
- Referrals user

General roles

- All Partner Center users
- Direct bill partners
- Indirect resellers
- CSP partners

Examples of correct usage

Applies to: Partner Center | PartnerCenterforMicrosoft Cloud for US Government

Appropriate roles: Global admin | Admin agent

Write a Partner Center announcement

6/2/2021 • 8 minutes to read

The purpose of an announcement is to inform Microsoft partners about offer promotions, upcoming new features, changes to policy, or other changes that might affect their businesses.

Treat an announcement like a news story. That is, get right to the point. Briefly cover only what partners need to know, and avoid providing the details you might ordinarily handle in a longer article, such as a Help topic. Announcements are not where partners learn how to complete a task. Ideally, your announcement would link to a Help topic or to the Operations gallery for more in-depth information.

Required sections

- **Categories**: Include the date of the announcement.
- **Summary**: Provide high-level information, including the date when the feature or the change becomes effective.
- **Impacted audience**: List who is affected by this information.
- **Date published**: Specify when the announcement goes live.
- **Details**: Describe the announcement in detail.
- **Next steps**: Provide resources to help partner prepare for the change, release, or event.
- **Questions**: (Optional) Link to Partner Center support, Yammer communities, or other resources so that partners can ask questions about the announcement.
- **Change log**: (Optional) List dates of changes or updates to the announcement.

Heading formatting

- **Heading length**: For the main heading, or announcement title, use no more than 60-75 characters, including spaces. The heading is displayed as a link on the main Partner Center announcements page and as an internal TOC entry on the announcements page for a specified month.

- **Heading size**: The announcement title is an H2 heading (prefixed with ## and a space); the section headings are H3s (prefixed with ### and a space).

Heading examples

- An effective heading:

  Audit log changes to include two new operation types for DAP (Why: this heading is short and to the point.)

- A less effective heading:

  Phase 1 Now Live: EU/EFTA & UK Change of Partner Billing Currency for new commerce offers (Why: this heading contains unnecessary information, and the timing ["Phase 1"] and audience [“new commerce”] are unclear.)

  - Here’s an improved version of the same announcement:

    Billing currency will change for some EU/EFTA offers (Relevant details are placed in the body of the announcement.)

Announcement examples

New offer promotion or offer change

(H2) Dynamics 365 offers for February 2021

(H3) Categories

- Date: 2021-01-04
- Offers

(H3) Summary

New Dynamics 365 offer changes will be launched in February 2021.

(H3) Impacted audience

Partners participating in the Cloud Solution Provider (CSP) direct bill partner, indirect provider, and indirect reseller incentive programs

(H3) Details

Microsoft is announcing upcoming new Dynamics 365 offer changes that will be launched in February 2021 for:

- Dynamics 365 Customer Voice Additional Response

...

(H3) Next steps

Review the resources about this topic and share them with the appropriate stakeholders in your organization.

(H3) Questions?

For any further questions about these offers, check your relevant Yammer communities.

Business process change

(H2) New updates to the Referrals module in Partner Center

(H3) Categories

- Date: February 5, 2021
- Capabilities

(H3) Impacted audience

- Partners with Co-sell Ready offers
- Partners with a transactable offer using an Azure incentive-eligible solution in the

...

API change

(H2) Reminder: Introducing API throttling to partners calling Partner Center APIs

(H3) Categories

- Date: 2021-02-16
- Grow your Business

(H3) Summary

Microsoft will be implementing API throttling to allow a more consistent performance for partners calling the Partner Center APIs.

(H3) **Impacted audience**

All partnerstransactingthrough the Cloud Solution Provider (CSP) program

(H3) **Details**

Beginning in February 2021, Microsoft will be implementing API throttling to allow a more consistent performance for partners calling the Partner Center APIs.

...

(H3) **Partnerbenefits**

This update will reduce the complexity and overhead with multi-currency invoicing in the EU/EFTA & UK for the new commerce experience.

- Partners will receive a consolidated invoice in a single currency and will no longer receive an invoice for each customer location currency
- Incentive payouts will be in the same currency as the partner's invoice currency

...

Deprecation notice

(H2) **Deprecation of PUT and GET Qualification APIs for Education customer validation**

(H3) **Categories**

- Date: February 4, 2021
- Capabilities

(H3) **Impacted audience**

Partners selling Academic, Non-Profit, and GCC offers through the Cloud Solution Provider program using the Partner Center API

...

Pricing change

(H2) **Microsoft 365 E3 price changes coming for March 2021**

(H3) **Categories**

- Date: February 11, 2021
- Offers/Markets

(H3) **Impacted audience**

All partners transacting through the Cloud Solution Provider (CSP) program in EUR, AUD, GBP, and JPY.

Walkthrough: Creating a Partner Center announcement from soup to nuts

Step One: Receive the announcement

1. The process begins when you get an email (usually from Katie Kent) with the announcement attached as a Word document. The email Subject line should also contain the announcement number, title, contact, and publish date. They usually want it published in 48 hours, which is the lead time we've asked them for.
2. Within the hour if possible, create a deliverable for the announcement here, using the following:

FIELD	TEXT
Title	ANNOUNCEMENT 9315: Product launch calendar – March 25
Assigned to	(yourself)
State	Started
Iteration	(current)
Description	Publish attached announcement by Thursday, 3/25
Attachment	(original attachment that came in email)

3. Do a “Reply all” to the announcement email that looks something like this:

   Hi, Katie—

   We’ve created deliverable 32223758 to track announcement 9315: Product launch calendar, and it should go live Thursday as requested. I’ll keep you posted.

   Pete

   If you can't create a deliverable within the hour, send somethink like this and follow up later:

   Thanks, Katie—

   We should able to publish this Thursday as requested. I’ll keep you posted.

   Pete

   ...and then create the deliverable when you can.

Step Two: Clean up and structure the announcement

Here’s where we get the content ready for markdown. I do this in Word – you do what you like.

4. Open the announcement email, and save the attached Word document to a convenient working folder. Include the four-digit announcement number in the filename if it’s not already there, such as **9315 Product launch calendar.docx**.
5. Open the document, select and copy all content, paste into a new document as unformatted text.
6. Save that new document – I usually call mine something like **PA 9315.docx**.
7. Working in that new document now, remove extraneous text at the end, such as **"Let’s do great work together”** and any content after that.
8. Remove any extra spaces, line breaks, etc.
9. Structure the content into the standard announcement sections below (this is usually done already):

(H2)
[ANNOUNCEMENT TITLE]

(H3) **Categories**

[date]

[content]

(H3) **Impacted audience**

[content]

(H3) **Details**

[content]

(H3) **Next steps**

[content, bulleted list]

(H3) **Questions?**

[content]

Step Three: Edit the announcement

10. Once you've got it cleaned up and structured, there are generally four things to edit. Refer back to that original Word document as needed.

- Content
  - At the very least, make it Acrolinx-ready
  - Tone down anything that sounds too much like marketing
  - Watch for acronyms that aren't spelled out the first time – this is common
  - Improve the content as needed and time permitting
- Lists (bulleted and numbered)
- Tables
- Links – Go to Word document, right-click link => Edit Link => ctrl+c, go to announcement document, put [] around linked text followed by () with link inside. Make sure /en-us/ is edited out

Now you've got a Word document that's markdown-ready.

Step Four: Add the announcement to the announcements file

This is just standard Gitbash, Github, VS Code, however you like to do it. I use Gitbash and VS Code.

11. Create a branch, such as `pete-announce-9315`.

12. Open VS Code.

13. Open this month's file: `2021-april.md`.

14. Copy your content from Word and paste into the file, right after this text:

    `This page provides the announcements for Microsoft Partner Center for April 2021.`

15. Do whatever cleanup VS Code requires – lists will probably need spaces redone, etc.

16. Look at what had been the latest announcement, see what number is in its title, and make yours the next number. For example, if `15` used to be at the top, you'll add yours at the top as number `16`.

17. Preview, fix, clean up, etc. and save.

18. Do the two things I always forget:

- Update the date on the article
- Search for "en-us" and remove if found

19. You're done editing. Save your file but keep VS Code open – next you'll do the index.

Step Five: Update the index

The index is essentially a list of all announcements, with the most recent one at the top. That's where yours is going to go.

20. Open the file called **index.md**.

21. You'll see a table with a lot of rows that look like this:

    | Grow Your Business | [ANNOUNCEMENT TITLE] (2021-april.md#7) | April 19 |

22. Create a new row for your announcement. I usually just copy an existing row, paste it at the top, then update it with the right **category**, **title**, **filename** and **#**, and the **date** you want it to go live.

23. Preview, fix, and save. When you're done, close VS Code.

24. The usual: add-commit-push, open PR, watch for build results, fix as necessary, sign off.

Step Six: Confirm publication and update the team

25. When the time and day of publication arrive, go to the **Announcements** page and confirm that yours is there.

26. When it's live and looking good, do another "Reply all" to that original email thread and say something like this:

    Hi, Katie—

    Announcement 9315: Product launch calendar is live as of this morning. Let me know if you have any questions. Thanks!

    Pete

27. Open your deliverable, mark it as **COMPLETED**, then save and close.

28. You're done!

Finally: A note on organization

It's quite common to be editing and publishing numerous announcements at once. It can be very useful to create a table for yourself to track the following for each announcement:

- Announcement number
- Deliverable number
- Owner or PM
- Subject
- Pull request number
- Requested publication date
- Status

...

How to contribute to Power BI documentation

5/20/2021 • 2 minutes to read

This article provides specific guidance for Microsoft employees to contribute to Power BI documentation.

GitHub repository

Use MicrosoftDocs/powerbi-docs-pr for internal contributions:

- Internal pull requests
- To get permissions to contribute to the repository, see Set up a GitHub account.

Use MicrosoftDocs/powerbi-docs for public contributions:

- Public pull requests

Email contact

Use the PbiSqlBiContent email alias to discuss technical content and pull requests related to the powerbi-docs-pr and powerbi-docs repositories.

Customer feedback channel

The Power BI docs use GitHub issues for customer feedback.

Visual Studio work item tracking

Use this new work template to track work items or bugs for this repository:

- Area Path: `TechnicalContent\Data and Analytics\BI\Power BI`
- Iteration Path: `TechnicalContent\3WeekSprints\Future`
- State: `Proposed`
- Tags: `Power BI` and other variations such as `Power BI Desktop`, `Power BI Embedded`, `Power BI Mobile`, and `Power BI Pro`.
- Description: Explain the issue or requirement and provide a link when possible. If there's a due date, explain why.
- Story Points: The number of days estimated to do the work. For example, 1.0, 0.5, or 7 (days).

Example article metadata

The following metadata block is an example for a Microsoft Power BI article. For detailed metadata information, see Metadata overview. Recommendations for Power BI docs:

- Mention the product name, Power BI, in the **title** and **description** tags for SEO.
- Specify the **services**, **ms.service**, and **ms.workload** tags by using technology-specific values such as `powerbi` or `power-bi-embedded`.
- Specify a **ms.subservice** value, such as `powerbi-desktop`, `powerbi-service`, `powerbi-admin`, `powerbi-gateways`, `powerbi-mobile`, `powerbi-developer`, `powerbi-report-server`, `powerbi-template-apps`, or `powerbi-custom-visuals`.
- Refresh the **ms.date** tag each time you make a significant change to the article for freshness.

```
---
title: Overview of Power BI Desktop
description: Power BI Desktop lets you build advanced queries, models, and reports that visualize data.
ms.service: powerbi
ms.workload: powerbi
ms.subservice: powerbi-desktop
author: githandle
ms.author: msalias
ms.topic: conceptual
ms.date: 04/23/2021
---
```

Free sign-up recommendation

Use this sentence in each quickstart that you write to encourage users to try a free trial:

```
If you're not signed up for Power BI, [sign up for free](https://app.powerbi.com/signupredirect?
pbi_source=web) before you begin.
```

Open issues

- Docs team info: OneNote

Get started contributing to SQL documentation

6/2/2021 • 2 minutes to read

This article provides guidance on how Microsoft employees can contribute to SQL documentation published at https://docs.microsoft.com/sql/ and https://docs.microsoft.com/azure/. SQL Server documentation is managed in two main GitHub repositories:

- sql-docs-pr : SQL Server on-premises content.
- azure-docs-pr: SQL content on Azure.

General tasks

If you need to make a minor change to an existing article, see Quickly edit an existing article. For all other edits, use the following resources in the Contributor's Guide:

TASK	CONTRIBUTOR'S GUIDE RESOURCES
Setup a local repository for larger or frequent edits	Sign up for a GitHub account Install Git and VS Code Set up a Git repository locally
Submit larger or frequent edits	Create working branches and submit pull requests Edit content in Visual Studio Code
Follow content standards	Write an overview Write a quickstart Write a tutorial Write a sample Edit a TOC
Improve your writing	Follow writing principles Use Acrolinx Review the Microsoft style guide

TIP

If you are adding new content or making significant changes, please include a member of the content team on this work to avoid duplication and to align with ongoing content strategies.

SQL tasks

There are also writing tasks and resources specific to SQL documentation related to the tasks in the previous section.

TASK	SQL DOCS RESOURCES
Ask a question	Email the SQL content team
File a new content request	Open a new Azure Boards user story

TASK	SQL DOCS RESOURCES
View current work	View the SQL content dashboard
See staged content	Visit the review site and select a branch
SQL writing tasks	Edit metadata in SQL articles Use applies-to includes Manage versioning Work with SQL release branches

Next steps

For more information, see the Contributor's Guide.

SQL documentation guidance

6/1/2021 • 7 minutes to read

This article provides guidance for Microsoft employees who contribute to SQL documentation in **sql-docs-pr** and **azure-docs-pr**.

How to contribute

You can contribute to directly to SQL docs either by editing in your browser or by working locally with VS Code and GitBash. For more information, see Get started contributing to SQL Documentation.

> **IMPORTANT**
>
> If you need to make changes for a future release, please use an appropriate release branch and follow the steps on working with release branches in this article.

Author publishing process

Authors refer to doc owners, such as content writers, or program managers (PMs) from the product group. Authors should follow the publishing process outlined in this section. Writers that do not fit into these two categories should follow the contributor guidance.

To submit your documentation change, open a GitHub pull request (PR). For more information, see How to create a pull request. Please note the following:

- After you complete your changes, you must add #sign-off in the comments of the PR.
- The PR is reviewed for standards before merging. You will be asked to fix any issues.
- Any SQL PR that does not show activity for 30 days will be closed. You can reopen it at any time.

After your changes are merged, they are ready to publish. Twice a day, the **master** branches of **sql-docs-pr** and **azure-docs-pr** are pushed to the public repositories. This publishes the content live. Publishing happens at roughly 10:30am and 3:30pm PST.

> **NOTE**
>
> If you are working in a release branch, then the content is not published until that release branch is merged into **master**.

Email contact

Use the sqldocsPR email alias to discuss technical content and pull requests related to **sql-docs-pr** and **sql-docs**.

Visual Studio workitem tracking

To see what is happening in SQL Server content, access the SQL Azure Boards Dashboard.

If you have a new content idea or need to partner with the content team in any way, file a new Azure Boards work item.

SQL docs guidance

The following section provides some additional guidance on working in the **sql-docs** and **sql-docs-pr** repositories.

> **IMPORTANT**
>
> The information in this section is specific to **sql-docs**. If you are editing a SQL article in the Azure documentation, see the Readme for the azure-docs repository on GitHub and the Azure section of the internal contributor's guide.

The sql-docs-pr repository uses several standard folders to organize the content.

FOLDER	DESCRIPTION
docs/	Contains all published SQL Server content. Subfolders logically organize different areas of the content.
docs/includes/	Contains include files. These files are blocks of content that can be included in one or more other topics.
./media/	Each folder can have one **media** subfolder for article images. The **media** folder in turn has subfolders with the same name as the topics that the image appears in. Images should be .png files with all lower-case letters and no spaces.
toc.yml	A table-of-contents file, in the newer YAML format. Each subfolder has the option of using one toc.yml file.

Contributor guidance

Product group program managers or content developers approve all updates in the private repository. Contributors who are not in the product group may add PMs or content developers as reviewers, but they may not sign off on new content or updates. This policy ensures that content aligns with product group intent and supportability.

Procedure

Once a Pull Request (PR) has been submitted, you must:

1. Check the Acrolinx scorecard and ensure to implement the suggested changes.
2. Review the staged content to ensure formatting is okay and your changes look good. Do this by selecting the "View" link in the **Validation status: Passed** comment under **Preview URL** within your Pull Request.
3. Check the metadata for who the ms.author of the article is, and email them directly with the link your PR. CC the content team (rogue1@microsoft.com). The author or someone on the content team will sign off.

Guidelines

These are general guidelines that all authors must follow, but that are not explicitly stated elsewhere.

- When providing a link to content, content on docs take priority. Do not provide a link to a blog if a docs article exists. If providing a link to a blog, the blog should be vetted by the article author for technical accuracy. Ideally, the content of the blog should be migrated to docs, but the author needs to make that call.

Article metadata

Each article has a metadata block at the top. For detailed metadata information, see Metadata guidance. In the near future, we will post more specific metadata guidance for SQL topics here. For now, use metadata in a similar way as existing SQL articles in your area.

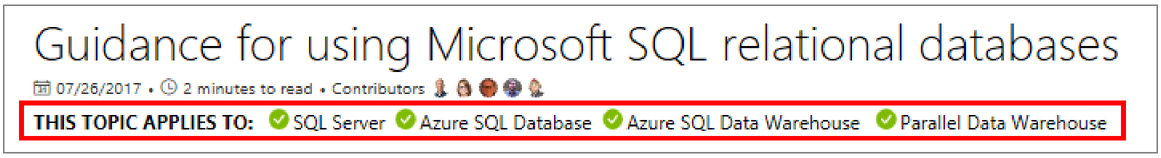
Applies-to includes

Each SQL Server article contains an **applies-to** include file after the title. This indicates what areas or versions of SQL Server the article applies to.

Consider the following Markdown example that pulls in the **appliesto-ss-asdb-asdw-pdw-md.md** include file.

```
[!INCLUDE[appliesto-ss-asdb-asdw-pdw-md](../includes/appliesto-ss-asdb-asdw-pdw-md.md)]
```

This adds the following text at the top of the article:



To find the correct applies-to include for your article, use the following tips:

- For a list of commonly used includes, see SQL Server version and applies-to include files.
- Look at other articles that cover the same feature or a related task. If you edit that article, you can copy the Markdown for the applies-to include link (you can cancel the edit without submitting it).
- Search the docs/includes directory for files containing the text "applies-to". You can use the **Find** button in github to quickly filter. Click on the file to see how it is rendered.
- Pay attention to the naming convention. If there are x's in the name, they are usually placeholders indicating the lack of support for a service. For example, **appliesto-xx-xxxx-asdw-xxx-md.md** indicates support for only Azure SQL Data Warehouse, because only **asdw** is spelled out, whereas the other fields have x's.
- Some includes specify a version number, such as **tsql-appliesto-ss2017-xxxx-xxxx-xxx-md.md**. Only use these includes when you know the feature was introduced with a specific version of SQL Server.

There is some overlap between applies-to includes and versioning, but it is not complete. For now, there is a need to use both applies-to includes and versioning to help customers navigate the SQL documentation.

Versioning

In addition to applies-to includes, SQL Server content has implemented versioning. Users can select a SQL Server service or product version from the left-hand TOC to scope the content that is displayed. To understand how to use versioning in SQL topics, see SQL versioning guidance and scenarios for writers.

Branches

SQL Server content uses the following branches for releasing content. The instructions in this guide use the **master** branch, which is for immediate release. But there are also release branches for delayed releases.

BRANCH	DESCRIPTION
master	Changes publish twice a day from this branch.
release-sql2017	Changes for the next cumulative update of SQL Server 2017.

BRANCH	DESCRIPTION
release-sqlseattle	Changes for the next CTP of SQL Server vNext.

Work in a release branch

Use the following steps to work in a release branch like **release-sql2017** or **release-sqlseattle**.

1. If you have not locally cloned the **sql-docs-pr** repository and installed gitbash and VS Code, see Setup a local repository for larger or frequent edits.

2. Follow the steps in Check out a release branch to check out a working branch for your release branch. For example, if you want to make changes to SQL Server vNext, you could run the following command to create a working branch **mychanges** for **release-sqlseattle**...

```
cd /sql-docs-pr/docs
git checkout -B mychanges upstream/release-sqlseattle
git pull upstream release-sqlseattle
```

3. Make all of your changes and then Commit and push your working branch.

4. Then create a pull request for your changes. It helps to add `@githubusername` to the comments for any content lead or PM associated with the change to facilitate awareness and collaboration.

   Caution

   You must insure that your pull request pulls your changes into the corresponding release branch, such as **release-sqlseattle**, and not **master**. If you accidentally pull your changes into **master**, all of the pre-release content will go live (not just your PR). Having a lot of extra commits that are not associated with your change is a clue that you have made this mistake.

If you have any questions or need any help on how to work with release branches, email sqldocsPR.

SQL Syntax colorizer

Most SQL Server articles about Transact-SQL statements contain two types of code *blocks*, sometimes called *fenced code blocks*. A fenced code block is a set of several lines, with a triple-backtick as the first line, and another triple-backtick line as the last line.

The first triple-backtick line should have a trailing *colorizer code*, sometimes called a *language specifier*.

- *Regular executable SQL example* - In a code block for a regular SQL code example that the customer could perhaps execute, the colorizer code is `sql`.
- *Syntax description of SQL* - In a code block for an SQL syntax description, the colorizer code is `syntaxsql`.
  - `syntaxsql` became available on March 17, 2020.
  - An example follows.

````
```syntaxsql
DROP DATABASE [ IF EXISTS ] { database_name | database_snapshot_name }
    [ ,...n ] [;]
```
````

For a list of available language codes for fenced code blocks, see the devlang allowlist. To request new allowlist values, email DevRelIA as described in How to request metadata changes.

Contributor resources

- Contributor Guide for Docs
- Markdown basics

Next steps

Find an article, submit a change, and help improve our SQL Server content for customers.

Questions? Email sqldocsPR.

Thank you!

SQL versioning guidance and scenarios for writers

6/11/2021 • 14 minutes to read

This article explains the effects of the OPS Conceptual Versioning system to Writers, emphasizing the particular esoteric needs of the **MicrosoftDocs/sql-docs-pr** GitHub repo.

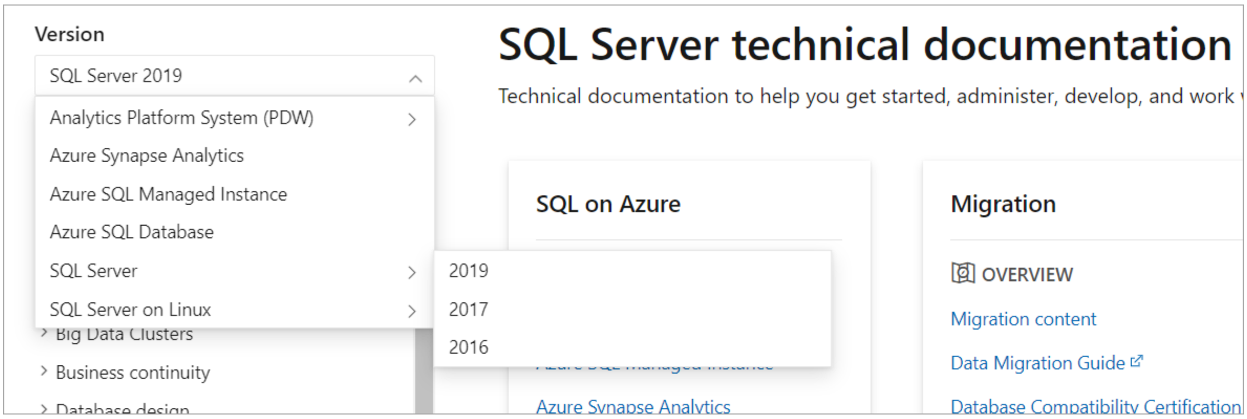
The emphasis is on the *decisions* and *actions* that Writers must take to cooperate with the Versioning system. However, you have to gain and understand basic knowledge about the Versioning system in **sql-docs-pr** before we can explain those decisions and actions.

If you want a quick overview of how we've implemented versioning or if you have an update you need to make in a single article, see the SQL versioning quickstart section.

For regular contributors, it's important to review how we implemented versioning and the special scenarios in which we use versioning. See the SQL versioning guidance section.

SQL versioning quickstart

SQL Server content has implemented versioning. Users can select a SQL Server service or product version from the left-hand TOC to scope the content that Docs displays.



You can associate SQL services and product versions with monikers. The following table explains the current monikers that you can use in SQL articles to control versioning:

AREA	MONIKER	ASSOCIATED SERVICE OR PRODUCT VERSION
All	`sqlallproducts-allversions`	Everything. No filter.
SQL Server	`sql-server-ver15`	SQL Server 2019
	`sql-server-2017`	SQL Server 2017
	`sql-server-2016`	SQL Server 2016
	`sql-server-previousversions`	SQL Server previous versions
SQL Server (Linux)	`sql-server-linux-ver15`	SQL Server 2019 for Linux

AREA	MONIKER	ASSOCIATED SERVICE OR PRODUCT VERSION
	`sql-server-linux-2017`	SQL Server 2017 for Linux
SQL Database	`azuresqldb-current`	Azure SQL Database
	`azuresqldb-mi-current`	Azure SQL Database Managed Instances
SSAS	`sql-analysis-services-2016`	Analysis Services for SQL Server 2016
SQL Server PDW	`aps-pdw-2016`	APS and PDW for SQL Server 2016-AU6
	`aps-pdw-2016-au7`	APS and PDW for SQL Server 2016-AU7
SQL DW	`azure-sqldw-latest`	Azure SQL Data Warehouse

Versioning happens at three levels:

- Folder
- Article
- Zone

Folder-level defaults

Each folder in the repo has a versioning default assigned in the **sql-docs-pr/docs/docfx.json** file. As a result, new articles are automatically versioned just by their location.

If you think there should be changes to folder defaults, contact the SQL content team. Changes are often more complex than they appear at a surface level.

Article-level metadata

When the folder default versioning doesn't work for an article, you can add a **monikerRange** metadata value to the article. The metadata value overrides the folder-level defaults. Consider the following portion of metadata:

```
author: "craigg-msft"
ms.author: "craigg"
monikerRange: ">= sql-server-linux-2017 || >= sql-server-2016 || = sqlallproducts-allversions"
---
```

In this example, the article applies to:

- Linux for SQL Server 2017 and greater: `>= sql-server-linux-2017`
- All versions of SQL Server 2016 and greater: `>= sql-server-2016`
- User selected All SQL: `= sqlallproducts-allversions`

Zone-level versioning within articles

Inside an article, you can view version-specific **zones** of text. Consider the following examples:

```
::: moniker range="=sql-server-2017"
|What's new  | Release notes  |
|---------|---------|
|[What's New in SQL Server 2017](../sql-server/what-s-new-in-sql-server-2017.md) |[SQL Server 2017 Release Notes](../sql-server/sql-server-2017-release-notes.md)|
::: moniker-end

::: moniker range="=sql-server-2016"
|What's new  | Release notes  |
|---------|---------|
|[What's New in SQL Server 2016](../sql-server/what-s-new-in-sql-server-2016.md)| [SQL Server 2016 Release Notes](../sql-server/sql-server-2016-release-notes.md)|
::: moniker-end
```

The examples use the same moniker range concept, but it applies to zones inside the article. Docs displays a different table when it's scoped to SQL Server 2017 versus SQL Server 2016. Docs customizes the user's experience.

> **TIP**
>
> For a complete example, see the markdown for the main SQL Server Documentation article. Experiment with the versioning selector on the left. Then select the **Edit** button on the right to view the markdown (just don't submit any edits).

SQL versioning guidance

The following sections go into more detail about concepts behind SQL versioning and scenarios involving versioning. It's important that all regular contributors review and learn this information.

The **sql-docs-pr** repo has the documentation content for Microsoft SQL Server 2016 and all later versions. Much of the documentation also applies to related products or services like Azure SQL Database. And before June 2018 the repo will also have the content for SQL Server 2014. Writer decisions and actions must account for the added residence of SQL 2014, sometimes.

> **NOTE**
>
> Versions 2005-2012 are available in the SQL Server previous versions documentation. But these much older versions do not participate in Versioning.

If you have read this article before, and just want a quick refresher, see the short Summary of actions near the end of this article.

SQL has complex Versioning requirements

Most other documentation repos have simpler requirements for Versioning than does our **sql-docs-pr**. The reasons include:

- Our high-level directories or folders under **sql-docs-pr/docs/** were never designed to be strict groupings of articles by the Versioning attributes. It isn't clear whether such a goal would even be realistic. But many other repos inherently have such high-level folders.
- The family of SQL Server and the different closely related products is more diverse and nuanced than the families of other repos. SQL includes:
    - On-premises.
    - Two products in the Azure cloud, one being Azure SQL Database Managed Instance.
    - Offshoot products like Azure SQL data warehouse.

- A Linux-based product.
- Several tools that might or might not quite be a direct part of the primary SQL Server product.
- Tools you can install like SSMS. They're designed so that customers never have any motivation to use an older version.

- Many articles apply only to SQL Server on-premises, and don't apply to Azure SQL Database.
- In the **sql-docs-pr** repo, the high-level folders under **docs/** were never designed to group articles by their Versioning attributes.

Prerequisites

OPS documentation

Per the rules of the internal Contributor Guide, this article contains as little redundancy as possible with the general OPS Versioning system docs. You should read the general Versioning documentation first:

- Conceptual Versioning

If you already know a little about OPS Versioning, you might learn everything else you need to know just by reading this article.

Terminology

As you read this article, you'll come across several precisely used terms. Whenever you're unsure about a term, see Terminology of OPS Versioning.

docfx.json encodes the default monikers

The following snippet of JSON code is an abbreviation of the **fileMetadata** element in our real **docs/docfx.json** file.

```
"fileMetadata": {
  "monikerRange": {

    "2014/**/*.md": "=sql-server-2014 || =sqlallproducts-allversions",

    "relational-databases/**/*.md": ">=sql-server-2016 || =sqlallproducts-allversions",

    "t-sql/**/*.md": "=azuresqldb-current || >=sql-server-2016 || =sqlallproducts-allversions"
  }
},
```

> **WARNING**
>
> By glancing at the high-level folder names in the JSON **fileMetadata**, you can see how dangerous it could be for a Writer to rename a high-level folder.

> **NOTE**
>
> Many other GitHub repos that implement OPS Versioning do not use the **fileMetadata** element. Instead, the majority rely on the **Groups** JSON element to assign default moniker ranges to high-level folders, or to their equivalent of our **docs/** folder. Our **docs/docfx.json** does use the **Groups** JSON element, but only lightly to support **fileMetadata**.

Action: Move a file

In this section, we refer to the earlier JSON that has the **fileMetadata** element from **docs/docfx.json**.

Suppose we have an article **sql-article-1.md** under the **docs/t-sql/** folder. The article has no moniker range specified within its MD file. So system assigns the moniker range we've specified in the **fileManager** JSON file of the **t-sql/** folder. The default moniker range is correct for the file. The article applies to both SQL Server on-premises and Azure SQL Database.

Then we want to move the **sql-article-1.md** file laterally, from **docs/t-sql/** to **docs/relational-databases/**. This move puts the file under a different high-level folder. In the **fileMetadata** element, its new folder has a different moniker range than its old folder has. And the new range doesn't exactly match the Versioning needs of the article. So when we move the MD file, we must add a moniker range to the metadata of the file. In this particular case, a copy of the moniker range from **t-sql/** is exactly the range we should put inside the article that we move. See the **monikerRange:** name in this example YAML metadata:

```
---
title: All about SQL article 1 | Microsoft Docs
monikerRange: =azuresqldb-current || >=sql-server-2016 || =sqlallproducts-allversions
.....More metadata here.....
---
# SQL Article 1
```

NOTE

Avoid adding a **monikerRange** into an article's YAML metadata when it would merely duplicate all the same moniker comparators that the article otherwise inherits from its high-level folder.

WARNING

For now we are ignoring an important coordination consideration that exists because of the existence of SQL Server 2014 documentation under **docs/2014/**. See the later section about SQL 2014.

Move a file lower

If you move a file from **docs/relational-databases/** further down to **docs/relational-databases/in-memory/**, there are no Versioning considerations related to monikers. But there could be considerations involving SQL 2014 under **docs/2014/**, as explained later.

Action: Add a file

Given the preceding explanation of *moving* a file, the following steps should suffice for *adding* a new file:

1. Decide under which high-level folder to store the new MD file in the **sql-docs-pr** repo.
2. Go to the **fileMetadata** element in the **docs/docfx.json** file.
3. Assess whether the default moniker range of the high-level folder exactly matches the needs of your new article.
   - If yes, then you have no more Versioning actions to take. Just add the file as usual.
   - If no, then add a **monikerRange** into the YAML metadata of the new article.

Which moniker comparators can you use?

The only valid list of moniker comparators you can choose from exists in the **Groups** element of the **docs/docfx.json** file!

Action: Rename a file

Renaming a file has Versioning considerations only if the same *path/filename.md* exists for both SQL 2016+ and SQL 2014. See the later section about SQL 2014.

Action: Remove a file

There are no Versioning considerations when you remove an obsolete MD file. SQL Server 2014 documentation under **docs/2014/** causes no considerations either.

SQL Server 2014 and Versioning complications

All the documentation for SQL Server 2014 is under **docs/2014/**. No documentation for any other product or version is under **docs/2014/**.

The only regular moniker comparator for the **2014/** folder is the equality to SQL Server 2014: = **sql-server-2014**.

> **NOTE**
>
> Ideally the set of 10,000 articles under **docs/2014/** would be entirely self-contained. However, all SQL 2014 articles that contain links to a Transact-SQL article link out to articles under **docs/t-sql/**. Also, as of May 2018, some 2014 articles link to **docs/mref/** (although when OPS Reference Versioning is fully implemented, this might change).

Paths of SQL 2014 and 2016 must correspond

About 80% of the 10,000 *path/filename.md* items under **docs/2014/** have an exact match in SQL Server 2016 documentation under **docs/**. By *match* we mean both of the following two conditions:

- The **assetGuid** from CAPS for the articles that migrated to **docs/2014/** matches the YAML **ms.assetid** in a SQL 2016 article elsewhere under **docs/**.
- From the end of their respective HTTP root folder paths in the OPS system, the rest of their HTTP addresses are identical to each other.
  - In the **sql-docs-pr** repo, for all articles under **docs/2014/**, the OPS portal at https://ops.microsoft.com/ shows that the GitHub folder **docs/2014/** maps to the HTTP address node of **sql/**.
  - For all articles elsewhere under **docs/**, the OPS portal shows that the GitHub folder **docs/** maps to the *same* HTTP address node of **sql/**.
  - For example, https://docs.microsoft.com/**sql/**relational-databases/in-memory/cool-sql-2.

These matches or *article version pairs* work in the Versioning system only because both articles have the exact same complete HTTP address as each other. Well, the one difference is their value for the URL parameter **?view=**. Here *view* means *the product and version of the article to display*. The parameter value is a moniker known to the OPS system.

Next is an informal example that shows an *article version pair* as it could look in GitHub. The two GitHub folder paths match identically where the **relational-database/** folder starts.

```
docs/     relational-databases/in-memory/cool-sql-2.md

docs/2014/relational-databases/in-memory/cool-sql-2.md
```

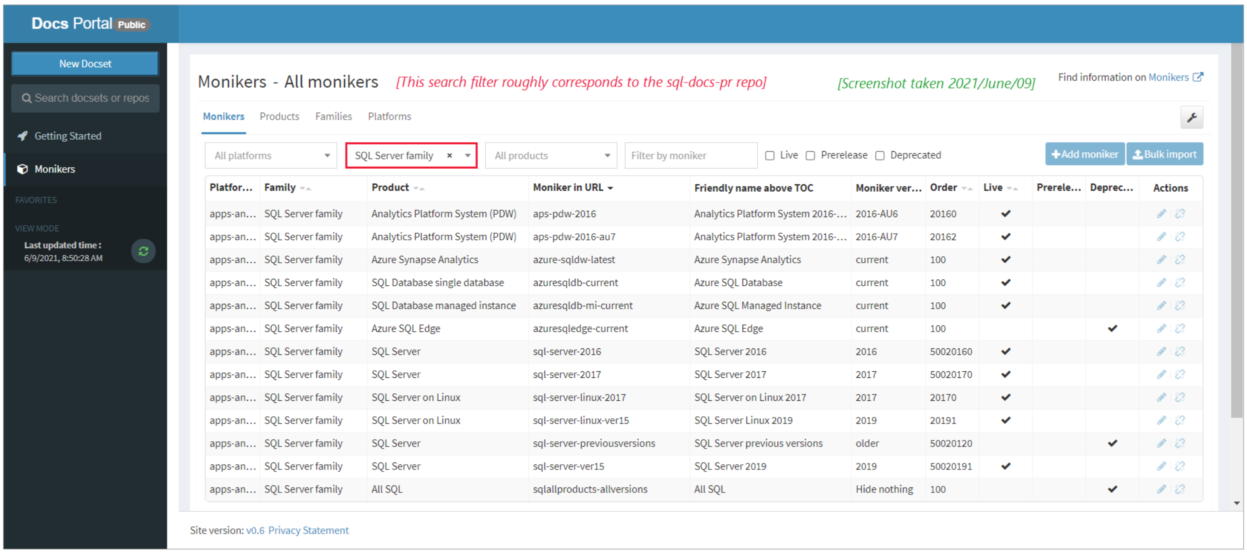
SQL 2014 migration guided by correspondence

As we migrate the **SQL14** portfolio from CAPS into repo *sql-docs-pr* in April-May 2018, we're careful to match on SQL 2014's articleGuid from CAPS to the **ms.assetid** metadata left from the earlier February 2017 migration of portfolio **SQL Server 2016**. Whenever we find a guid match, we carefully place the SQL 2014 article under **docs/2014/** in the same corresponding path where the SQL 2016 article already is elsewhere under **docs/**.

OPS portal for our packages

The only monikers that the Versioning system recognizes and honors are the monikers that we've tagged as *Live* on the [Packages display in the OPS portal](). The monikers that are useful to our **sql-docs-pr** repo are part of our *Family* named **SQL Server family**. Technically we of the **sql-docs-pr** repo don't officially *own* the **SQL Server family**, but in practice we should and do own it. Outsiders shouldn't be adding or changing monikers in **SQL Server family**.

Next is a screenshot of our packages in the portal.



Platfor...	Family	Product	Moniker in URL	Friendly name above TOC	Moniker ver...	Order	Live	Prerele...	Deprec...	Actions
apps-an...	SQL Server family	Analytics Platform System (PDW)	aps-pdw-2016	Analytics Platform System 2016-...	2016-AU6	20160	✔			
apps-an...	SQL Server family	Analytics Platform System (PDW)	aps-pdw-2016-au7	Analytics Platform System 2016-...	2016-AU7	20162	✔			
apps-an...	SQL Server family	Azure Synapse Analytics	azure-sqldw-latest	Azure Synapse Analytics	current	100	✔			
apps-an...	SQL Server family	SQL Database single database	azuresqldb-current	Azure SQL Database	current	100	✔			
apps-an...	SQL Server family	SQL Database managed instance	azuresqldb-mi-current	Azure SQL Managed Instance	current	100	✔			
apps-an...	SQL Server family	Azure SQL Edge	azuresqledge-current	Azure SQL Edge	current	100			✔	
apps-an...	SQL Server family	SQL Server	sql-server-2016	SQL Server 2016	2016	50020160	✔			
apps-an...	SQL Server family	SQL Server	sql-server-2017	SQL Server 2017	2017	50020170	✔			
apps-an...	SQL Server family	SQL Server on Linux	sql-server-linux-2017	SQL Server on Linux 2017	2017	20170	✔			
apps-an...	SQL Server family	SQL Server on Linux	sql-server-linux-ver15	SQL Server Linux 2019	2019	20191	✔			
apps-an...	SQL Server family	SQL Server	sql-server-previousversions	SQL Server previous versions	older	50020120			✔	
apps-an...	SQL Server family	SQL Server	sql-server-ver15	SQL Server 2019	2019	50020191	✔			
apps-an...	SQL Server family	All SQL	sqlallproducts-allversions	All SQL	Hide nothing	100			✔	

Beyond our family

We own monikers beyond just the ones in the **SQL Server family**. Other monikers we own and use include:

- azure-sqldw-latest
- aps-pdw-2016

When you need to add a moniker range in the YAML of an article's MD file, you're restricted in which moniker comparators you can use. You can find the list of moniker comparators that you can choose from in the **Groups** element of the **docs/docfx.json** file.

Not involving editions in our Versioning

For the **sql-docs-pr** repo, we designed our monikers to provide filtering at the level of *product* and *version*. We found those two dimensions complicated enough. We believe it's unrealistic to reach for the more granular concept of *edition* which are the differences between the on-premises SQL Server editions:

- *Developer*
- *Standard*
- *Enterprise*

Most or all other teams at Microsoft have come to the same conclusion.

sqlallproducts-allversions is a special moniker

After the OPS Versioning implementation, each article webpage shows a drop-down box containing clickable lines of product and version combinations. When the customer selects a different line, Docs might instantly reduce the TOC to display only the subset of articles that have a moniker range that includes the chosen product and version line. The Versioning requirements for the **sql-docs-pr** repo are complex enough that sometimes the customer could become frustrated by their inability to make the whole TOC reappear (it's a long explanation).

To mitigate this ongoing frustration of a much hidden TOC, we added the special moniker *sqlallproducts-allversions* into our family of monikers. In the `sql-docs-pr` repo, we always include the moniker comparator of `= sqlallproducts-allversions` in every moniker range, at all precedence levels! Most other repos have no similar moniker, but at least one other repo does.

> **WARNING**
> All our moniker ranges, at all precedence levels, must include the moniker comparator of `= sqlallproducts-allversions`!

Live examples of precedence levels

How to find which articles contain a YAML moniker range

Find by using a cmd.exe window

```
cd sql-docs-pr
cd docs
findstr /s /l /i /n /c:"monikerRange" *.md
```

Find by using PowerShell

The following code example shows long syntax commented out, and then the equivalent short syntax.

```
# Get-ChildItem -Recurse:$true -Include "*.md" | Select-String "monikerRange";
gci -r -i "*.md" | sls "monikerRange";
```

Find by using a bash.exe window

```
grep 'monikerRange' -E -r *
```

How to find which articles contain a ::: zone

By using a `cmd.exe` window, you can find articles that contain a Versioning *zone*.

```
findstr /s /l /i /n /c:":::" *.md
```

Terminology of OPS Versioning

- *Conceptual Versioning:* Is distinct from *Reference Versioning*, which applies to autogenerated reference content.
- *moniker* The unique ID or unique name for each Package row in the OPS portal. A moniker is a combination of *product* and *version*, for example `sql-server-2017`. A moniker of a *shippable unit*.
  - By itself, a moniker doesn't include any relational operator or conjoining operator.
- *Moniker-Friendly Name:* A seemingly irrelevant column of values in the OPS portal for Packages. We care about the monikers as they appear in the column *Moniker Name*.
- *moniker comparator:* For example, `>= sql-server-2016`. A moniker comparator is the combination of a relational operator and moniker. It's equivalent to saying a node in a moniker range. See *moniker range*.
- *moniker range:* A string of one or more moniker comparators, conjoined by a conjoining operator.

    - For example, the moniker range of >= **sql-server-2016 || = azuresqldb-current** has two *moniker comparators*. We've conjoined the two comparators by the conjoining operator OR.
- *operator:* There are two kinds of operators used with monikers to form ranges. Consider the example *moniker comparator* of >= **sql-server-2016 || = azuresqldb-current**:
    - *relational operator:* The >=, meaning later than or equal to. Absence defaults to =.
    - *conjoining operator:* The **||**, meaning boolean OR. Absence defaults to AND. The explicit syntax **&&** for AND is invalid. AND is rarely used.
- *OPS:* Open Publishing System of our Microsoft group named *Content & Learning*.
- *OPS portal:* https://ops.microsoft.com/
- *precedence hierarchy:* You can specify moniker ranges at increasingly granular levels. Levels that are more granular take precedence over levels that are less granular. Only one precedence level applies in any given scenario. There's *no additive effect* from the less granular levels when a more granular level takes precedence. Only the less granular moniker range applies.
    - *Folder-based:* The least granular level. It's specified in the **docfx.json** file.
        - Unless it's explicitly set up otherwise, the folder moniker range cascades down to all its child and other descendant folders in GitHub.
    - *Article YAML-based:* The metadata value **monikerRange** in the YAML at the top inside at article's MD file.
    - *Zone-based:* A ::: delimited section within the body of an article. It's the most granular level.

Summary of actions

- Ensure that any two sister articles, where one exists under **docs/2014/** and the other exists elsewhere under **docs/**, remain in equivalent corresponding *path/filename.md* values.
- Sometimes an article comes to exist under a high-level folder whose moniker range from docfx.json doesn't exactly match the Versioning needs of the article. In such cases, add a **monikerRange** vale to the YAML metadata inside the article's MD file.
- If one section of your article needs a different moniker range, use the ::: syntax.

Related links

- Versioning & Monikers

Update documentation for a servicing update

11/2/2020 • 3 minutes to read

This article explains how to submit a documentation update for a servicing update in the SQL Server repository, **MicrosoftDocs/sql-docs-pr**. A servicing update is either a cumulative update (CU) or service pack (SP).

The steps in the article demonstrate how to submit a pull request to a release branch with your browser.

Before you start

To complete these tasks, you need a GitHub account belongs to both the **Microsoft** and the **MicrosoftDocs** organizations. To configure your account, follow the instructions in GitHub account setup.

Identify your release branch

Update content for a servicing update in a release branch. When the product group releases the servicing update to general availability (GA), the SQL Server content team merges the release branch into the default branch. The updated content becomes public at the next normal publishing cycle. See Pull request review and publishing schedules.

> **NOTE**
>
> At this time, the default branch for sql-docs-pr repository is `master`. Eventually the name of the default branch will be `main`.

To update the documentation for a servicing update, locate the appropriate branch in GitHub. The sql-docs repository has servicing branches for CUs and SPs.

- SQL Server 2019: sql-docs-pr/release-2019-cu
- SQL Server 2017: sql-docs-pr/release-2017-cu
- SQL Server 2016: sql-docs-pr/release-2016-cu

For example, to submit an update to the machine learning (ML) cabinet (CAB) file for the next SQL Server 2019 CU, select the link above for SQL Server 2019.

> **NOTE**
>
> **sql-docs-pr** has release branches for the most frequently updated products. The label **-cu** on the release branch name is applied because most updates are CUs. If your update is to a service pack, or other type of release, contact the SQL Server content developers for information about the appropriate release branch.
>
> Use these branches for whatever the next CU will be for a version. The branches publish when a CU releases. Stage the update in the appropriate branch, so that it publishes in sync with the update release to GA.
>
> Do not use these branches for a CU that is further in the future than next. All changes in these branches becomes public on the day the CU releases.

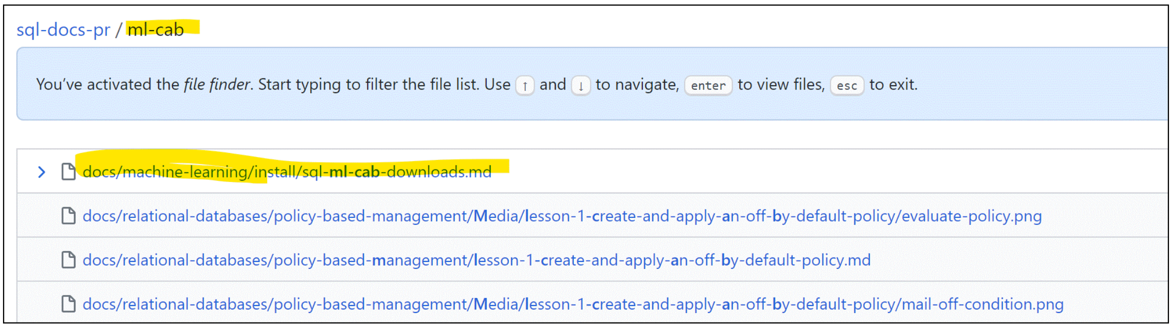
Locate the file you want to update

To locate the file in the repository on GitHub:

1. Verify that you are in the correct branch.

2. Select **Go to file**.

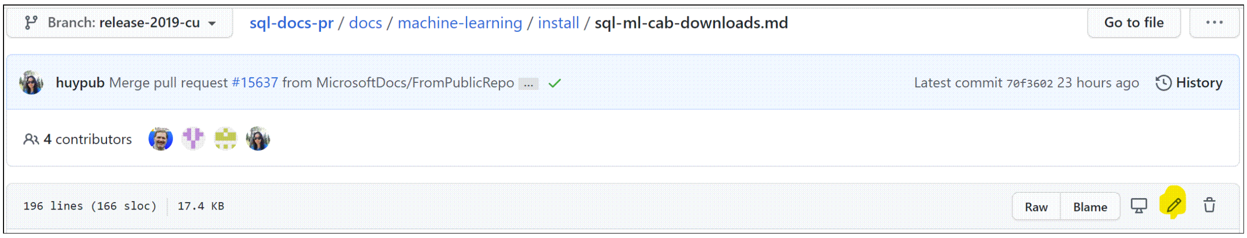
3. Type part of the file name to locate the file you want to update in this branch. For example, to locate *sql-ml-cab-downloads.md*, type `ml-cab`.



4. Select the file.

Update the file

In the GitHub page for the file, locate the pencil icon, and select it to edit the file.

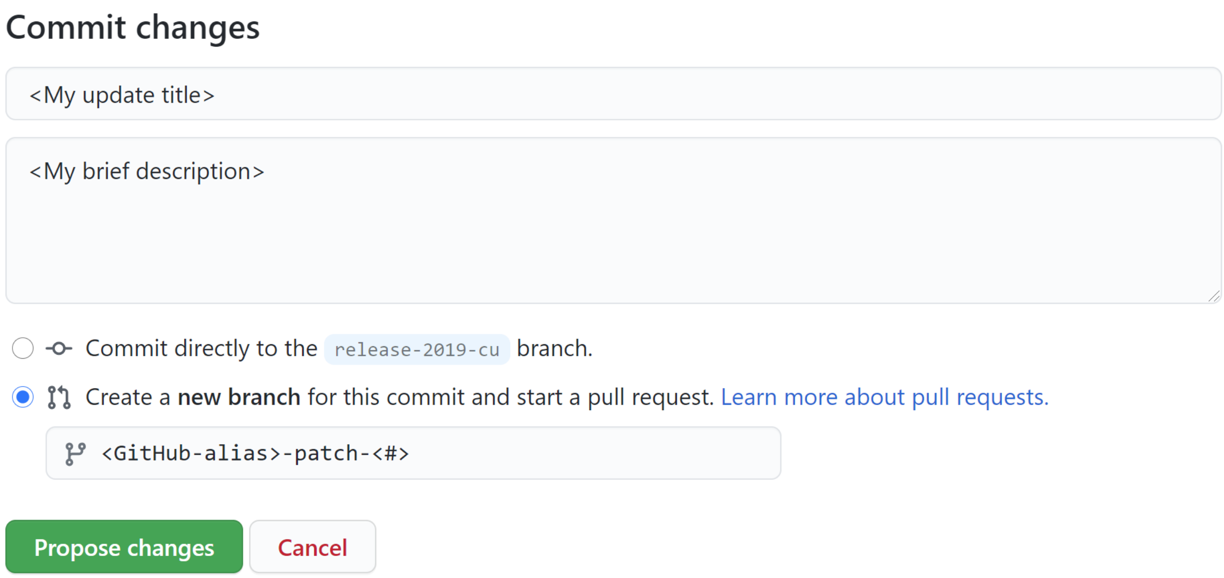


In the browser, GitHub provides an interface for you to update the content of the article in Markdown. For an introduction to Markdown, see Docs Markdown reference.

Submit the pull request

After you update the file, go to the bottom of the page in GitHub. Under **Commit Changes**, type a brief description of your update. If you need more space, use the second field.

Choose, **Create a new branch for this commit and start a pull request**.



Select **Propose changes**.

Check the pull request

After you submit the pull request, the open publishing system (OPS) validates the build. This process normally takes about 10 minutes. After validation, you can preview your updated content from the links in GitHub. The preview links are in the pull request under **Conversation**.

If your content validates correctly, you can sign-off on the content.

> **TIP**
>
> You can find all of your pull requests at https://github.com/pulls.

Sign-off and publish

To sign-off on a pull request, wait for it to validate. In the pull request **Conversation**, add `#sign-off` as a comment. After you sign-off, the open publishing system notifies the reviewers to review and merge the content. OPS automatically merges small changes. Larger changes require a human reviewer.

An automated workflow routes the pull request to the reviewers. The reviewer will either merge the update or block the update. If the update is blocked, the reviewer will add a comment to explain the blocking violations. For more information, see Pull request submission best practices.

If you have any questions, email the SQL content team (sqldocs).

Related links

Get started contributing to SQL Documentation

ILDC-specific publishing and pull request procedures

5/10/2021 • 2 minutes to read

Pull request review

To view the standard Redmond-based pull request review schedule translated into ILDC hours, see the PR review and publishing schedules article.

To facilitate the workflow on Sundays in Israel and to compensate for the usual 10-hour time differential, IDLC team members can request local pull request review.

The designated ILDC pull request reviewers are authorized to merge content in the following repositories:

- azure-docs-pr
- cloudappsecurity-docs-pr
- atadocs-pr

When the ILDC pull request reviewer merges a PR, they assign the **ILDC-PR** label in GitHub. The Redmond-based PR review team will review these merged PRs daily and provide any additional feedback necessary directly to the author via a comment in the merged PR with an @mention to the author requesting follow up.

Publishing

To view the standard Redmond-based publishing schedule translated into ILDC hours, see the PR review and publishing schedules article.

The ILDC team determines when publishing occurs for the following repositories because they are wholly managed by the content team:

- cloudappsecurity-docs-pr
- atadocs-pr

If the ILDC team needs to publish Azure content from the azure-docs-pr repository on Sundays or other days of the week, the team has a couple options that allow publishing to occur without pushing the entire contents of the Azure-docs-pr master branch live. These options preserve the ability of US-based writers to correctly manage their planned release times.

- **Sunday publishing branch**: There is an ongoing release branch named **release-ildc-sunday** that members of the ILDC team can use for all content in the Azure repo that they want to release during the Sunday business day in Israel. The designated ILDC pull request reviewers are authorized to publish on Sundays using these steps. This process ensures that the commits for Sunday publishing go into live and master without releasing any other pending content in master:
  1. Create a pull request from the **release-ildc-sunday** branch directly to **live** and merge after a successful build.
  2. Create a pull request from the **release-ildc-sunday** branch directly to **master** and merge after a successful build.
- **Hotfix publishing**: The designated ILDC pull request reviewers can follow the standard process for hotfix publishing to publish content as needed during the ILDC business day.

If necessary, the ILDC team can use the Sunday publishing branch model with release branches if an ILDC release needs to occur during a US holiday or to target a specific event that occurs during ILDC business hours.

Content freezes

The ILDC team must respect content publishing freezes that occur as part of a planned release. The team should work with Redmond-based release management if an exception to the publishing freeze is required. Contact the designated release manager for the event to work out a solution.

Get started contributing to Visual Studio documentation

5/18/2021 • 2 minutes to read

This article provides guidance on how Microsoft employees can contribute to Visual Studio documentation published at https://docs.microsoft.com/visualstudio/. Visual Studio and related documentation is managed in the following Git Hub repositories:

- visualstudio-docs-pr: Visual Studio content
- visualstudio-docs-dotnet-api-pr: Visual Studio SDK reference content
- intellicode-pr: IntelliCode content
- liveshare-docs: LiveShare content

This guidance applies to the visualstudio-docs-pr repo.

> **NOTE**
> Visual Studio Code documentation is not on the docs platform. For authoring information, see Visual Studio Code documentation on GitHub.

General tasks

If you need to make a minor change to an existing article, see Quickly edit an existing article. For all other edits, use the following resources in the Contributor's Guide:

TASK	CONTRIBUTOR'S GUIDE RESOURCES
Setup a local repository for larger or frequent edits	Sign up for a GitHub account Install Git and VS Code Set up a Git repository locally
Submit larger or frequent edits	Create working branches and submit pull requests Edit content in Visual Studio Code
Follow content standards	Write an overview Write a quickstart Write a tutorial Write a sample Edit a TOC
Improve your writing	Follow writing principles Use Acrolinx Review the Microsoft style guide

> **TIP**
> If you are adding new content or making significant changes, please include a member of the content team on this work to avoid duplication and to align with ongoing content strategies.

Visual Studio tasks

In addition to the general writing tasks in the previous section, there are writing tasks and resources specific to Visual Studio documentation.

TASK	VISUAL STUDIO DOCS RESOURCES
Ask a question	Email the Visual Studio content team
View current work	View the Visual Studio and VS Code dashboard
See staged content	Visit the review site and select a branch
Visual Studio writing tasks	Edit metadata in Visual Studio articles Manage versioning Work with Visual Studio release branches

Next steps

For more information, see the Contributor's Guide.

Visual Studio docs guidance

6/1/2021 • 6 minutes to read

This article provides guidance for Microsoft employees who contribute to Visual Studio documentation in **visualstudio-docs-pr**.

How to contribute

You can contribute to directly to Visual Studio docs either by editing in your browser or by working locally with VS Code and Git Bash. For more information, see Get started contributing to Visual Studio Documentation.

> **IMPORTANT**
>
> If you need to make changes for a future release, please use an appropriate release branch and follow the steps on working with release branches in this article.

Author publishing process

Authors refer to doc owners, such as content writers, or program managers (PM) from the product group. Authors should follow the publishing process outlined in this section. Writers that do not fit into these two categories should follow the contributor guidance.

To submit your documentation change, open a GitHub pull request (PR). For more information, see How to create a PR. Please note the following:

- After you complete your changes, you must add #sign-off in the comments of the PR.
- The PR is reviewed for standards before merging. You will be asked to fix any issues.

After your changes are merged, they are ready to publish. Twice a day, the **master** branches of **visualstudio-docs-pr** and are pushed to the public repositories. This publishes the content live. Publishing happens at roughly 10:30am and 4:30pm PST.

> **NOTE**
>
> If you are working in a release branch, then the content is not published until that release branch is merged into **master**.

Email contact

Use the vs-content email alias to discuss technical content and PRs related to **visualstudio-docs-pr** and **visualstudio-docs**.

Visual Studio workitem tracking

To see what is happening in Visual Studio content, access the Visual Studio and VS Code Azure Boards Dashboard.

If you have a new content idea or need to partner with the content team in any way, contact the Visual Studio Content Team.

Visual Studio docs guidance

The following section provides some additional guidance on working in the **visualstudio-docs-pr** repository.

The visualstudio-docs-pr repository uses several standard folders to organize the content.

FOLDER	DESCRIPTION
docs/	Contains all published Visual Studio content. Subfolders logically organize different areas of the content.
docs/includes/	Contains include files. These files are blocks of content that can be included in one or more other topics.
./media/	Each folder can have one **media** subfolder for article images. The **media** folder in turn has subfolders with the same name as the topics that the image appears in. Images should be .png files with all lower-case letters and no spaces.
toc.yml	A table-of-contents file, in the newer YAML format. Each subfolder has the option of using one toc.yml file.

Contributor guidance

Product group PMs or content developers approve all updates in the private repository. Contributors who are not in the product group may add PMs or content developers as reviewers, but they may not sign off on new content or updates. This policy ensures that content aligns with product group intent and supportability.

Procedure

Once a PR has been submitted, you must:

1. Check the Acrolinx scorecard and ensure to implement the suggested changes.
2. Review the staged content to ensure formatting is okay and your changes look good. Do this by selecting the "View" link in the **Validation status: Passed** comment under **Preview URL** within your PR.
3. Check the metadata for who the ms.author of the article is, and email them directly with the link your PR.

Guidelines

These are general guidelines that all authors must follow, but that are not explicitly stated elsewhere.

- When providing a link to content, content on Docs takes priority. Do not provide a link to a blog if a Docs article exists. If providing a link to a blog, the blog should be vetted by the article author for technical accuracy. Ideally, the content of the blog should be migrated to Docs, but the author needs to make that call.

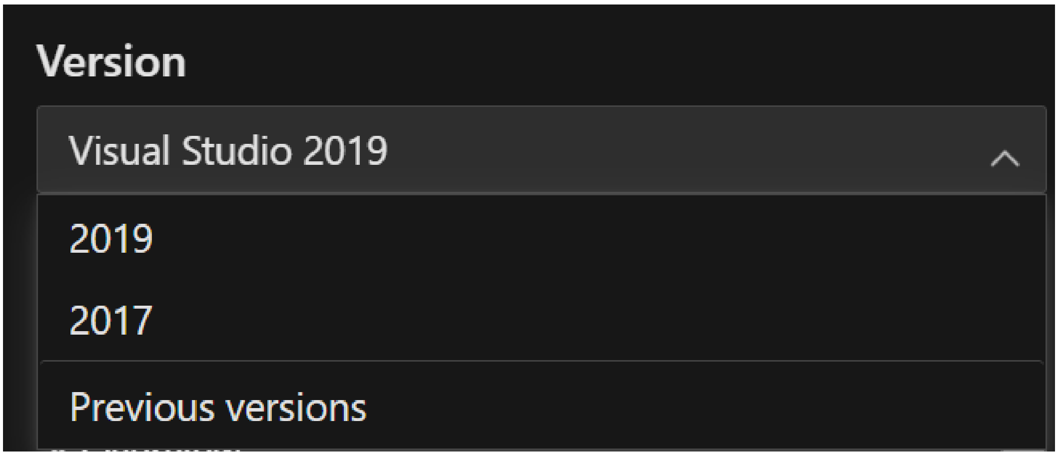
Article metadata

Each article has a metadata block at the top. For detailed metadata information, see Metadata guidance. In the near future, we will post more specific metadata guidance for Visual Studio topics here. For now, use metadata in a similar way as existing Visual Studio articles in your area.

Versioning

Visual Studio content has implemented versioning. Users can select a Visual Studio service or product version from the left-hand TOC to scope the content that is displayed.

In our documentation, we create content that applies to several recent versions of Visual Studio. Most of the time, that content is uniform across versions. But sometimes, it isn't. To make sure that folks can see content for the version of Visual Studio that they are using, we include a version drop-down control on the left side of the page so that they can do so.



To provide this versioning experience, we use monikers in Markdown to tag content that applies to a specific version. For example, we currently publish content for Visual Studio 2017, Visual Studio 2019, and Visual Studio 2022 (Preview as of this writing). So, we use the following monikers to offset sections or pages for specific versions.

- vs-2017
- vs-2019
- vs-2022

Here's an example of how we use them to set off text:

```
::: moniker range="vs-2017"

If you want to check that your unit test is working, chooseTest>Run>All Testsfrom the menu bar. A window
calledTest Exploreropens, and you should see that theTestGetCurrentDatetest passes.

::: moniker-end

::: moniker range="vs-2019"

If you want to check that your unit test is working, chooseTest>Run All Testsfrom the menu bar. A window
calledTest Exploreropens, and you should see that theTestGetCurrentDatetest passes.

::: moniker-end
```

When using monikers for product changes that are expected to persist to future versions, use the moniker range syntax that specifies versions greater than or equal to the current, as shown in the following example. Otherwise, the content in the moniker block will not be included in the docs for future releases.

```
::: moniker range=">=vs-2022"
```

NOTE

Monikers sometimes don't work well in lists, such as numbered lists or bulleted lists, or in a table. In many cases, it's best to repeat an entire section in a moniker tag rather than to try to disambiguate entries in a list or in a table.

Branches

Visual Studio content uses the following branches for releasing content. The instructions in this guide use the **master** branch, which is for immediate release. But there are also release branches for delayed releases.

BRANCH	DESCRIPTION
master	Changes publish twice a day from this branch.
release-16.xx	Changes for the next update of Visual Studio 2019.
release-dev17-preview1	Changes for Preview 1 of Visual Studio "Dev17".

Work in a release branch

Use the following steps to work in a release branch like **release-16.xx** or **release-dev17-preview1**.

1. If you have not locally cloned the **visualstudio-docs-pr** repository and installed Git Bash and VS Code, see Set up a local repository for larger or frequent edits.

2. Follow the steps in Check out a release branch to check out a working branch for your release branch. For example, if you want to make changes to Visual Studio Dev17, you could run the following command to create a working branch **mychanges** for **release-dev17-preview1**:

```
cd /visualstudio-docs-pr/docs
git checkout -B mychanges upstream/release-dev17-preview1
git pull upstream release-dev17-preview1
```

3. Make all of your changes and then Commit and push your working branch.

4. Then create a PR for your changes. It helps to add `@githubusername` to the comments for any content lead or PM associated with the change to facilitate awareness and collaboration.

   Caution

   You must ensure that your PR pulls your changes into the corresponding release branch, such as **release-dev17-preview1**, and not **master**. If you accidentally pull your changes into **master**, all of the pre-release content will go live (not just your PR). Having a lot of extra commits that are not associated with your change is a clue that you have made this mistake.

If you have any questions or need any help on how to work with release branches, email the repo admin for visualstudio-docs-pr.

Contributor resources

- Contributor Guide for Docs
- Markdown basics

Next steps

Find an article, submit a change, and help improve our Visual Studio content for customers.

Questions? Email the repo admin for visualstudio-docs-pr.

Thank you!

Principles for publishing preview content for Visual Studio and VS Code on docs.microsoft.com

5/13/2021 • 4 minutes to read

As a rule, the Visual Studio & VS Code content team does not author preview content on docs.microsoft.com or code.visualstudio.com/docs. Preview content could be content associated with a specific preview release, or with an insider build, or an a/b test or experimental feature that only some users see.

Have questions? Reach out to: vs-content@microsoft.com.

Why we don't publish preview content

We don't publish preview content because it:

- Changes more often. Product churn can result in costly doc churn, still too unstable for scarce writer resources
- Usually has a small audience, and therefore is lower impact and lower priority
- Can create a confusing experience to readers, for example if the UI or the available features don't match the current released version, or if the preview content interwoven into the existing content would create confusion
- Requires tracking so that we can remove preview text and caveats after the feature or service is no longer in preview

Where you might publish instead

When contributors want to cover preview scenarios, we recommend some or all of these options:

- **Option: Publish a blog post** Make a blog post with links or other info about how to join the preview. Blogs are useful when there is only a small amount of content to publish and it can fit in a single post (or maybe a series of a few posts). Also useful if the feature is simple to use and discoverable and therefore doesn't need advanced documentation. You might want to link from related content on docs.microsoft.com to the blog post about the preview. We can convert blogs to official docs later when the release code is more stable.
- **Option: Publish on visualstudio.microsoft.com (VSCOM)** If the goal is to share value prop + high-level information, not technical details, consider the product site. Also, good when the feature is simple & discoverable. See example.
- **Option: Create a PDF to send to preview participants** For a limited, private preview where the information about the product or feature is not public, send a pdf (or use option 4). Also useful if there are security requirements, for example, the content must be watermarked, must not allow copying, etc. We can convert this content to official docs later when the release code is more stable.
- **Option: Publish on a separate (private) GitHub repo** When the engineering team already has a GitHub repo for source code, and especially if the project is open source, the repo can be a good solution. Also useful if there are security requirements, for example, the content must be kept in a private repo. We can convert this content to official docs later when the release code is more stable.

Exceptions

In some cases, however, we do publish preview content on docs.microsoft.com.

Exception 1

Preview release notes from the release team are permitted.

Exception 2

Updates to the build numbers page are permitted.

Exception 3

If the content does not fit into any of the previous options and all the following are true, then publishing on docs.microsoft.com is permitted.

- The preview is public. (See What can I publish on docs.microsoft.com page and in the Preview content page in this guide)
- The content aligns with a high priority strategy for the product group
- The product team expects to have many users and many pageviews for this preview feature, or a small set of strategic users
- The product group must be willing to do the following with advisory input from the VS/VS Code writing team:
  1. Own the repo (if required) and the content
  2. Go through the docs onboarding training
  3. Write, publish, and maintain the content
  4. Update the content for every change and churn of the preview feature, such as screenshots for UI changes, code snippet updates for API churn, etc.
  5. Resolve GitHub issues on the content. While you can disable GitHub issues, we usually want feedback during a preview, so we don't recommend turning off the docs feedback control.

Formatting and structure for exceptions

If you publish have an exception and can publish preview content on docs.microsoft.com, you must clearly let the reader know that this content is for a feature or service that is not yet available to everyone.

Best practices include:

- Append "(Preview)" to the H1 or H2 for a small feature that doesn't require a page
- Note any requirements for having a preview version installed in prerequisites or procedures as needed
- Include text near the top of the page, above the fold (or near the H2 preview text if it's a small feature), to let users know the feature is a preview, or requires a Preview version. The text could be something like: *"This feature is currently in preview. Previews are made available to you on the condition that you agree to the supplemental terms of use. Some aspects of this feature may change prior to general availability (GA)."*
- You might want to add a blurb about the preview feature and content to the What's New or Release Notes pages, to generate energy/buzz about the preview. In some cases, VS Code has included info about a preview for several months in a row in the release notes to build excitement. The VS Code release notes then indicate how to get the preview feature, either in the Insiders release or in the "Stable" release but requiring a setting switch. In either the VS What's New page or the VS Code Release Notes page, work with the page owner to determine if the preview info is appropriate.
- For Visual Studio preview releases, refer to the preview release name in this format: "Visual Studio 2019

version 16.7 Preview 4 or later." Note: It's ok to shorten to "16.x Preview Y."

Contribute to Xamarin documentation

11/2/2020 • 2 minutes to read

Xamarin documentation is available publicly at:

- https://docs.microsoft.com/xamarin
- https://github.com/MicrosoftDocs/xamarin-docs

For staging and internal contributions, visit:

- https://review.docs.microsoft.com/xamarin?branch=master
- https://github.com/MicrosoftDocs/xamarin-docs-pr

If you have a question about contributing, email the Xamarin docs team alias.

Article metadata

The Xamarin docfx.json defines default values for most required metadata (based on directory structure), such as:

- `ms.prod: xamarin`
- `ms.technology: xamarin-forms | xamarin-android | xamarin-ios | xamarin-mac | xamarin.essentials | xamarin-skiasharp | xamarin-crossplatform`
- `ms.topic: conceptual`
- `searchScope: Xamarin | Xamarin.Android | Xamarin.Forms | Xamarin.iOS | Xamarin.Mac | SkiaSharp`

These defaults mean the minimum required metadata block at the top of new Markdown files should be:

```
---
title: "Xamarin page title"
description: "Page description is required"
author: github-alias
ms.author: ms-alias
ms.date: 01/31/2020
---
```

Don't forget to update author info for new or significantly changed pages. Similarly, the date should only be updated if the article is considered *completely up to date* (don't update the date for a minor typo fix). Date format is mm/dd/yyyy.

Pull requests

Xamarin repos have PRMerger installed. Review your changes carefully, and ask for a review if appropriate. Use PRMerger commands like `#sign-off` when your PR is complete and ready for publishing.

Screenshots

Ensure screenshots are clear, readable, and include accessible alt text. Trim extraneous information from a screenshot, and if required add a higher-resolution lightboxed version.

Example of a regular image (less than 800 pixels wide) with alt text:

`![Create a new project](images/win-2019.png)`

Lightbox example with a small (fewer than 800 pixels wide) image as well as a larger image to zoom in:

`[![Create a new project](images/win-2019-sml.png)](images/win-2019-lg.png#lightbox)`

Tabs usage

Tabs are used to let customers view information specific to one or more platforms they are targeting: **Android**, **iOS**, or **UWP**.

You can see an example on this Xamarin.Essentials page (source).

```
# [Android](#tab/android)
Android content
# [iOS](#tab/ios)
iOS content
# [UWP](#tab/uwp)
Windows content
--------------
```

Pivot usage

Pivots are used to let customers switch between **Visual Studio 2019** and **Visual Studio for Mac**. The pivot will autoselect based on the operating system the customer is browsing on.

You can see an example pivot on this quickstart page (source).

```
::: zone pivot="windows"
Windows content
::: zone-end
::: zone pivot="macos"
Mac content
::: zone-end
```

Pivots are defined in zone-pivot-groups.yml – this file should not be updated unless we ship new versions of Visual Studio.

Product versions

Xamarin docs target the current stable version.

There is no support for monikers that let customers choose a version.

Release notes

Xamarin release notes are in a separate repo:

- https://github.com/MicrosoftDocs/xamarin-engineering-docs-pr

API reference

Xamarin docs use the xref syntax extensively for build-time-checked reference links, for example, `[Effect](xref:Xamarin.Forms.Effect)`.

You can test an xref in any browser by changing the `uid` in this query: `https://xref.docs.microsoft.com/query?uid=Xamarin.Forms.Effect`.

Reference source

Xamarin reference content, which drives the docs site and IntelliSense, is stored in the following repos:

- Xamarin.Forms reference
- Xamarin.Android reference from Google
- Xamarin.iOS and Mac reference
- SkiaSharp reference

Samples

Xamarin code samples can be browsed on docs.microsoft.com/samples. The source code is available in the following repos:

- Xamarin.Forms samples
- Xamarin.Android samples
- Xamarin.iOS samples
- Xamarin.Mac samples

Meriwether diff reporting tool guide

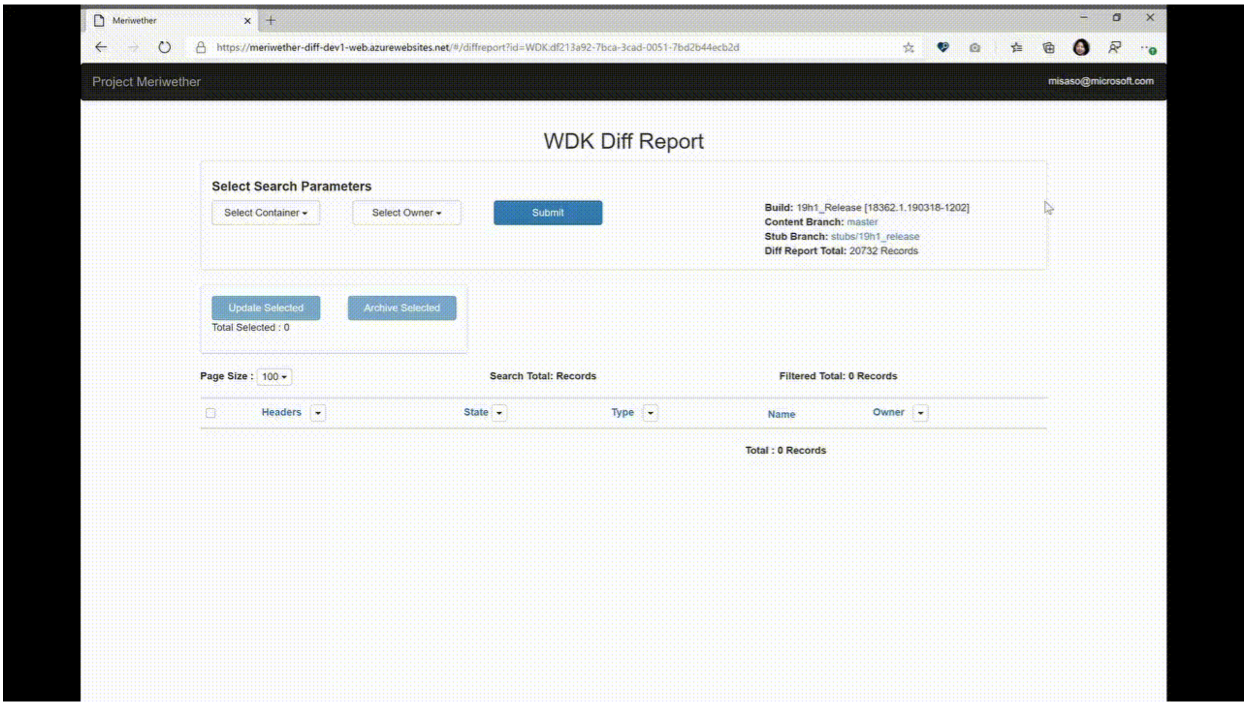
4/16/2021 • 3 minutes to read

The Meriwether diff reporting tool at https://aka.ms/meriwether is an internal tool that helps with Windows API content authoring. Using the tool, authors can:

- View APIs that are new, modified, or deleted, in a given release
- View APIs that are undocumented in previous releases
- View and select APIs by author, header file, or namespace
- Filter APIs by New, Changed, Delete, or Undocumented
- Put APIs in Hold state for further filtering
- Obtain raw markdown stubs for authoring

This tool supports the following API reference docs:

- *Windows Runtime*
- *Windows UI*
- *Win32 and COM APIs*
- *Windows Driver Kit DDIs*



Using the tool

No setup is required to use the diff reporting tool.

To use the tool, navigate to https://aka.ms/meriwether, and select the report you wish to work with. The title of the report indicates the name of the release, to pull diff information on. For example, WDK_Iron will pull up a diff report for the Iron release of Windows WDK DDI, and is a comparison of the stub branch of the Iron release, against the authoring branch of the Iron release. Note that the report does not diff against branches in different releases.

> **NOTE**
>
> The report will take a few seconds to load.

Each report contains a link to the content branch, a link to the stub branch, the version and build number, and two report totals. **Diff Report Total** represents the total changes, including New, Undocumented, and Changed APIs for the selected release. **Diff in Release** represents the total number of New APIs in the selected release, and not in the authoring branch for that release.

Under **Search Options**, select up to 10 headers from the **Select Container** drop-down text box, or select an alias from the **Select Owner** drop-down text box. To view API that are Undocumented in all previous releases, select the box next to **Include Undocumented**.

Once you've made your selection(s), choose **Submit**, and the report will display a list of APIs organized by headings: Headers, State, Type, Name, and Owner. You may filter these heading by clicking the drop-down arrow next to each heading. You can also order the APIs by a heading. API States are either New, Changed, Deleted, or *Undocumented* - if the **Include Undocumented** box was checked.

The following describes the API States in the report:

STATE	DESCRIPTION
New	APIs with unique IDs appear in the selected version stub branch, but NOT in the content authoring branch, and NOT in the previous version stub branch.
Undocumented	APIs with unique IDs appear in the stub branch of all versions prior to the selected version, but NOT in the content authoring branch, and NOT in the selected version's stub branch.
Changed	APIs with unique IDs appear in both the selected version's stub branch, and the authoring branch, but the keys differ.
Delete	Delete means the UID exists in the content branch but not in the stub branch.

Click each API name to view the API members.

The **Search Total** displays a total from the **Search Options** filter box, and the **Filtered Total** displays a total of the APIs filtered by the headings (Headers, State, Type, Name, Owner).

Get topic stubs

Topic stubs are autogenerated for each new/changed API. To get these stubs, select the checkbox next to the API, and choose **Update Selected**. A branch containing the stubs will be created for you in the content repo. To find your branch, navigate to the content repo to which you selected the report for, and on the left navigation, choose **Branches**, then choose **All**. The branch will be in the form of:

```
<alias>/diffrequest-on-<stub branch>-<yyyymmdd>-<hhmmss>
```

Note: the new branch won't be listed under **Mine** because it was created by the tool.

You can view the new or changed files by viewing the latest commit history of the new branch.

Mark as hold

You can use the **Toggle Hold** button to set state for selected APIs to Hold. This state can be used for APIs that you want to set aside for now and not have appear in the diff report.

Notes

- You can only select a maximum of 10 headers at a time.
- User state of the tool doesn't persist, so clicking the back button of the browser will navigate back to the start page.
- At this time, the branches created by the diff report can't be deleted.
- Only 2 reports are available for each API reference. For example, reports for the current release and the next release are the only ones available for WDK DDI.
- Under **Search Options**, a user may only select headers OR owners, but not both. Selections do not filter the other options.

Issues

File bugs or issues by using this bug template.

How to use Microsoft Docs Documentation

4/22/2021 • 6 minutes to read

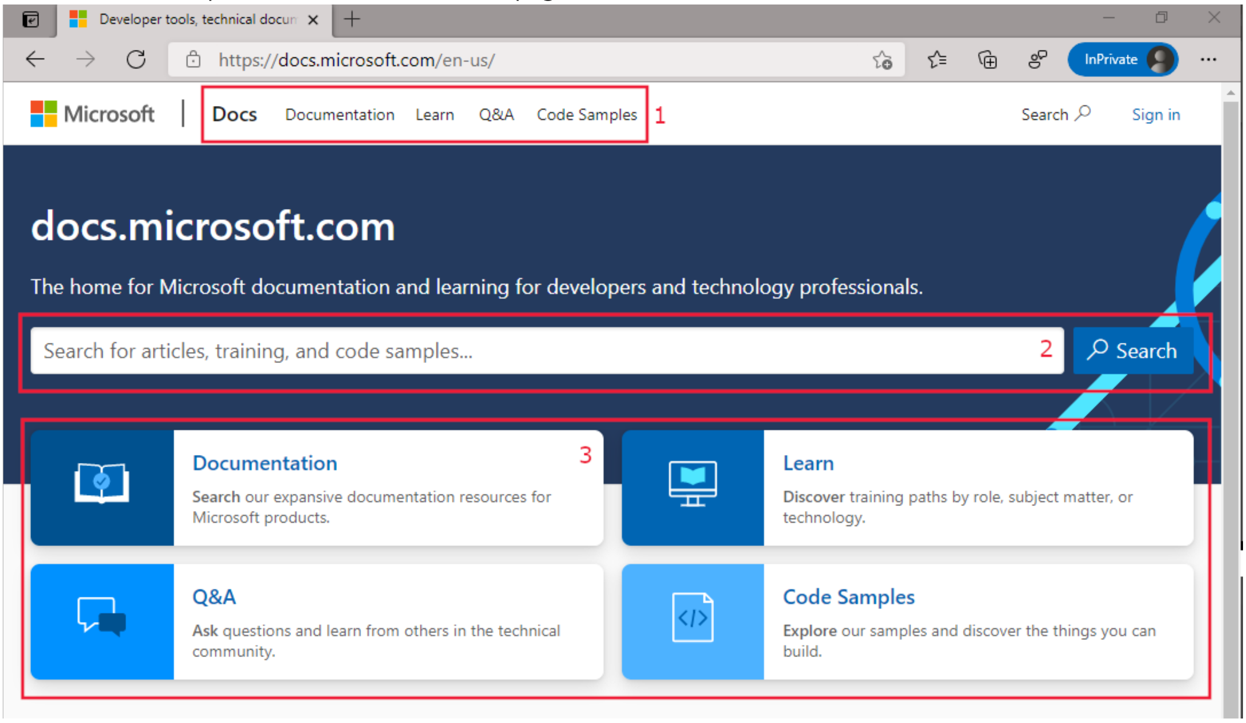
`docs.microsoft.com` (also known as the Microsoft Docs platform or DMC) is the home for Microsoft documentation and learning for developers and technology professionals. DMC launched in June 2016 as a replacement for the Microsoft Developer Network and TechNet libraries, which previously hosted this documentation. Over the years, DMC has expanded its charter and now includes four main services:

- **Documentation:** View conceptual articles, tutorials, guides, and references for our products and services.
- **Learn:** Discover training paths and certifications by role, subject matter, or technology.
- **Q&A:** Engage with others in the technical community.
- **Code Samples:** Explore our code samples to discover what you can build using Microsoft tools and technologies.

In this article, you'll learn how to navigate DMC's Documentation node. We'll also explore the common user interface shared by most Documentation articles.

DMC home page

Let's start at the top level with the DMC home page:



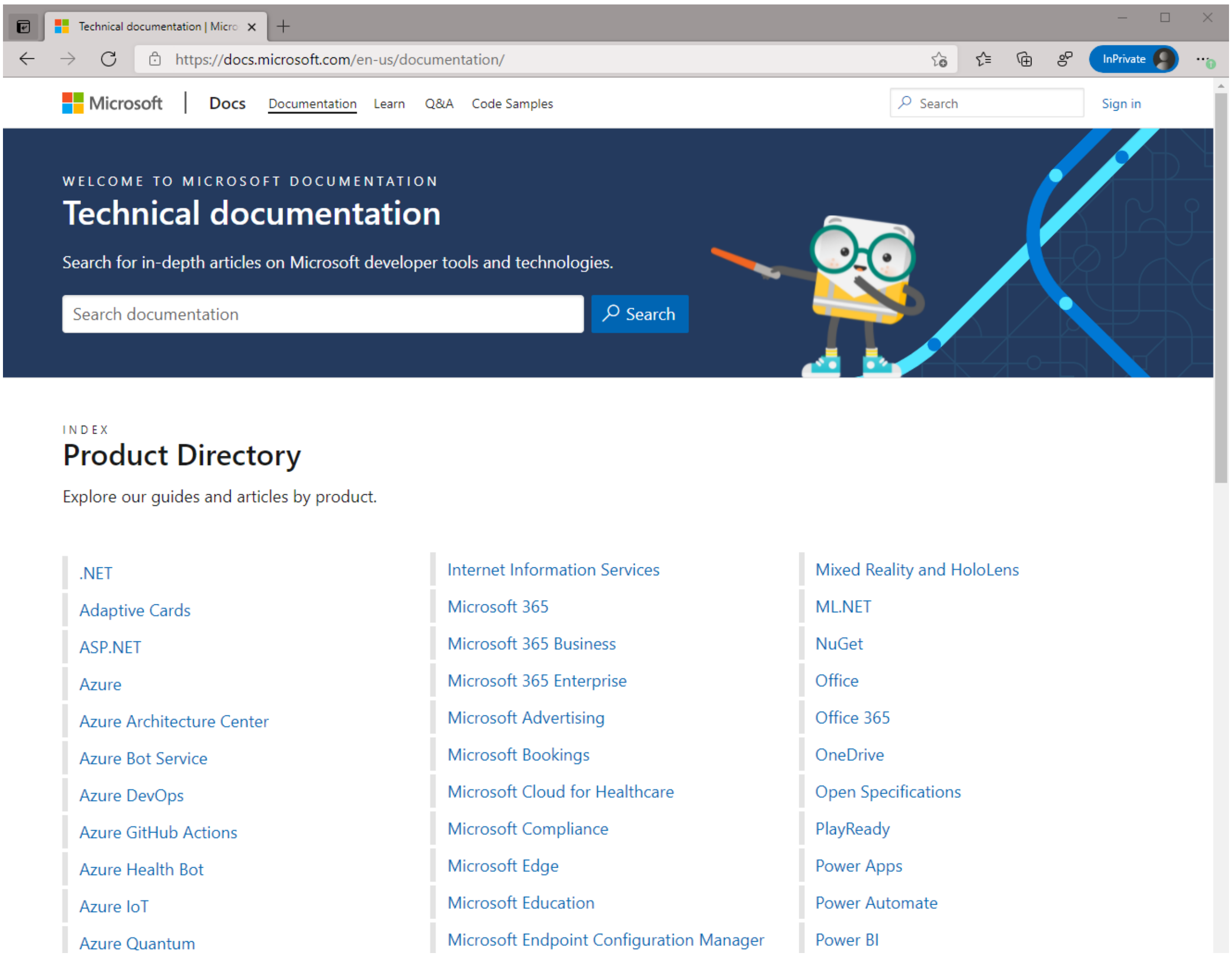
The DMC home page includes these elements:

1. **Global navigation:** All site content in DMC shares the global navigation (also referred to as Primary Nav). The global navigation provides access to the top-level subsites of Docs; to date, these subsites are Documentation, Learn, Q&A, and Code Samples.
2. **Search bar:** Use the search bar to search across all content in Documentation, Learn, Q&A, and Code Samples.
3. **Quick links to each DMC area:** If you know what you're looking for, jump to the content you need.

For the purposes of this article, we'll focus specifically on the **Documentation** area of DMC.

Documentation index

From the DMC home page, clicking **Documentation** in either the global navigation or the quick links brings you to the index for all Microsoft Technical Documentation:

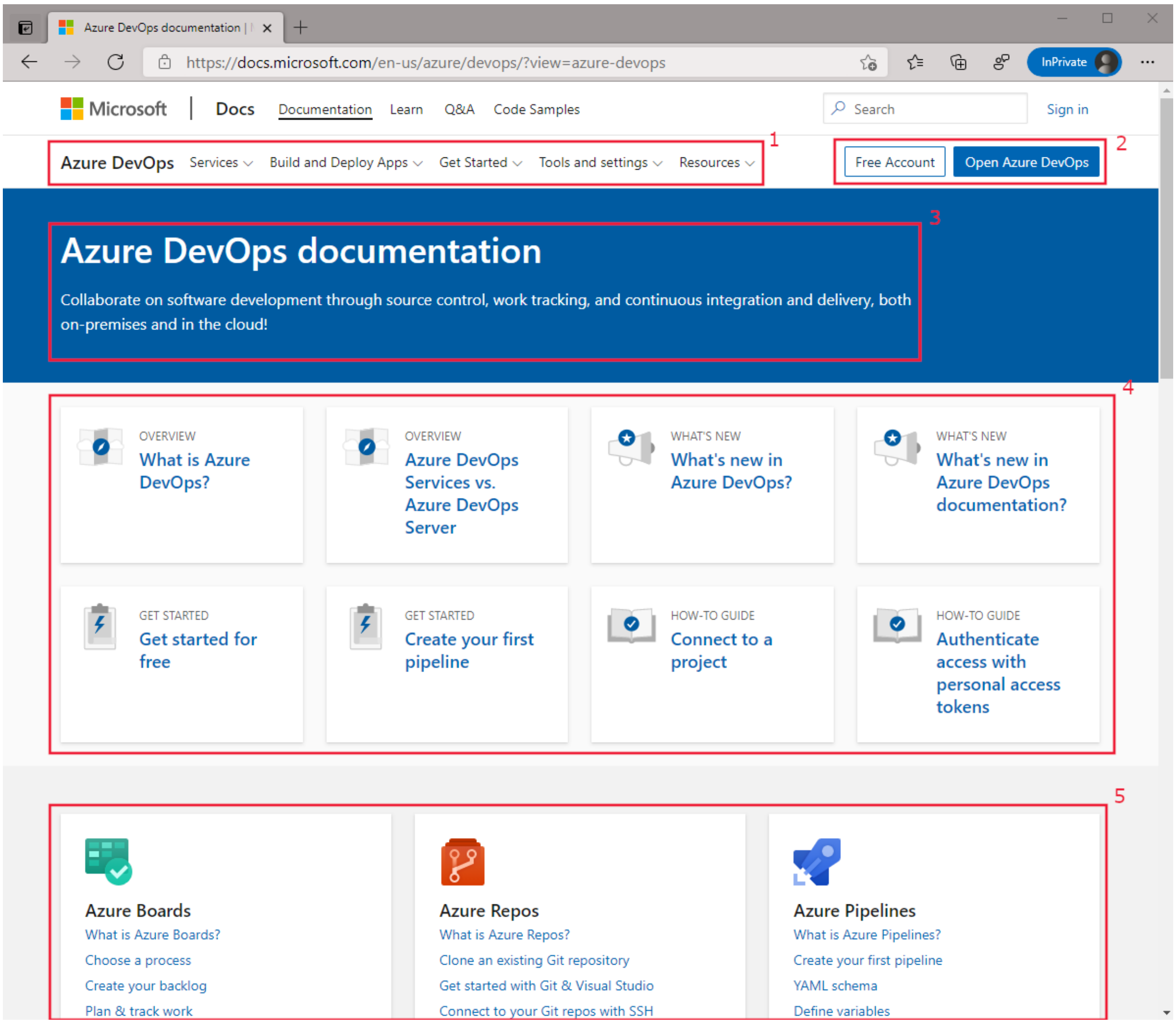


Notice how the global navigation remains and the **Documentation** link is now underlined. Use the search bar to search Documentation, or use the links in the Product Directory index to navigate to the documentation for a specific product.

Hub pages

Hub pages are collections of broader but related services, products, or languages. We use hub pages to help our customers better understand the bigger picture, see how components fit together, and quickly guide them to their area of interest.

From the Documentation index, select **Azure DevOps** from the product index to see an example hub page:



The Azure DevOps hub page includes these elements:

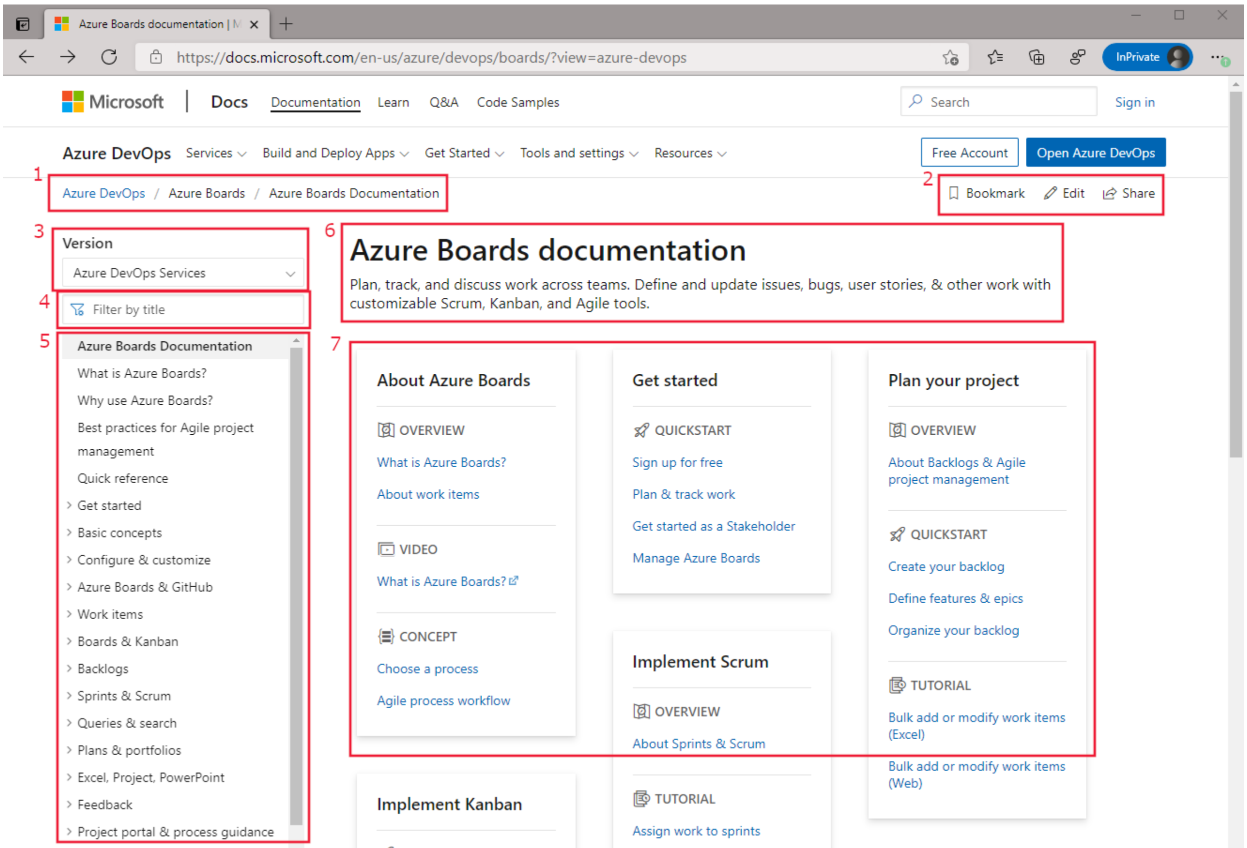
1. **Product anchor and product-level navigation**
   a. Most hub collections have a product anchor, which is usually the product name. If enabled, the anchor links to the top-level hub page.
   b. The product-level navigation helps users navigate product content. The Level 1 categories convey the breadth of documentation available for the product and should highlight meaningful areas of the product's documentation. Dropdowns in Level 1 provide links to Level 2 items.
2. **Action/Status bars**: Authors can add a blue primary button and a white secondary button to create a high-level action or indicate status. These buttons are part of the product-level navigation and persist across pages in the docset. In this example, the primary button opens Azure DevOps, and the secondary button helps the user create a free account.
3. **Root**: The root section contains the page title and summary for the hub collection.
4. **Highlighted content cards**: This section highlights content the authoring team thinks is most critical to its readers, like overviews, getting-started guides, how-tos, and content updates.
5. **Product directory cards**: The authoring team curates the directory to showcase product-related content.

Highlighted content and the product directory are optional sections and may not appear on every hub page. Other optional sections you may find include conceptual content cards, tools, and custom content.

Landing pages

Landing pages are an entry point for customers into a single docset. Usually, these pages map to a single table of contents (TOC) and appear at the top of the TOC. Authors use landing pages to surface top customer tasks or subjects for their product or service. Links on these pages point directly to articles within the docset. Landing pages can be a child of an overall hub page.

From the product navigation on the Azure DevOps hub page, select **Services --> Azure Boards** to see an example of a landing page:



Notice how the global and product-level navigation bars remain.

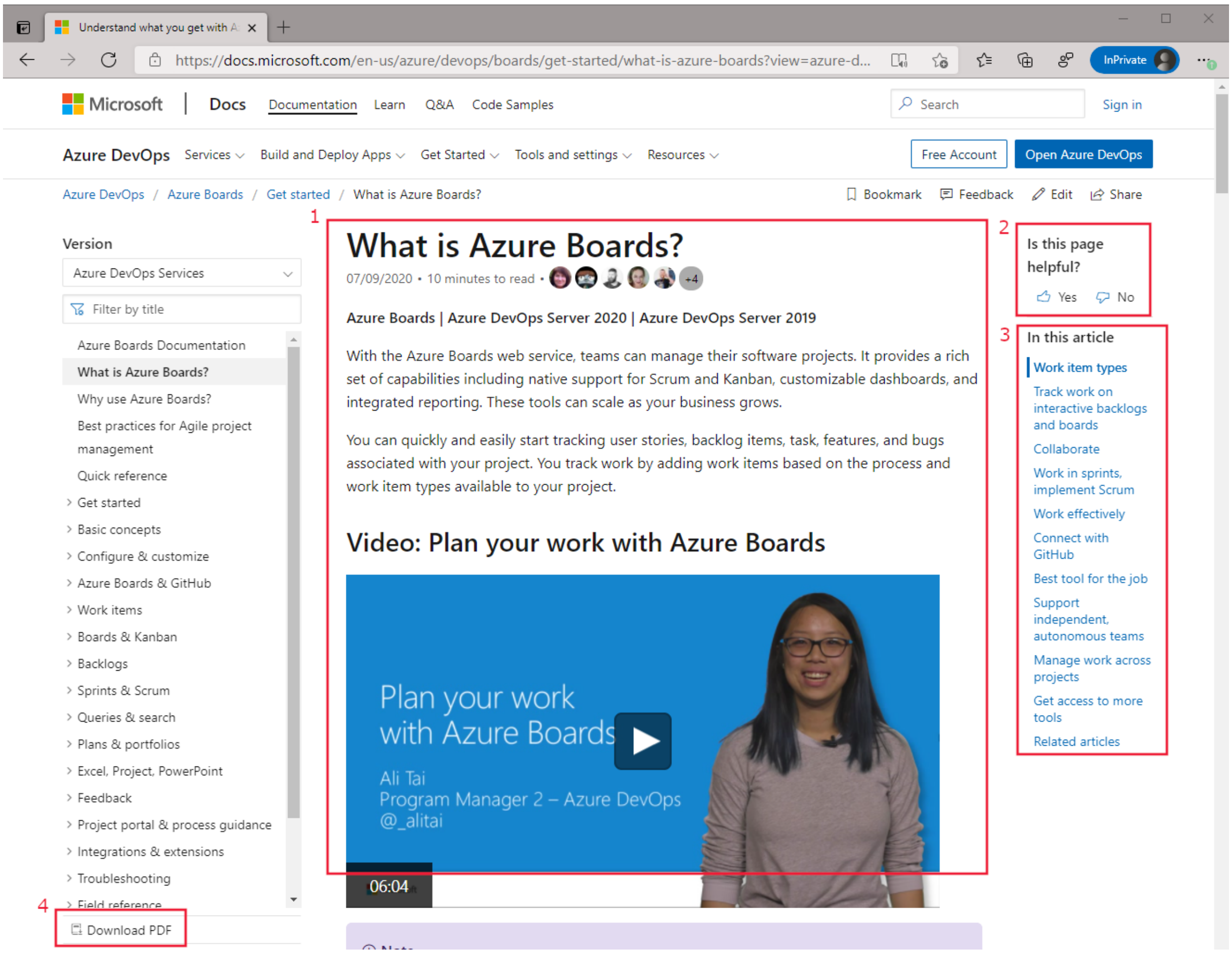
The Azure Boards landing page contains these elements:

1. **Breadcrumb**: Breadcrumbs are a series of links at the top of a page that convey the page's position in the overall site hierarchy. Breadcrumbs help set the context of the type of content you're reading, and they allow you to navigate back up the TOC tree. The page you're viewing appears on the far right, and each parent page appears to the left.
2. **Bookmark, contribute, and share**
   a. If you're logged in, bookmark a page to save it to your profile. Find all your bookmarks by clicking on your profile icon in the upper right-hand corner and selecting **Bookmarks**.
   b. If enabled by the authors, the **Edit** button allows anyone to contribute to the current page using GitHub for the web. Customers can use this contribution method to make small changes to an article.
   c. The **Share** button provides options to share an article via Twitter, LinkedIn, Facebook, or email.
3. **Version filter**: Some technical documentation covers content for more than one version of a product or service. Features can vary between versions, and so the content itself varies sometimes too. Use the version filter to select the appropriate version of the product you're viewing.
4. **TOC filter**: Search the titles within the TOC to find content faster.
5. **TOC**: Most services have one TOC with one landing page. For Azure documentation, the standard TOC nodes include Overview, Quickstarts, Tutorials, Samples, Concepts, How-to guides, Reference, and Resources.
6. **Title and summary**: Landing pages have a title and brief summary of the ideas covered by the docset.
7. **Top-task cards**: A landing page may have as many as 12 cards displaying the top customer tasks or top subjects, with supporting links.

Article pages

Articles make up the bulk of the pages within Documentation and cover top customer tasks. Articles can take the form of overviews, quickstarts, tutorials, concepts, how-to guides, FAQs, and more.

From the Azure Boards landing page, select **What is Azure Boards?** from the TOC to see an example of an article:



Notice how this page looks similar to the landing page. Landing pages and articles share many page elements, but there are a few elements that are unique to articles:

1. **Quality content:**
   a. All DMC content should align to our unified content model and writing principles. Documentation articles should include conceptual or procedural information required to understand and use the product. The goal of technical content is to show people *how* to do something, with a secondary focus on the *what* and *why*.
   b. Authors stamp each article with the date of the most recent major update to help you gauge freshness. You'll also find an estimated reading time and a list of contributors.
2. **Helpfulness feedback**: You can use the thumbs-up to give kudos to a helpful article, or you can use the thumbs-down to flag an unhelpful article. Selecting either will open a dialog box where you can provide optional comments before submitting your rating. Ratings and comments are private.
3. **Article navigation:** The right-hand navigation pane allows you to quickly navigate to sections within an article and identify your location.
4. **Offline documentation:** If enabled for the docset, you can download a PDF for offline reading. The PDF includes all articles within the entire TOC, not just the article you're viewing.

At the bottom of a Docs page, you'll find the Feedback control, if enabled:

Feedback

Submit and view feedback for

This product | This page

The Feedback control allows you to submit product or documentation feedback.

- **This product** is the product-feedback button and directs you to the online location where you can provide feedback directly to the appropriate product team. For many products, feedback is collected in a UserVoice forum.
- **This page** is the page-feedback button and requires you to sign in to your GitHub account to submit feedback. Once signed in, you complete a feedback form, which is converted into a GitHub issue in the public GitHub repository where the content is published.

Content and Learning office hours

3/5/2021 • 2 minutes to read

Office Hours provide a dedicated time and place to get help with subjects related to our work. The current schedule of office hours, and guidance for hosting your own office hours, is below.

Current office hours schedule

Information architecture

Hosted by Rachel Price Thursdays 12-2pm Pacific

- Teams Meeting

A regular forum for partner teams, writers, onboarding PMs, etc. to bring questions about information architecture on Docs. Example questions could be:

- Is this TOC too deep?
- Are we using this feature correctly?
- What's a better way to organize this landing page?
- I think we need a hub page; how do we plan what it should look like?
- Wait, what's the difference between a hub and landing page?

Instructional design

Hosted by Mark Taparauskas Tuesdays 11am-12pm Pacific

- Teams Meeting

Whether you're currently writing content for Learn or just have some questions about how it works, stop by and say hello! The Learn Instructional Design team will be available to answer your questions about how to write titles, learning objectives, knowledge checks, etc. Our authoring guidelines are here: https://aka.ms/learn/id

PowerShell-Docs Office Hours

Hosted by Sean Wheeler every other Thursday 11:00am-12:00pm Pacific Time

Join the PowerShell-Docs Office Hours Teams channel

If your content set contains PowerShell examples or cmdlet reference, we are here to help. Come get help with your PowerShell content. Get answers from PowerShell specialists, share learnings and best practices, and get help onboarding your PowerShell content.

Get started by reading the PowerShell Docs Contributor Guide - https://aka.ms/PSDocsContributor

Email your questions to psdocsteam@microsoft.com.

Technical content weekly

Hosted by Cory Fowler Mondays 1:15-2pm Pacific

- Calendar ics
- Teams Meeting

Rotating topics.

Reference docs

Hosted by Robert Outlaw every Tuesday 12:30-1:30pm Pacific

- Calendar ics
- Teams meeting

Want to host office hours?

Anyone can host an office hours, and no permission is required to do so. However, we do ask that you follow these best practices.

Best practices

Always

- Always post your office hours in the Office Hours Schedule (above) and keep the information up to date.
- Always provide an ics link for the meeting in the Office Hours Schedule so people can add to their calendars as desired.
- Always provide a link to the Teams meeting in the Office Hours Schedule.
- Always record the meeting.
- Always send out brief notes after the session, with a link to the recording.

Consider

- Consider creating a page in the Contributor's Guide to describe the purpose in more detail and to provide links to related resources and information.
- Consider a specific agenda to avoid impractical discussions.
- Consider scheduling a conference room for in-person attendance.
- Consider scheduling at times where people in other time zones can attend. Earlier in the West Coast day is better.
- Consider a bi-weekly cadence. However, a weekly or monthly may be more appropriate.
- Consider running an office hour only for a few months as a pilot before establishing it for the long term. Consider ending an office hours series if fewer than 5 people are attending each session.

Content & Learning Vendor Supported Quality Programs

5/10/2021 • 2 minutes to read

Aquent and Acrolinx continue to provide quality-related support to the Content & Learning organization. The statement of work for managed editorial and authoring services is funded and scoped for the following work:

Editorial automation

Acrolinx provides instant automated feedback to authors about spelling, grammar, punctuation, terminology, and style. It runs automatically in every pull request and provides a link to results. You can also run Acrolinx locally in Visual Studio Code. Where Acrolinx is running, a minimum Acrolinx score of 80 is required to merge a pull request. Acrolinx is our first line of defense for baseline editorial quality.

- Acrolinx overview

Core editorial services

You can request a copy edit or developmental edit on technical documentation after you publish it. The vendor team also refers content for copy edit from the pull request queue if the content appears to need a copy edit. For copy edit, the editors review and provide feedback on spelling, grammar, punctuation, and clarity. They also review for basic Microsoft Style Guide items and product-specific terminology. For developmental edits, the team reviews the content for alignment to the C+L content model, to a clear customer intent, and to the the writing principles. Developmental edits include copy edits.

- Edit service description and instructions for requesting an edit
- Dev edit checklist for tutorials
- Editing Azure tech docs - resource for vendor editors

Pull request review

The resources on the managed service team review pull requests, provide authors with feedback, and merge pull requests that meet the defined quality criteria. This process applies to most Content & Learning conceptual content repositories, including the Architecture Center and the Learn repos.

The reviewers monitor pull requests across more than 20 repositories.

SLA

For pull requests that require human review, the target SLA is to pick up 50% of pull requests for review within 2 business hours of sign-off and the remaining 50% of pull requests within 4 business hours of sign-off. There is no SLA on how long it will take to review content or get it published after the author signs off because of variability in the volume of changes in pull requests, the amount of feedback needed, and the varied quality issues pull requests might have.

Resources

The following articles are related to this process:

- Pull request review and publishing schedules
- Criteria for merging
- Content release planning + process

- Content channel guidance
- Pull request submission best practices
- Public repo pull request review
- Query aggregation tool for supported repositories
- Pull request review process (private)
- Pull request review process (public)
- Aquent team wiki for PR review

Content authoring

A pool of technical writers is available for general content maintenance tasks such as freshness reviews, updates, and broken link fixing. The program functions under a fixed per-month budget. The team is focused on updating existing content that is low and medium priority and that has no fixed due date. These parameters ensure that the service can operate independently as a managed service. Items in this workflow are subject to prioritization by the Content & Learning leadership team and depend on organizational priorities, resourcing, and budget.

- Detailed content authoring program description
- Vendor team skillsets

Managed service contacts

Authoring, editing, and pull request review:

- Justin Chappell – Microsoft project owner
- Tami Fosmark – Aquent project manager

Acrolinx:

- Monica Rush – Microsoft project owner

Tutorial review checklist

11/2/2020 • 5 minutes to read

The content model for docs.microsoft.com was created to support quality and consistency. The model provides the standards for several content types that work together to support a complete customer experience with our products and services. Tutorials are a part of that customer experience.

Tutorials are 100 – 200 level articles for an audience new to the service, product, or scenario. They're created from a list of top user tasks and provide the single best procedure for completing that task. Make sure you're familiar with the detailed guidance for writing tutorials before you continue with your review.

Clear cut criteria

CATEGORY	CRITERIA	ACTION	GUIDANCE
TOC	The tutorial is listed in the Tutorials node of the TOC.	Verify	Write tutorials - TOC
TOC	The article title in the TOC doesn't wrap to a second line when viewed in a full desktop browser screen.	Verify	Write tutorials - TOC
Customer intent	The customer intent statement is a markdown comment in the last line in the metadata section.	Verify	Write tutorials - Customer intent
Metadata	The ms.topic attribute has a value of "tutorial".	Verify	Write tutorials - Metadata
Metadata	The title attribute includes word "tutorial".	Verify	Write tutorials - Metadata
Metadata	The description attribute includes the word "tutorial".	Verify	Write tutorials - Metadata
H1	The H1 starts with "Tutorial: <verb>...".	Verify	Write tutorials - H1
Introduction	The introductory information is no more than 5 or 6 sentences.	Verify	Write tutorials - Introduction
Introduction	The introductory paragraph is followed by a sentence that says, "In this tutorial you'll do X..."		

CATEGORY	CRITERIA	ACTION	GUIDANCE
Introduction	The "In this tutorial you'll do X..." sentence is followed by a list of what is covered in the article using the green checkmark format.	Verify	Write tutorials - Introduction
Free account	There's a sentence that provides a link to a free account just before the first H2. For example, "If you don't have an Azure subscription, create a free account before you begin."	Verify	Write tutorials - Free account
Content	The content is referred to as a tutorial, not as a "guide", "article", or "topic".		
Screenshots	If there are screenshots of the portal, the first screenshot includes the full browser (Google Chrome or Safari (for Mac)).	Verify	Write tutorials
Screenshots	Screenshots aren't of an internal version of the product/service. Tip-off: Screenshots don't include these indicators in the banner at the top of the portal window:  	Verify	Screenshots
Prerequisites	If the article has prerequisites, **Prerequisites** is the first H2 in the article.	Verify	Write tutorials - Prerequisites
Links	There are no links to outside content after the **Prerequisites** section and before the **Next steps** section.	Verify	Write tutorials - Linking
H2s	The H2s don't wrap in the right-hand nav.	Verify	Write tutorials - H2s
H2s and H3s	The H2 and H3 headings aren't numbered.	Verify	Write tutorials - H2s
H2s and H3s	The H2s and H3s start with a verb.	Verify	Write tutorials - H2s

CATEGORY	CRITERIA	ACTION	GUIDANCE
Clean up resources	The article includes a **Clean up resources** H2 that immediately precedes **Next steps**. If there are no resources to clean up, the section should state that.	Verify	Write tutorials - Clean up resources
Next steps	The article includes a **Next steps** H2 section.	Verify	Write tutorials - Next steps
Next steps	There's a single link in the **Next steps** section that uses the "blue box" format.	Verify	Write tutorials - Next steps

Criteria that requires a judgment call

CATEGORY	CRITERIA	ACTION	GUIDANCE
Customer intent	The customer intent statement follows the *As a < type of user >, I want < what? > so that < why? >* format. Tutorials should be scoped to a single customer intent. Review the customer intent statement and ask yourself if it represents a task that a customer would likely need an answer to. For example, it's unlikely that a customer would need a series of unrelated management tasks as an answer to a single problem.	Verify	Write tutorials - Customer intent
Content type	The tutorial documents only one way to complete the task defined by the customer intent statement. Tip-offs: Optional steps, various ways to do the same task.	Verify	Writing principles checklist
Content type	The tutorial creates an example or proof of concept environment. Tip-offs: fictitious names, specific values for inputs, sample environments.		

CATEGORY	CRITERIA	ACTION	GUIDANCE
Content type	The bulk of the article is procedural steps. Conceptual content exists only to help the customer better understand how or why they would complete a step.	Verify	Choose a content type
Introduction	The introductory paragraph explains the customer intent and what the customer will accomplish in the article.	Verify	Write tutorials - Introduction
CLI	If the article uses the CLI, and if all commands run in Cloud Shell, the introduction includes the cloud-shell-try-it.md include file.	Verify	Write tutorials - Includes
PowerShell	If the article uses PowerShell, and if all commands run in Cloud Shell, the introduction includes the cloud-shell-powershell include file.	Verify	Write tutorials - Includes
CLI/PowerShell	If the article uses the CLI or PowerShell and all the commands don't run in Cloud Shell, the introduction includes instructions for the user to install the CLI or PowerShell locally.	Verify	Write tutorials - Includes
Screenshots	Screenshots offer value for the customer (explain complex UI, provide context, validate they've taken the correct steps or are in the right place, etc.).		
H2s	The H2s align with, but don'tnecessarily mimic the green checkmarks in the introduction.	Verify	Write tutorials - H2s
H2s	Each H2 is followed by one or two transitional sentences explaining why the steps need to be performed or how they contribute to the whole.		

CATEGORY	CRITERIA	ACTION	GUIDANCE
Alerts	If alerts are used, they're used sparingly. No more than two per article, and multiple notes should never be next to each other.		
Code blocks	Code blocks include language indicators.		
Body	CLI or PowerShell focused articles do not include numbers/bullets for the steps.	Verify	Write tutorials - Bullets
Everyday words	The article includes keywords that would the customer search for that would take them directly to this article.		
Everyday words	The article doesn't sound too formal.		
Everyday words	The article doesn't include unnecessary Microsoft jargon.		
Write concisely	The article has documented only what is necessary to achieve the customer intent.		
Write concisely	The article has no unnecessary information getting in the way.		
Write concisely	The article uses locators (>) for "click, click, click" instructions.		
Easy to scan	The article is not too long to allow for scanning.		
Easy to scan	The most important information is first in the article.		
Easy to scan	The section headers lead the customer through the article.		
Easy to scan	The article use tables and bolding to organize information or break up long blocks of text.		

CATEGORY	CRITERIA	ACTION	GUIDANCE
Easy to scan	Articles that focus on the CLI or PowerShell don'tinclude numbers or bullets.		
Empathy	The article reads like we're a partner to the customer.		

Vendor pull request review: public repos

5/10/2021 • 10 minutes to read

This article documents the responsibilities agreed to by Aquent and Content & Learning for the vendor team that reviews public pull requests in Content & Learning repos.

- Check each assigned public repository at least twice daily to review pull requests that have the `ready-to-merge` label assigned and merge them.
- Facilitate and unblock the workflow when people have minor issues with process, GitHub, or Git.
- Escalate issues and questions to the appropriate FTE through the Aquent team lead.

General process for reviewing PRs

1. Run the Public repos query for Aquent PR reviewers query to find all the signed off pull requests for the public repos. Run the query periodically throughout the day. The first round of reviews needs to be complete by 9:45 AM each day.
2. Check for email sent to azdocspr by a peer, colleague, lead, or manager who is signing off on a public PR for the listed author.
3. Review each signed off pull request from steps 1 and 2:
   - Review the PR to verify the changes are updates or corrections of a technical, editorial, or formatting nature.
   - The changes should align to the normal criteria used for PRs in the private repo.
   - If you see obvious editorial issues in a PR, fix them using the GitHub UI. Make sure to fix obvious problems with spelling, grammar, punctuation, and product names. Remove locales (en-us, en-gb) from Microsoft website links where the locale isn't needed.
   - Verify link fixes.
   - When you complete the review and the PR is ready to merge, add a brief but specific comment. Then, merge each pull request.
4. Run the Public repos query - all open PRs to see all open pull requests in the public repos. This shows pull requests where an employee intended to sign off but did not follow the defined process. Any PRs that are returned need to be carefully evaluated to determine intent and whether the person who attempted to sign off is qualified to do so (they must be a trusted source from the content or core PM team).
5. For new pull requests since the last time the team checked, review the pull request titles looking for indicators of minor editorial changes. For these sorts of pull requests, verify that the change is indeed trivial or purely editorial in nature, and merge. If a change seems truly editorial in nature and the affected text is a programming element or code block, the reviewer can merge the PR.
6. Check for the **review-team-triage** tag on any pull requests - these pull requests need to be manually evaluated and manually assigned.
   - If a pull request appears to affect many unrelated files, contact the repository admin and ask them for help. For the Azure repo, contact the PR review program PM.
   - Includes do not always contain metadata. If a PR is filed against an include file, search the repo for articles that use the include file. Assign the PR to the author of the articles that use the include file most frequently.

- If a PR is filed for images only, look up the article that uses the image. Assign the PR to the author listed for that article.

7. Check for sign-offs that don't follow the process.

8. Check for pull requests where the contributor license agreement (CLA) has not been completed. In the list of open PRs, the PR is marked with a yellow dot instead of a green check mark. If the CLA has been pending for more than 15 days, close the PR with the comment "Closing pending CLA completion."

Configuration file modifications in public repos

Public contributions should typically be against technical content in the repository. Any pull requests that update a configuration file need to be raised up to the repo admin and Cory Fowler. In general, we should not accept such changes from the public repos. Config files are usually identifiable because most include a period as the first element in the file name:

.gitignore .gitattributes .openpublishing.build.ps1 docfx.json

Specifically, we do not want to merge .gitattributes changes to manage line endings - this is work that needs to come from our internal engineering team. For more info, see https://github.com/MicrosoftDocs/SCCMdocs/pull/1732.

Understand sign-offs

There are two ways people can sign off on public pull requests:

- The author listed in the article metadata can type #sign-off in the pull request to assign the ready-to-merge label.
- An internal Microsoft colleague can send an email to the techdocprs alias to sign off for the author listed in the article metadata. This person can be a peer PM or writer, or it can be a content team lead or manager. These people are considered "trusted sources". Sending an email to request a merge is the exception process for cases where the listed author isn't available to sign off directly in the PR.

The sign-off automation is rigid because it's looking only for the author to sign off. If an author signs off the wrong way, merge the PR and then point them nicely to the correct process. If a peer PM or a content developer in Erin’s org signs off, merge. Ideally, they send mail, but we accept sign off from these folks even though the automation doesn’t. These process issues mean you need to query the repo for “sign-off”, “approved”, or other known variants to see if a trusted source signed off. Also query for “please-close” in case the automation isn't working.

Additionally, authors may list their GitHub account incorrectly in the article metadata. An incorrect GitHub account in the metada also breaks the automation.

Figuring out who is a trusted source for sign-off

- Anyone listed in the existing article metadata in the author or ms.author value is a trusted source.
- If a user is in Erin Rifkin's organization, they're considered a trusted source.
- Program managers on core product teams are trusted sources - usually, they are in C+AI. Cognitive Services PMs are in the Research org. Check the address book to see if their address book entry mentions the product area they work in. If the area they work in is "storage" and their address book title mentions "storage", consider them trusted. Check the address book or Dr. Whom to see if the PM who is signing off is a peer of the PM listed as the author. There are tools to help you determine identities and relationships:
    - Chrome extension for GitHub identities: https://repos.opensource.microsoft.com/settings/security/tokens/extension

- Dr. Whom: https://who
- If you are unsure, ask the Aquent team lead.

Merge conflict policy

- When a writer signs off on a pull request, but the PR is blocked by a merge conflict, the PR reviewer resolves the merge conflict by using the GitHub UI. The reviewer resolves the conflict by taking the public contributor's changes. In each conflict, the text above the divider is from the contributor's branch, and the text below the divider is what is in master. This may mean some changes from private master are lost or regressed. To avoid this problem, authors should pull the changes into the private repo and resolve the conflict to their satisfaction. Instructions are in Resolve merge conflicts article in this contributor guide.

- If a merge conflict occurs for the content of the entire article, it's usually because an author moved or deleted the article after the public PR was filed on the article. The reviewer can verify whether the article exists by searching the repo for the file name. (Tip - while you are at the base URL of a repo, press T on your keyboard to open the file filter for the repo. Type the file name to search for it in the repo.) If the file has been deleted or moved to a different folder in the repository, the pull request has to be closed. This typically happens only when a pull request sits open for too long in the PR queue. When you close a PR like this, make sure to explain why you closed it.

> **IMPORTANT**
> The GitHub merge conflict tool in the UI creates a commit back from the target branch to the source branch. Under certain conditions, it can cause an accidental content release into master from a release branch. **Never use it in the private repository** - use the command line to pull the merge conflict to your local clone and resolve.

Formatting changes

Pull requests that modify formatting elements (tables, code blocks, line end spaces) have the potential to render incorrectly. PR reviewers should pull these sorts of changes to the private repository for build validation and staging. Instructions are in the How to work with public pull requests article.

New articles and significant rewrites in a public repo

When someone submits a new article or a significant rewrite in a public repository, verify whether the contributor is internal or external.

- **Internal (Microsoft employee or vendor)**: Close the pull request and direct the internal contributor to work in the corresponding private repository. There are two resources they can use:
  - Public vs private repos
  - Fork and clone a repository
- **External (community)**: Contact the business approver for the service or product to let them know that a new article or significant rewrite from the community needs to be helped through the process. Point them to the process documentation in the How to work with public pull requests article.

Monitoring and unblocking the workflow

There are many problems in the public repos that can cause changes to get stuck in the process. Other problems can uneccesarily slowdown the process. Remember the job is to help authors and contributors complete their work and get approved PRs merged. The reviewers need to help unblock the PR workflow with a minimum of redundant communication and delays.

- Let the PR review program PM know if you see many PRs from a single user.

- If you see PRs in the public repo that update the ms.date attribute and nothing else, refer them to the business approver for the service or product area.
- Escalate anything unusual, unexpected, or concerning to the program PM.

Involving the FTE business approver

For the following kinds of issues in public pull requests, the PR reviewers must contact the business approver and the vendor program PM:

- You have questions about whether someone is a trusted source.
- Someone wants to change the listed author. These changes should be made in the private repo.
- New articles or significant rewrites. If the contributor is external to Microsoft, the business approver needs to pull in the appropriate writer to work with the external contributor to facilitate the contribution process. If the contributor is internal to Microsoft, the business approver needs to work with the contributor to help them work in the private repo.
- Any PR that contains more than five changed files. Customers usually submit changes article by article. Employees working on deeper content updates tend to modify many files. This level of change indicates that the work should be happening in the private repo.
- Any PR where the situation is getting convoluted or the customer is becoming agitated.
- Any PR where the comments indicate a trusted source approved the PR, but the conversation isn't driving the PR into the standard workflow.
- Any PR with conflicting guidance to hold or merge.
- Anything out of the ordinary.

How do the public repo changes get into the private repo?

The overall publishing process allows changes from the private and public repos to be merged to each other. The publishing system has a synchronization feature that is enabled for some repositories to help automate some of the syncing. The result of the publishing process is that changes in the public repo go live to docs.microsoft.com along with the publish of the changes in the private repo. The syncing and publishing work like this:

- When a public PR is merged, a new pull request is automatically created from the public repo to the private repo. This sync PR accumulates commits as PRs are merged in the public repo.
- As the first part of a publishing run, Pubdesk merges this sync PR in the private repo to the default branch in the private repo.
- Pubdesk then merges the default branch in the private repo to the live branch in the private repo.
- Within about 10 minutes of the merge to the live branch, an automatic commit is made from that live branch (in the private repo) to the default branch in the public repo. This is a **direct commit**, so there's no pull request created. This commit brings the two repos even with each other ("sync'd") until the next PR is merged in either repo.

This cycle restarts with the next merged commit in the public repo.

Aquent technical content authoring managed service

5/13/2021 • 7 minutes to read

Scope of service

Aquent Studios provides content authoring services to the Content & Learning team as a managed service. These services are a Content & Learning-funded service that helps extend the capabilities of Content & Learning content developers. It's scoped to articles and tasks that the content team owns and manages. This service isn't available to product teams for self-managed content. If you need more information about resourcing options for your content projects, contact your content publishing manager or Justin Chappell.

Content & Learning content teams can use the Aquent services for the following types of content updates and other knowledge-based tasks:

- Full freshness reviews and updates
- Testing procedures against the product and updating the corresponding content
- Adding new sections to existing articles based on bugs
- Converting content to new article templates/update to align to the content model
- Updating or editing existing screenshots, taking new screenshots
- Updating existing code samples
- Changing product or service names when more than a simple search and replace is needed
- SEO-driven updates to improve discoverability
- Documentation refactoring
- Content archival or deletion
- Metadata updates
- Updates to address geopol issues flagged by PoliCheck
- Localization fixes

The vendor authoring team focuses on:

- Normal and medium priority work
- Work that doesn't have a specific due date

The team schedules and completes work in the order in which it's received. The following work is out of scope:

- High priority, high-visibility work
- Work that is required by a specific deadline
- Content owned by program management
- Creating brand-new content

How to request the vendor team take on your project

Follow these guidelines when you create a user story in Azure Boards:

- Use this authoring request template to create your user story.
- In the title, specify "Content update" or "New content", followed by a description of the work.
- Provide the repository where the content resides, and list the files or URLs for any existing pages, as well as a page view report, as this will help us prioritize limited resources. For new articles, provide a filename list and

destination folder.
- Include scheduling constraints and the requested completion date.
- Determine a priority, and the business reason behind the priority.
- Scope your user story to list concrete deliverables, such as:
  - Create 10 articles for public preview for service X, due by date Y.
  - Make the fixes requested in content bugs X, Y, and Z filed against existing content.
  - Update product name X in entire content set for service Y, including when the context requires revisions due to this name change.
- Submit only work that requires core technical writing skills. Production-type work goes to the production vendor.
- List the FTE content SME who should review the final work to determine whether it meets the content team's expectations. This SME could be a manager or individual contributor, and doesn't have to be the person who created the user story.
- If you have a large work request, batch it into multiple requests, targeting a maximum of 10 URLs per request.

If you have an existing task or user story, Aquent can streamline this process by creating a new user story and linking to your tracking item. You're welcome to set up a call to talk through these details and Aquent will create the user story for you. In these cases, contact Tami Fosmark.

User story review and assignment

Once a user story is received, the Aquent project manager will assess the work request:

- Schedule a conference call to talk through the details of the user story. This step helps to uncover assumptions and agree on scope, and will be recorded as a reference for writers when work is ready to begin.
- Provide a rough estimate for when the work is likely to start. Due to queue volumes and recurring tasks (like high-priority validation fixes and build issues), and the first-in-first-out model, committing to specific dates is not feasible.
- If the request exceeds the capacity of the managed service team, work with the Microsoft program manager to determine the prioritization of the request.
- Once there are one or more resources available, Aquent starts work and monitors to ensure the work is completed as defined.

Microsoft and Aquent share responsibility of keeping team at capacity.

- Aquent provides weekly reporting on the queue, work status, and on-time completion.
- Microsoft ensures that adequate work is in the queue.
- Both share responsibility for negotiating competing project deadlines.

Rapid response

The Aquent team responds to issues in the daily builds for azure-docs-pr, such as merge conflicts, validation issues, and other warnings. When these issues block builds, the publishing team sends email to pr-fix@microsoft.com. The Aquent team acknowledges receipt and addresses the issues as quickly as possible, within 90 minutes of notification.

Ongoing quality tasks

The Aquent team handles some quality-driven tasks operationally, as ongoing work. The cadences for these ongoing quality tasks are specified per repo.

SLAs and scope

- One business day SLA to acknowledge a new user story.
- Aquent will notify listed authors when work is starting, and when content changes are ready for review.
- If the scope changes, the Aquent project manager and the requestor will determine whether to accept the changes or submit the work as a new user story.
- QA for Aquent ensures that the user story was completed within the scope outlined. Aquent is not providing deep technical review of the content.
- Listed authors and SMEs must review changes within five business days of notification; before the end of that period, authors and SMEs can provide an alternative date by which they will review the changes (within 10 business days). For work where there is no response from a listed author, or where the response is not within the agreed time window, the author's manager will be contacted for signoff within 4 business days. If there is no response from the listed author or the author's manager after the designated time, the Aquent author will sign off on the pull request to have the change merged.

Prioritization

Content teams should plan work early to get into the authoring queue. At times, prioritization may allow specific work to “leapfrog” the first-in-first-out queue. If in-progress work is affected by a prioritization change, or the impact to other work in the queue is large, Aquent and the Microsoft PM for the managed service will triage and prioritize work in the queue in consultation with the affected teams.

In general, Aquent will prioritize the queue in the following manner:

- P1 - Issues triage/response, merge conflict resolution for publishing are always committed work.
- P2 - Broad content changes caused by a major product update (ex: Azure portal UI), branding and terminology changes that affect large content sets.
- P3 - Freshness updates and all other content maintenance. Items that are P3 will be set aside for higher-priority work and could be set aside indefinitely, depending on the queue above.

Approval of completed work

If there is a technical SME or someone on the content team who needs to review the final work to ensure it meets the content team's expectations, the requestor must specify that FTE for technical review signoff in the user story description of work.

- For individual pull requests, the listed ms.author will indicate approval in a GitHub comment in the pull request.
- For user stories that require multiple pull requests, the vendor team may request a release branch in which to do the work. When the vendor team has completed the work in the release branch, they will create a pull request to master to highlight the changes for review. The FTE approver will indicate approval in a GitHub comment in the pull request. Then, the vendor team will follow the standard release process to get the branch merged and published.
- The FTE approver, SME, and ms.author can provide feedback on what needs to be fixed before approval. Teams should expect to provide some basic feedback. However, if there are significant or recurring issues or problems, the FTE approver should document the issues in the pull request and communicate the issue via email to Justin Chappell and Tami Fosmark.

Rules of engagement

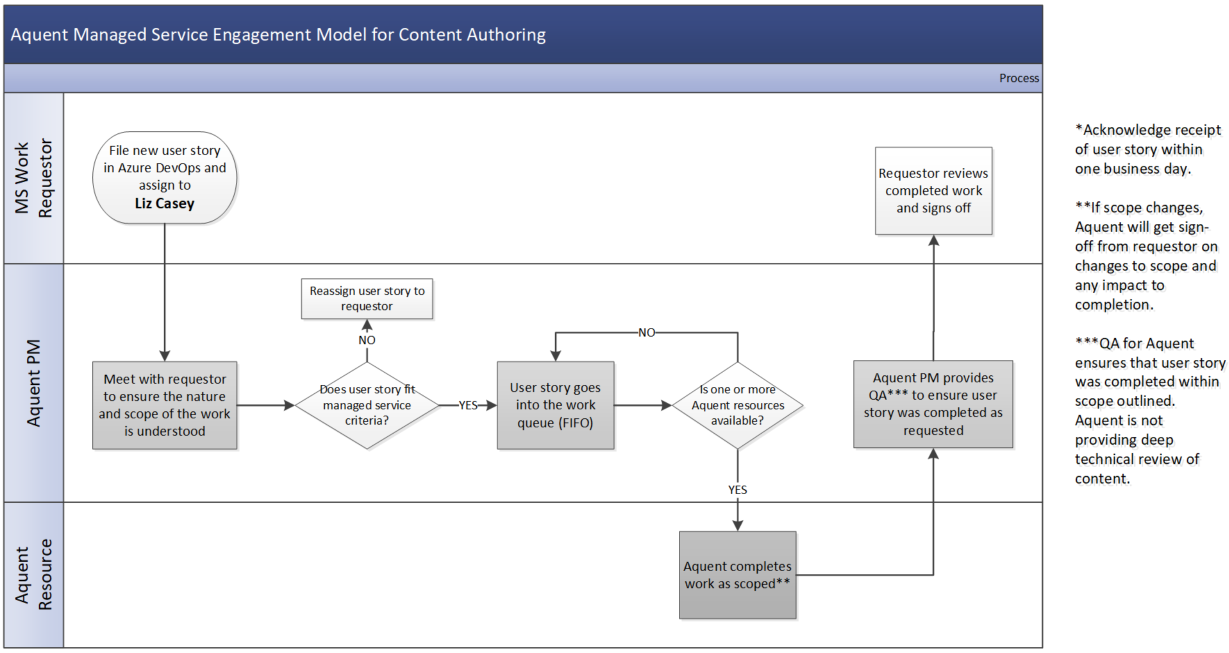
The following expectations provide compliance with Microsoft policies for external resourcing:

- Any new work using a product group budget will be completed using the staff augmentation model, rather than the managed service.

- If you need vendor writers to whom you can assign an area of work like you would to an FTE, use staff augmentation vendors. Staff augmentation positions require a job description and must be posted. In this case, Microsoft will interview the candidates, make the hiring decision, and manage the vendor writer.
- All work must be tracked using user stories in Azure Boards.
- Each user story's work will be scoped before it begins, to provide both an understanding of the volume of work and its position in the Aquent work queue/schedule.
- FTEs can't interview or select the staff within the managed service who will perform the work. Microsoft will let Aquent know the knowledge/skills needed by the managed service team, and Aquent will staff the service to meet those needs.
- Route feedback in a timely manner to Justin Chappell.

Work request process diagram

You can use the Aquent content authoring managed service for clearly defined and scoped content authoring tasks. Request work by creating user stories in Azure Boards:



How to obtain a staff augmentation vendor

If your content needs can't be met by this managed service model, hiring a staff augmentation vendor writer might be an option.

Questions/concerns/feedback

- Justin Chappell – Microsoft Program Manager for Aquent managed service
- Laura Belmont – Aquent service manager

Aquent content developer experience and skills

5/10/2021 • 2 minutes to read

The Aquent content authoring team provides a wide variety of skills and knowledge. This is a managed service available to Content & Learning content teams only. For more information on scope and engagement, see Aquent technical content authoring managed service.

If a proposed task requires expertise that is not listed below, feel free to contact Aquent's service manager Laura Belmont (v-laurab) to see whether the team can accommodate that task.

Coding skills

The Aquent team has expertise in reading, editing, and writing the following coding languages:

- .NET
- ASP.NET
- Azure CLI
- C#
- C++
- Java
- JavaScript
- JSON
- Node.js
- PHP
- PowerShell
- Python
- Ruby
- SQL
- Shell scripting (Bash)
- T-SQL
- VBA
- XML/XSLT

Technology/platform familiarity

The Aquent team has varying levels of familiarity with a wide range of technology areas, as shown in the following table, where:

- Familiar: know what it is and could get up to speed quickly.
- Conversant: good general knowledge but not an expert.
- Expert: deep understanding.

TECHNOLOGY	FAMILIAR	CONVERSANT	EXPERT
Azure Compute (Virtual Machines, Azure Batch, Azure Service Fabric)		X	

TECHNOLOGY	FAMILIAR	CONVERSANT	EXPERT
Azure Containers (Azure Batch, Azure Service Fabric)		X	
Azure Data + Analytics (HDInsight, Stream Analytics, Data Lake Analytics, Data Lake Store)		X	
Azure Databases (SQL, SQL Data Warehouse, Azure Cosmos DB, Azure Database for MySQL)		X	
Azure Developer Tools (Azure DevTest Labs, Azure DevOps, Xamarin, Application Insights)			X
Azure IOT (Hub, Edge, Steam Analytics, Machine Learning)			X
Azure Monitoring + Management (Azure Policy, Azure Monitor, Azure Backup)	X		
Azure Networking (Virtual Network, Load Balancer, Azure DNS)	X		
Azure Security + Identity (Active Directory, Multi-Factor Authentication)	X		
Azure Stack	X		
Azure Storage (Storage, Data Lake Store)		X	
Azure Web + Mobile (App Service, CDN, Media Services, Azure Search)	X		
Enterprise Mobility + Security (Intune, Identity Manager)		X	
Power Apps		X	
Power BI		X	
SQL Server			X
System Center		X	

TECHNOLOGY	FAMILIAR	CONVERSANT	EXPERT
Visual Studio			X

The Aquent team is also familiar with many platforms and operating systems, including:

PLATFORM	FAMILIAR	CONVERSANT	EXPERT
Windows			X
Linux		X	
Mac OS		X	
UNIX		X	

Content types

The Aquent content authoring team has extensive experience with writing conceptual technical content, how-to content, developer reference, and code samples, among others.

Schedule of ongoing quality tasks

5/10/2021 • 2 minutes to read

The Lionbridge content production team handles some ongoing tasks, in addition to ad-hoc work submitted by content teams. They handle localization fixes on demand, but other tasks follow regular cadences. Cadences are defined for each type of quality work, per repo.

The broken link workstream was migrated to Lionbridge in June 2020, and the team is phasing into the work, by adding a new repo each week. The Policheck workstream is a restart of a workstream that has not been in operation for a year or longer, and is restarting now as a pilot, with the team adding a new repo every other week as per the cadences below.

REPO	BROKEN LINKS	POLICHECK
architecture-center-pr	Active – Weekly	Active – Weekly
azure-csp-pr	Active – Monthly	Active – Monthly
azure-dev-docs-pr	Active – Weekly	Active – Weekly
azure-devops-docs-pr	Active – Weekly	Active – Weekly
azure-docs-pr	Active – Weekly	Active – Weekly
azure-reference-other-pr	Active – Monthly	Active – Monthly
azure-stack-docs-pr	Active – Monthly	Active – Monthly
cloud-adoption-framework-pr	Active – Weekly	Active – Weekly
cloudappsecuritydocs-pr	Active - Quarterly	Active - Quarterly
cpp-docs-pr	Active – Monthly	Active – Monthly
dataexplorer-docs-pr	Active – Weekly	Active – Weekly
emdocs-pr	Active - Quarterly	Active - Quarterly
intellicode-pr	Active – Monthly	Active – Monthly
IntuneEDU-pr	Active - Quarterly	Active - Quarterly
learn-pr	Active – Weekly	Active – Weekly
memdocs-pr	Active – Weekly	Active – Weekly
MIMDocs-pr	Active - Quarterly	Active - Quarterly
powerbi-docs-pr	Active – Monthly	Active – Monthly

REPO	BROKEN LINKS	POLICHECK
sql-docs-pr	Active – Weekly	Active – Weekly
SystemCenterDocs-pr	Active – Monthly	Active – Monthly
visualstudio-docs-pr	Active – Weekly	Active – Weekly

Check on in-progress work

The Lionbridge team tracks work via User Stories. These queries can help you find the current User Story for the repo(s) you're interested in:

- Query for in-progress broken link user tasks
- Query for localization fix user stories

Questions/concerns/feedback

- Justin Chappell – Microsoft project owner and content business SME
- Christopher McClister – Lionbridge project manager

Azure top 150 content freshness/maintenance

5/10/2021 • 2 minutes to read

To ensure that the most trafficked Azure articles remain “fresh”, the Aquent vendor content authoring team will perform a review each month and do a freshness pass on content that has not been updated in the last four months. This recurring task is now part of the ongoing quality work that’s part of the managed service.

Mission

On a monthly basis, the Aquent vendor content authoring team will identify Azure articles in the top 150 most-trafficked articles to identify any articles that have exceeded the 120 day freshness point (E:>120). An Aquent PM or writer will contact the author to let them know the content is past 120 days and that the Aquent team is available to update the article if the author agrees and can provide tech review at the end of the update. If the author agrees to the update, the Aquent team will conduct a full freshness pass.

Process

The Aquent vendor team will use this process to identify content for this work stream:

1. On or after the 10th of the month, visit https://aka.ms/contentperformancedashboard. Verify that data is posted for the prior month.
2. On the **PerformanceOverview** tab, click the link to download the page level detailed report.
3. From the OneDrive share, navigate to folder for the previous month and then download the Excel worksheet for high-traffic content. Choose the option to open the Desktop app for Excel.
4. In the Excel worksheet, enable content and turn on filtering. Then filter and sort as follows:
   - In the **LiveUrl** column, click **Text Filters** > **Contains** and enter `docs.microsoft.com/en-us/azure`.
   - Sort the **PageViews** column from largest to smallest.
   - Filter the **Freshness** column to `E:>120`.

   The list of articles that remains should be conceptual Azure content where the `ms.date` value has not been changed in over 120 days (4 months).
5. For each article, check value in the **LastCommitDateTime** column for the last commit. Frequently, a recent date may show for the last commit – check the **LinesChanged** column to determine if it was a major or minor update.
6. Contact the listed author by email using this template:

   > Hello *name*,
   >
   > I’m reaching out to you because you are listed as the author of this Azure technical article on docs.microsoft.com:
   >
   > - *Article name, hotlinked to the live page*
   >
   > This article is one of the top 150 most-visited Azure articles on docs.microsoft.com this month, and it does not appear to have been updated recently (based on the ms.date value). My team is tasked with ensuring the top 150 articles are kept up to date. Could you let me know if you are okay with us updating this article to ensure it is still complete, accurate, and current? When I’ve finished the

> update, I will ask you for review and sign off – I'll need your sign off within 5 days of my completion of the update (see the SLA details for more info).
>
> If you would rather update the article yourself, please let me know.
>
> Thanks, *Aquent staffer*

7. If you don't hear back from the author within 5 days, try to contact the author by IM. If the author is on vacation, please note their return date from their OOF and log this in the work item; follow up with them when they return by IM and email. At that point, if there is no response, let the author know via email that you are proceeding with the update.

8. Complete the article update per standard freshness processes.

Questions/concerns/feedback

- Justin Chappell – Microsoft project owner and content business SME
- Tami Fosmark – Aquent project manager

Repository coverage for PRMerger automation, vendor pull request review, and Acrolinx

6/9/2021 • 2 minutes to read

Private repos

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
Azure/azure-docs-powershell-aip	No	No	Yes
Azure/azure-docs-rest-apis	Yes	Yes	Yes
dotnet/docs-internal	No	No	Yes
dotnet/docs-maui-internal	Yes	No	Yes
MicrosoftDocs/ai-docs-pr	No	No	Yes
MicrosoftDocs/altspace-vr-pr	No	No	Yes
MicrosoftDocs/appcenter-docs-pr	Yes	No	Yes
MicrosoftDocs/architecture-center-pr	Yes	Yes	Yes
MicrosoftDocs/atadocs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-csp-pr	Yes	Yes	Yes
MicrosoftDocs/azure-dev-docs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-devops-docs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-docs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-reference-other-pr	Yes	Yes	Yes, VSCode only
MicrosoftDocs/Azure-RMSDocs-pr	Yes	Yes	Yes
MicrosoftDocs/azure-stack-docs-pr	Yes	Yes	Yes

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/azure-stack-tzl-docs-pr	Yes		No
MicrosoftDocs/bi-shared-docs-pr	Yes	Yes	Yes, GitHub only
MicrosoftDocs/bot-docs-pr			Yes, GitHub only
MicrosoftDocs/cloud-adoption-framework-pr	Yes	Yes	Yes
MicrosoftDocs/cloudappsecuritydocs-pr	Yes	Yes	Yes
MicrosoftDocs/composer-docs-pr			Yes, GitHub only
MicrosoftDocs/cpp-docs-pr	Yes	Yes	Yes
MicrosoftDocs/cspp-pr	No	No	Yes
MicrosoftDocs/dataexplorer-docs-pr	Yes	Yes	Yes
MicrosoftDocs/emdocs-pr	Yes	Yes	Yes
MicrosoftDocs/enterprise-graph-docs-pr	No	No	Yes
MicrosoftDocs/fslogix-docs-pr	No	No	Yes
MicrosoftDocs/hybrid-pr	Yes	Yes	Yes
MicrosoftDocs/intellicode-pr	Yes	Yes	Yes
MicrosoftDocs/IntuneEDU-pr	Yes	Yes	Yes
MicrosoftDocs/learn-bizapps-pr	Yes	Yes	Yes
MicrosoftDocs/learn-dynamics-pr	Yes	Yes	Yes
MicrosoftDocs/learn-m365-pr	Yes	Yes	Yes
MicrosoftDocs/learn-mec-pr	No	Yes	Yes
MicrosoftDocs/learn-pr	Yes	Yes	Yes

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/LearnShared	Yes	No	Yes
MicrosoftDocs/machine-learning-server-docs-pr	No	No	Yes
MicrosoftDocs/media-services-pr		No	Yes
MicrosoftDocs/MEMDocs-pr	Yes	Yes	Yes
MicrosoftDocs/MIMDocs-pr	Yes	Yes	Yes
MicrosoftDocs/mixed-reality-pr	No	No	Yes
MicrosoftDocs/partner-center-pr	Yes	Yes	Yes
MicrosoftDocs/partner-center-sdk-pr	Yes	Yes	Yes
MicrosoftDocs/power-automate-docs-pr	Yes	No	Yes
MicrosoftDocs/powerbi-docs-javascript		No	Yes, VSCode only
MicrosoftDocs/powerbi-docs-pr	Yes	Yes	Yes
MicrosoftDocs/salvador-sdk-pr	No	No	Yes
MicrosoftDocs/sccm-docs-powershell	Yes	No	Yes
MicrosoftDocs/security-benchmark-docs-pr	No	No	Yes
MicrosoftDocs/socrates-pr	No	No	Yes
MicrosoftDocs/sql-docs-pr	Yes	Yes	Yes
MicrosoftDocs/SystemCenterDocs-pr	Yes	Yes	Yes
MicrosoftDocs/visualstudio-docs-pr	Yes	Yes	Yes, VSCode only
MicrosoftDocs/windows-ai-docs-pr	No	No	Yes, VSCode only

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/windows-uwp-pr	Yes	Yes	No
MicrosoftDocs/windowsserverdocs-pr	Yes	Yes	Yes
MicrosoftDocs/winrt-related-pr	Yes	Yes	No
MicrosoftDocs/workload-repository-pr	No	No	No
MicrosoftDocs/Xamarin-docs-pr	Yes	No	Yes

Public repos

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
aspnet/Docs	No	No	Yes, VSCode only
aspnet/EntityFramework.Docs	No	No	Yes, VSCode only
Azure/azure-docs-powershell	n/a	n/a	No
Azure/azure-docs-powershell-azuread	n/a	n/a	No
Azure/azure-docs-sdk-dotnet	No	No	No
Azure/azure-docs-sdk-java	No	No	No
Azure/azure-docs-sdk-node	No	No	No
Azure/azure-docs-sdk-python	No	No	No
dotnet/docs	No	No	Yes, VSCode only
dotnet/dotnet-api-docs	No	No	Yes, VSCode only
dotnet/docs-maui	Yes	No	Yes, VSCode only
Microsoft/vscode-docs	No	No	Yes, VSCode only
MicrosoftDocs/architecture-center	Yes	Yes	Yes

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/atadocs	Yes	Yes	Yes
MicrosoftDocs/azure-dev-docs	Yes	Yes	Yes, VSCode only
MicrosoftDocs/azure-devops-docs	Yes	Yes	Yes
MicrosoftDocs/azure-docs	Yes	Yes	Yes
MicrosoftDocs/azure-reference-other	Yes	Yes	No
MicrosoftDocs/Azure-RMSDocs	Yes	Yes	Yes
MicrosoftDocs/azure-stack-docs	Yes	Yes	Yes
MicrosoftDocs/biztalk-docs	No	No	No
MicrosoftDocs/cloud-adoption-framework	Yes	Yes	Yes
MicrosoftDocs/cloudappsecuritydocs	Yes	Yes	Yes
MicrosoftDocs/cpp-docs	Yes	Yes	Yes
MicrosoftDocs/dataexplorer-docs	Yes	Yes	Yes
MicrosoftDocs/docs-powershell (This repo has only one article and only 35 PRs over time)	n/a	n/a	n/a
MicrosoftDocs/emdocs	Yes	Yes	Yes
MicrosoftDocs/hybrid	Yes	Yes	Yes
MicrosoftDocs/IntuneEDU	Yes	Yes	Yes
MicrosoftDocs/machine-learning-server-docs	No	No	Yes
MicrosoftDocs/MEMDocs	Yes	Yes	Yes
MicrosoftDocs/MIMDocs	Yes	Yes	Yes
MicrosoftDocs/mixed-reality	No	No	Yes

REPOSITORY	PRMERGER ENABLED?	VENDOR PR REVIEW?	ACROLINX?
MicrosoftDocs/msix-docs	No	No	Yes, VSCode only
MicrosoftDocs/partner-center-sdk	Yes	Yes	Yes
MicrosoftDocs/power-automate-docs	Yes	No	Yes
MicrosoftDocs/powerbi-docs	Yes	Yes	Yes
MicrosoftDocs/PowerShell-Docs	n/a	n/a	Yes, VSCode only
MicrosoftDocs/project-rome	No	No	Yes, VSCode only
MicrosoftDocs/sql-docs	Yes	Yes	Yes
MicrosoftDocs/visualstudio-docs	Yes	Yes	Yes, VSCode only
MicrosoftDocs/windows-iotcore-docs	No	No	Yes, VSCode only
MicrosoftDocs/windows-uwp	Yes	Yes	No
MicrosoftDocs/winrt-api	Yes	Yes	No
MicrosoftDocs/winrt-related	Yes	Yes	No
MicrosoftDocs/winui-api	Yes	Yes	No

DOCS Common Tasks

3/5/2021 • 2 minutes to read

TASK	ARTICLE	LINK
Onboarding		
How can I get Private Repo Access?	Contribution Quickstart	https://aka.ms/msdocsonboard
Where is my content?	List of business approvers and repositories	https://aka.ms/msdocsrepos
Contribute Content		
I want to make a small doc change	GitHub contribution workflow for minor or infrequent changes	https://aka.ms/msdocsminor
I want to make a large doc change	GitHub contribution workflow for major or long-running changes	https://aka.ms/msdocsmajor
What's all this Metadata?	Metadata Reference	https://aka.ms/msdocsmetadata
Table of Contents updates		https://aka.ms/msdocsupdatetoc
Taking Screenshots		https://aka.ms/msdocimages
Documentation Performance		
How do I get metrics on my Docs?		https://aka.ms/get-mdm
How do I improve the SEO of my content?		https://aka.ms/msdocsseo
Repository Guidance		
Move content between Public and Private Repos	Article authors working in the public repo	https://aka.ms/msdocsreposwap
Can I use CodeFlow for content reviews?	Use CodeFlow to review PRs	https://aka.ms/msdocscodeflow
Can I get an Editing pass?	Repository coverage for PRMerger automation, vendor pull request review, and Acrolinx	https://aka.ms/msdocsrepoautomation
Reference Documentation		
What repo is my reference content published from?	List of reference repos by language and product	https://aka.ms/refrepolist

TASK	ARTICLE	LINK
How do I create or update REST API reference docs?	Update and onboarding instructions for REST API content	REST API Documentation
Community Engagement		
How do I respond to questions/comments on a Doc		https://aka.ms/msdocscommunity

Best practices for contributors

6/1/2021 • 2 minutes to read

Set up and maintain your work environment

Don't put your local repo in a OneDrive-synced folder

When you clone a repo, don't install to a folder that syncs with OneDrive (for example, your **Documents** folder). OneDrive can cause unintended merge conflicts when you merge your changes back to the main branch. For simplicity, it's best to create a **GitHub** folder on your C:\ drive (for example, ***C:\GitHub***).

Make sure you can access help channels

Many content teams have a Microsoft Teams channel you can join to get answers to your questions. Use these channels if you can't find help in the contributor guides or if you get stuck. Join the help channel that corresponds to your content group.

CONTENT GROUP	HELP CHANNEL
Content & Learning	Docs Support
MARVEL	Ask an admin

Work in branches to isolate changes

Never work directly in a main branch. You should always work in your own working branch and then merge those changes to the main branch.

If you'll be working on different projects, or even just updating topics with different timelines, you should work in multiple working branches. Using multiple branches let you work and save your changes in isolated environments.

Keep your branches (and forks) in sync

Refresh your branches and forks often and *always* before you start editing content. Keeping your branches in sync is the best way to minimize annoying merge conflicts. For more information, see Keep branches up to date in GitHub.

Author content

Always check which branch you're in before you start work

It's easy to open a file in Visual Studio Code and start editing, only to discover you're in the wrong branch. You can move your changes to your correct branch, but it's prudent to get in the habit of checking which branch you're in before you start editing.

Stash your work

If you're working in one of your branches on a longer-term project and need to make some quick updates to some other files, you can stash your changes before switching branches. Git will preserve the state of your in-progress files while you work in another branch. When you return to the first branch, you restore your stashed files and pick up where you left off.

You can stash files using Git Bash or GitHub Desktop.

Visual Studio Code and Markdown

- When you create a new article, copy the metadata from another topic and update as needed.
- Yellow squiggly lines indicate some kind of syntax warning. Hover over the squiggle to see an explanation and possible fix.
- Check the preview window for errors that are hard to spot in the code window (for example, text formatting, links, and images).

Publish content

Pull requests

Before you submit your pull request, check your content against the pull release criteria to minimize your risk of blocking issues. Each content organization has its own criteria. Be sure to check your work against the correct list.

- C&L criteria
- MARVEL criteria

Review content

Set up read-only or editable review links to make it easier for your reviewers. For more information, see Set up a link for tech reviewers.

Work with partners

Help internal partners who aren't familiar with the PubOps system and GitHub get started by creating stub files and providing them with branch and repo names. For more information, see Working with internal contributors.

Visual Studio Code keyboard shortcuts

6/1/2021 • 4 minutes to read

These are the most common keyboard shortcuts that writers use in Visual Studio Code. The complete set of Vshortcuts can found in Visual Studio Code under **File > Preferences > Keyboard Shortcuts**. For keyboard shortcuts for other operating systems and additional unassigned shortcuts, see Key bindings for Visual Studio Code.

General

COMMAND	ACTION
Ctrl + Shift+ P, F1	Show **Command Pallet**
Ctrl + P	Quick open, go to ...
Ctrl + Shift + N	New window/instance
Ctrl + Shift + W	Close window/instance
Ctrl + ,	User settings
Ctrl + K + S	Keyboard shortcuts

Basic editing

COMMAN	ACTION
Ctrl + X	Cut line (empty selection)
Ctrl + C	Copy line (empty selection)
Alt + ↑/↓	Move line up/down
Shift + Alt + ↑/↓	Copy line up/down
Ctrl + Shift + K	Delete line
Ctrl + Enter	Insert line below
Ctrl + Shift + Enter	Insert line above
Ctrl + Shift + \	Insert line above
Ctrl +]/[	Indent/outdent line
Home/End	Go to beginning/end of line

COMMAN	ACTION
Ctrl + Home	Go to beginning of file
Ctrl + End	Go to end of file
Ctrl + ↑/↓	Scroll line up/down
Alt + PgUp/PgDn	Scroll page up/down
Ctrl + Shift + [	Fold (collapse) region
Ctrl + Shift +]	Unfold (expand) region
Ctrl + K + [	Fold (collapse) all subregions
Ctrl + K +]	Unfold (expand) all subregions
Ctrl + K + 0	Fold (collapse) all regions
Ctrl + K + J	Unfold (expand) all regions
Ctrl + K + C	Add line comment
Ctrl + K + U	Remove line comment
Ctrl + /	Toggle line comment
Shift + Alt + A	Toggle block comment
Alt + Z	Toggle word wrap

Navigation

COMMAND	ACTION
Ctrl + T	Show all symbols
Ctrl + G	Go to line...
Ctrl + P	Go to file...
Ctrl + Shift + O	Go to symbol...
Ctrl + Shift + M	Show **Problems** panel
F8	Go to next error or warning
Shift + F8	Go to previous error or warning
Ctrl + Shift + Tab	Navigate editor group history

COMMAND	ACTION
Alt + ←/→	Go back/forward
Ctrl + M	Toggle **Tab Moves Focus** setting

Search and replace

COMMAND	ACTION
Ctrl + F	Find
Ctrl + H	Replace
F3/Shift + F3	Find next/previous
Alt + Enter	Select all occurrences of **Find** match
Ctrl + D	Add selection to next **Find** match
Ctrl + K + D	Move last selection to next **Find** match
Alt + C/R/W	Toggle **Case-sensitive/Regex/Whole word**

Multicursor and selection

COMMAND	ACTION
Alt + Click	Insert cursor
Ctrl + Alt + ↑/↓	Insert cursor above/below
Ctrl + U	Undo last cursor operation
Shift + Alt + I	Insert cursor at end of each selected line
Ctrl + I	Select current line
Ctrl + Shift + L	Select all occurrences of current selection
Ctrl + F2	Select all occurrences of current word
Shift + Alt + →	Expand selection
Shift + Alt + ←	Shrink selection
Shift + Alt + (drag mouse)	Column selection
Ctrl + Shift + Alt + (arrow keys)	Column selection
Ctrl + Shift + Alt + PgUp/PgDn	Column selection page up/page down

Rich languages editing

COMMAND	ACTION
Ctrl + Space	Trigger suggestion
Ctrl + Shift + Space	Trigger parameter hints
Shift + Alt + F	Format document
Ctrl + K + F	Format selection
F12	Go to definition
Alt + F12	Peak definition
Ctrl + K F12	Open definition to the side
Ctrl + .	Quick fix
Shift + F12	Show references
F2	Rename symbol
Ctrl + K + X	Trim trailing whitespace
Ctrl + K M	Change file language

Editor management

COMMAND	ACTION
Ctrl + F4 + W	Close editor
Ctrl + K F	Close folder
Ctrl + \	Split editor
Ctrl + 1/2/3	Focus into first/second/third editor group
Ctrl + K + ←/→	Focus into previous/next editor group
Ctrl + Shift + PgUp/PgDn	Move editor left/right
Ctrl + K ←/→	Move active editor group

File management

COMMAND	ACTION
Ctrl + N	New file

COMMAND	ACTION
Ctrl + O	Open file
Ctrl + S	Save
Ctrl + Shift + S	Save as...
Ctrl + K S	Save all
Ctrl + F4	Close
Ctrl + K + W	Close all
Ctrl + Shift + T	Reopen closed editor
Ctrl + K Enter	Keep preview mode editor open
Ctrl + Tab	Open next
Ctrl + Shift + Tab	Open previous
Ctrl + K P	Copy path of active file
Ctrl + K R	Reveal active file in Explorer
Ctrl + K O	Show active file in new window/instance

Display

COMMAND	ACTION
F11	Toggle full screen
Shift + Alt + 0	Toggle editor layout (horizontal or vertical)
Ctrl + =/-	Zoom in/out
Ctrl + B	Toggle sidebar visibility
Ctrl + Shift + E	Show Explorer/toggle focus
Ctrl + Shift + F	Show Search
Ctrl + Shift + G	Show Source Control
Ctrl + Shift + D	Show Debug
Ctrl + Shift + X	Show Extensions
Ctrl + Shift + H	Replace in files

COMMAND	ACTION
Ctrl + Shift + J	Toggle Search details
Ctrl + Shift + U	Show Output panel
Ctrl + Shift + V	Open Markdown preview
Ctrl + K V	Open Markdown preview to the side
Ctrl + K Z	Zen mode (Esc + Esc to exit)

Debug

COMMAND	ACTION
F9	Toggle breakpoint
F5	Start/continue
Shift + F5	Stop
F11/Shift + F11	Step into/out
F10	Step over
Ctrl + K + I	Show hover

Integrated terminal

COMMAND	ACTION
Ctrl + '	Show integrated terminal
Ctrl + Shift + '	Create new terminal
Ctrl + C	Copy selection
Ctrl + V	Paste into active terminal
Ctrl + ↑/↓	Scroll up/down
Shift + PgUp/PgDn	Scroll page up/down
Ctrl + Home/End	Scroll to top/bottom

Repository roles and responsibilities

5/10/2021 • 4 minutes to read

In Content & Learning documentation repositories, as a writer and contributor, you'll interact with different people in different roles within the publishing and quality review process. This article describes the roles and the responsibilities that correspond to those roles. The roles are:

- Repository administrator
- Business approver
- Quality program manager
- Pull request reviewer
- GitHub program manager (DocOps PM)

Different repositories handle the roles of repo admin and business approver differently.

- In some repos, the repo admin may also be the business approver.
- In the azure-docs-pr repository, there are different business approvers for different things, and the repo admin is a separate person.

Repository admin

The repository admin is responsible for the integrity of the repository. The admin is also responsible for the day-to-day operation of the repository.

- Makes sure that publishing is occurring on a documented, planned, and recurring schedule. Manages OOB and hotfix publishing.
- Creates release branches.
- Responsible for ensuring that the repository is not blocked by warnings, errors, or other publishing pipeline problems.
- Maintains the integrity of the repository by reviewing and merging pull requests that are out of scope for the pull request review team.
- Makes sure that configuration file changes are correct per documentation and that new features are turned on only after approval by the contributor guide team, by leadership, or based on current status of testing or experimentation.
- Makes sure that permissions are current and appropriate for the repository.
- Makes sure to engage the support team when necessary to keep authors unblocked.
- Until automation is provided, the admin closes abandoned pull requests that are more than three months old on a monthly basis and cleans up old or unused branches.

Business approver

The business approver is responsible for ensuring that all content that is merged meets the guidelines described in the contributor guide and other relevant standards. As an accountability in Content & Learning, the business approver role maps to content team managers.

For most routine content, the accountability for this work is delegated to the vendor pull request team managed by the vendor supported quality PM.

Business approvers must review and approve or decline certain types of change:

- Hub and landing page updates
- Escalations: pull requests that don't meet the pull request criteria and where the author indicates they want an exception.
- Merging of stub articles to a release branch. Content managers must approve these PRs to ensure that follow up occurs so that the stub articles are complete before publication. Even with approval, the stub articles need to follow the PR criteria for stub articles.

Business approvers must be documented in the contributor guide and should be content team managers. Business requirements may mean others are assigned as business approvers based on domain expertise – for example, Elizabeth Ross is a business approver for GDPR exceptions.

Most business approvers are tracked in the docs metadata tables in the ContentManager and ContentApprover columns. If ownership changes, follow the Change metadata process.

Azure business approvers are listed in the Service content owners article. Other business approvers are listed in the Business approvers and repos article.

Quality PM

The Quality PM leverages outsourced vendor and software resources to extend the capabilities of Content & Learning.

- Drives alignment with content guidelines across the organization through vendor PR review and editorial resources.
- Works with business approvers and repository admins on escalations and cross-repo processes to ensure consistency where multiple roles intersect in the PR review and merging process.
- Drives alignment with content guidelines across the organization through Acrolinx editorial and content strategy automation software.
- Collaborates with Contribution Guide owners to incorporate decisions into Acrolinx and the human PR review process.
- Drives vendor-based programs to improve content freshness and accuracy at time of publish.
- Manages Aquent and Acrolinx contracts and relationships. Single point of contact for these engagements.

Pull request reviewer

The pull request reviewer is accountable for ensuring that content meets the defined minimum quality bar for publication. Pull request reviewers review all pull requests that contain new articles and significant changes. They provide feedback to authors when the new articles or updated content fail to meet the published criteria. In some repos, the administrator may also perform this role, but in most Content & Learning repositories, a vendor team does this work. This vendor team reports to the quality PM, whose role is described in the previous section.

Pull request reviewers evaluate content against the content channel guidance and against the pull request review criteria:

- Content channel guidance
- Pull request review criteria

GitHub/DocOps PM

The GitHub and workflow PM is responsible for maintaining and improving Content & Learning workflow patterns and automation across all repositories in support of content quality and improved contributor workflow.

- Ensures that feedback from authors, partners, and Content & Learning and non-Content & Learning stakeholders is collected and integrated into GitHub authoring workflow improvements and automation.
- Works to streamline and automate GitHub authoring processes where possible and in a way that supports the quality and governance imperatives of other stakeholders in the process.
- Works with Content & Learning engineering PM/dev to ensure Content & Learning requirements are part of engineering planning.
- Drives Content & Learning strategies all up for increasing the depth of engagement with community in GitHub (issues, public repos, etc.).
- Planning and managing the eventual break up of the azure-docs-pr repository.
- Large event release management (Build, Ignite).
- Works with other workflow stakeholders to incorporate requirements. .

List of business approvers and repositories

6/2/2021 • 3 minutes to read

Repository and approver

REPOSITORY	APPROVERS	GITHUB ALIAS
dotnet/AspNetDocs	Mark LeBlanc	mcleblanc
dotnet/AspNetCore.Docs	Mark LeBlanc	mcleblanc
dotnet/EntityFramework.Docs	Mark LeBlanc	mcleblanc
dotnet/docs	Mark LeBlanc	mcleblanc
MicrosoftDocs/architecture-center-pr	Ed Price	EdPrice-MSFT
MicrosoftDocs/atadocs-pr	Rachel Karlin	rkarlin
MicrosoftDocs/azure-china-pr	Yue Shi (CELA); Martin Ekuan	-
MicrosoftDocs/azure-devops-docs-pr	Mike Jacobs	mijacobs KathrynEE Steved0x
MicrosoftDocs/azure-devops-extensions	Mike Jacobs	mijacobs
MicrosoftDocs/azure-dev-docs-pr	Barbara Kess Karl Erickson	barbkess KarlErickson
MicrosoftDocs/azure-docs-pr	Martin Ekuan, Mike Tillman, and the content publishing manager for the service	martinekuan mtillman. For this repo, business approver and repo admin are separate. Cory Fowler is the admin.
MicrosoftDocs/azure-reference-other	Jason Howell Carolyn Gronlund	JasonWHowell (stream-analytics-docs, u-sql) CJGronlund (studio-module-reference)
MicrosoftDocs/Azure-RMSDocs-pr	Rachel Karlin	rkarlin
MicrosoftDocs/azure-stack-docs-pr	Femila Anilkumar	femila
MicrosoftDocs/bot-docs-pr	Kamran Iqbal, Erik Hopf, Nitin Mehrotra	Kaiqb, erhopf, nitinme
MicrosoftDocs/bi-shared-docs-pr	Owen Duncan	owend
MicrosoftDocs/bi-shared-docs	Owen Duncan	owend

REPOSITORY	APPROVERS	GITHUB ALIAS
MicrosoftDocs/cloud-adoption-framework-pr	Alex Buck	abuck
MicrosoftDocs/cloudappsecuritydocs-pr	Rachel Karlin	rkarlin
MicrosoftDocs/composer-docs-pr	Erik Hopf, Nitin Mehrotra	erhopf, nitinme
MicrosoftDocs/cognitive-toolkit-docs	Thiago Crepaldi Minna Xiao	thiagocrepaldi mx-iao liqunfu
MicrosoftDocs/commercialization-docs-pr	Eliot Graff Ted Maroutsos	EliotSeattle themar
MicrosoftDocs/cpp-docs-pr	Colin Robertson	corob-msft
MicrosoftDocs/dataexplorer-docs-pr	Rachel Karlin	rkarlin
MicrosoftDocs/devops-learn-pr	Mike Jacobs	mijacobs
MicrosoftDocs/DocsLegal-pr	Duncan Mackenzie	DuncanmaMSFT
MicrosoftDocs/emdocs-pr	Doug Eby	dougeby
MicrosoftDocs/flow-docs-pr	Kumar Vivek	KumarVivek
MicrosoftDocs/fslogix-docs-pr	Mike Tillman	mtillman
MicrosoftDocs/intellicode-pr	Mark Wilson-Thomas Josée Martens	markw-t j-martens
MicrosoftDocs/IntuneDocs-pr	Doug Eby	dougeby
MicrosoftDocs/IntuneEDU-pr	Doug Eby	dougeby
MicrosoftDocs/iis-docs	Mark LeBlanc	mcleblanc
MicrosoftDocs/IIS.Administration-docs	Mark LeBlanc	mcleblanc
MicrosoftDocs/kusto	Rachel Karlin	rkarlin
MicrosoftDocs/live-share	Jonathan Carter Josée Martens	lostintangent j-martens
MicrosoftDocs/machine-learning-server-docs-pr	David Peter Hansen	dphansen
MicrosoftDocs/MachineLearning-Python-pr	Carolyn Gronlund	cjgronlund
MicrosoftDocs/memdocs-pr	Doug Eby	dougeby

REPOSITORY	APPROVERS	GITHUB ALIAS
MicrosoftDocs/MIMDocs-pr	Mike Tillman Bill Mathers	mtillman billmath
MicrosoftDocs/mlnet-pr	Yiwen Zhu (NimbusML)	zyw400
MicrosoftDocs/msix-pr	John Kennedy	GrantMeStrength
MicrosoftDocs/partner-center-docs-powershell	Jason Howell	JasonWHowell
MicrosoftDocs/partner-center-java-docs	Jason Howell	JasonWHowell
MicrosoftDocs/partner-center-pr	Jason Howell	JasonWHowell
MicrosoftDocs/partner-center-sdk-pr	Jason Howell	JasonWHowell
MicrosoftDocs/partner-china-redirect	Jason Howell	JasonWHowell
MicrosoftDocs/partner-rest-pr	Jason Howell	JasonWHowell
MicrosoftDocs/powerbi-docs-pr	Kate Follis	kfollis
MicrosoftDocs/project-rome	John Kennedy	GrantMeStrength
MicrosoftDocs/query-docs-pr	Owen Duncan Vivek Kumar	owend kvivek
MicrosoftDocs/SCCMDocs-pr	Doug Eby	dougeby
MicrosoftDocs/stream-docs-pr	Archana Nair	archnair
MicrosoftDocs/sql-docs-pr	Craig Guyer Mike Ray Jason Roth	craigg-msft MikeRayMSFT rothja
MicrosoftDocs/visualstudio-docs-pr	Josée Martens Gordon Hogenson	j-martens ghogen
Microsoft/vscode-docs	Greg van Liew	gregvanl
MicrosoftDocs/win32-pr	John Kennedy	GrantMeStrength
MicrosoftDocs/windows-ai-docs-pr	John Kennedy	GrantMeStrength
MicrosoftDocs/windows-driver-docs-pr	Eliot Graff Ted Hudek	EliotSeattle tedhudek
MicrosoftDocs/windows-iotcore-docs	Martin Ekuan	martinek
MicrosoftDocs/windows-uwp-pr	John Kennedy Mike Jacobs	GrantMeStrength mijacobs

REPOSITORY	APPROVERS	GITHUB ALIAS
MicrosoftDocs/windowslegaldocs	Maggie Evans	MaggiePucciEvans
MicrosoftDocs/windowsserverdocs-pr	Liz Ross	eross-msft
microsoftgraph/microsoft-graph-docs/	Celeste De Guzman	celestedg
NuGet/docs.microsoft.com-nuget	Karan Nandwani	karann-msft
Samples		
Azure-Samples/azure-cli-samples	Sean Wheeler Delora Bradish	sdwheeler dbradish-microsoft
Azure/azure-docs-powershell-samples	Sean Wheeler Mike Robbins	sdwheeler mikefrobbins
dotnet/samples	Mark LeBlanc	mcleblanc
Microsoft/Windows-appsample-annotated-audio	Karl Erickson	KarlErickson
Microsoft/windows-appsample-coloringbook	Karl Erickson	KarlErickson
Microsoft/windows-appsample-customers-orders-database	Karl Erickson	KarlErickson
Microsoft/windows-appsample-familynotes	Joshua Partlow	JoshuaPartlow
Microsoft/windows-appsample-get-started-js2d	John Kennedy	GrantMeStrength
Microsoft/windows-appsample-get-started-js3d	John Kennedy	GrantMeStrength
Microsoft/windows-appsample-get-started-mg2d	John Kennedy	GrantMeStrength
Microsoft/windows-appsample-huelightcontroller	Joshua Partlow	JoshuaPartlow
Microsoft/windows-appsample-lunch-scheduler	Brendan Mitchell	bmitchell287
Microsoft/windows-appsample-marble-maze	Karl Erickson	KarlErickson
Microsoft/windows-appsample-networkhelper	Karl Erickson	KarlErickson
Microsoft/windows-appsample-photo-editor	Joshua Partlow	JoshuaPartlow

REPOSITORY	APPROVERS	GITHUB ALIAS
Microsoft/windows-appsample-photo-lab	Karl Erickson	KarlErickson
Microsoft/windows-appsample-remote-system-sessions	Brendan Mitchell	bmitchell287
Microsoft/windows-appsample-rssreader	Karl Erickson	KarlErickson
Microsoft/windows-appsample-trafficapp	Karl Erickson	KarlErickson
Microsoft/windows-appsample-xaml-hosting	Joshua Partlow	JoshuaPartlow
Microsoft/windows-task-snippets	Karl Erickson	KarlErickson
Microsoft/windows-tutorials-customer-database	Quinn Radich	QuinnRadich
Microsoft/windows-tutorials-inputs-and-devices	Karl Bridge	Karl-Bridge-Microsoft
API Reference		
aspnet/apidocs	Mark LeBlanc	mcleblanc
Azure/azure-docs-sdk-dotnet	Mark LeBlanc	mcleblanc
Azure/azure-docs-sdk-java	Robert Outlaw	rloutlaw
Azure/azure-docs-sdk-python	Robert Outlaw	rloutlaw
Azure/azure-docs-rest-apis	Robert Outlaw	rloutlaw
Azure/azure-docs-cli-python	Robert Outlaw	rloutlaw
dotnet/automl-api-docs	Natalie Kershaw	natke
dotnet/dotnet-api-docs	Mark LeBlanc	mcleblanc
dotnet/fsharp-api-docs	Mark LeBlanc	mcleblanc
dotnet/ml-api-docs	John Alexander	JRAlexander
dotnet/roslyn-api-docs	Mark LeBlanc	mcleblanc
MicrosoftDocs/azure-docs-sdk-node	Robert Outlaw	rloutlaw
MicrosoftDocs/azure-iot-c-sdk-reference	Wesley McSwain	wesmc7777

REPOSITORY	APPROVERS	GITHUB ALIAS
MicrosoftDocs/PowerShell-Docs	Sean Wheeler Joey Aiello	sdwheeler joeyaiello
MicrosoftDocs/powershell-docs-sdk-dotnet	Sean Wheeler Joey Aiello	sdwheeler joeyaiello
MicrosoftDocs/project-rome-java-api-docs	John Kennedy	GrantMeStrength
MicrosoftDocs/spark-dotnet-pr	Luis Quintanilla	luisquintanilla
MicrosoftDocs/visualstudio-docs-dotnet-api-pr	Anthony Cangialosi Josée Martens	acangialosi j-martens
MicrosoftDocs/vsts-docs-rest-apis	Will Smythe Jessie Huang	
MicrosoftDocs/vsts-docs-sdk-dotnet	Will Smythe Ted Chambers	
MicrosoftDocs/vsts-docs-cli-python	Will Smythe Ted Chambers	
MicrosoftDocs/windows-vision-skills-api-pr	John Kennedy	GrantMeStrength
MicrosoftDocs/windows-partner-center-dotnet	Jason Howell	JasonWHowell
MicrosoftDocs/winrt-related-pr	John Kennedy	jken

Azure DevOps repos:

REPOSITORY	APPROVERS	MSFT ALIAS
cpubwin/partner	Jason Howell	JasonWHowell
TechnicalContent/keros-docs-pr	Wendy Chisholm Mike Jacobs	wendc mijacobs
API reference		
cpubwin/advertising-sdk	John Kennedy	jken
cpubwin/sdk-api	John Kennedy	jken
cpubwin/store-api	John Kennedy	jken
cpubwin/wdk-ddi	Eliot Graff	EliotSeattle
cpubwin/winrt-api	John Kennedy	jken

REPOSITORY	APPROVERS	MSFT ALIAS
cpubwin/winui-api	John Kennedy	jken

Business approver list for Azure services

6/2/2021 • 7 minutes to read

This page lists the business approvers that oversee and ensure high quality for Azure services. The list helps PR Reviewers and other contributors know who to contact when approvals are needed or questions come up for a certain Azure technology. Business approver responsibilities are documented in the Roles and responsibilities article.

> **NOTE**
>
> There is a second similar list of managers that is used for reporting and automation. Follow the link to the **Detail** view from the `ms.service` list at Metadata taxonomies to find the corresponding Manager value for each row there.

How to update this list

By default, the content publishing manager accountable for a service is considered the business approver for that service. Content team managers are responsible for keeping this business approver list up to date. If an individual contributor is identified as a content lead with business approver responsibility, that person's manager needs to update this list.

There are two places to update:

1. Update this page (click Edit) to change or add new rows into the markdown table below. Keep the table sorted alphabetically, ignoring the prefix word Azure in the sort.
2. Also, follow the metadata changes steps to submit a request to update the ContentManager column in the metadata tracking database.

Business approver list

Refer to the table below for the name and GitHub alias of the business approver for each Azure Service:

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
A		
Active Directory	Mike Tillman	mtillman
Active Directory Domain Services	Mike Tillman	mtillman
Active Directory Federation Services	Mike Tillman	mtillman
Azure AD B2B	Celeste de Guzman	celestedg
Azure AD B2C	Celeste de Guzman	celestedg
Azure AD Application Management	Celeste de Guzman	celestedg
Azure AD Application & HR Provisioning	Celeste de Guzman	celestedg

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Azure AD for developers / Microsoft identity platform	Celeste de Guzman	celestedg
Azure Active Directory	Mike Tillman, Celeste de Guzman	mtillman, celestedg
Advisor	David Berry	DavidCBerry13
Analysis Services	Kate Follis	kfollis
API Management	George Wallace	georgewallace
Azure API for FHIR	Carolyn Gronlund	cjgronlund
App Center	George Wallace	georgewallace
App Configuration	John Kennedy	jken
App Service	George Wallace	georgewallace
Application Gateway	Kumud Dwivedi	KumudD
Application Insights	David Berry	DavidCBerry13
Arc for Servers	Maggie Evans	MaggiePucciEvans
Arcadia A365	Jason Roth, Craig Guyer	rothja, craigg-msft
Azure Artifacts	Mike Jacobs	mijacobs
Azure Communication Services	John Kennedy	jken
Automation	Maggie Evans	MaggiePucciEvans
Availability Zones	Mary McCready	mamccrea
Avere vFXT for Azure	Femila Anilkumar	femila
B		
Backup	Femila Anilkumar	femila
Bastion	Kumud Dwivedi	KumudD
Batch	Maggie Evans	MaggiePucciEvans
Batch AI	Carolyn Gronlund	cjgronlund
Big Compute scenarios	Maggie Evans	MaggiePucciEvans
Billing	Jason Howell	jasonwhowell

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Bing News Search	Nitin Mehrotra	nitinme
BizTalk Services	Doug Eby	dougeby
Blob	Tessa Wooley	twooley
Blockchain	Femila Anilkumar	femila
Azure Blueprints	Maggie Evans	MaggiePucciEvans
Azure Boards	Mike Jacobs	mijacobs
Azure Bot Service Composer	Erik Hopf, Nitin Mehrotra	erhopf, nitinme
Azure Bot Service SDK (Framework)	Erik Hopf, Nitin Mehrotra	erhopf, nitinme
C		
Azure Cache for Redis	Barbara Kess	barbkess
Azure CLI	Barbara Kess	barbkess
Content Delivery Network (CDN)	Kumud Dwivedi	KumudD
Cloud Services	Mary McCready	mamccrea
Cognitive Services	Nitin Mehrotra, Erik Hopf	nitinme, erhopf
ConfigMgr	Doug Eby	dougeby
Container Instances	George Wallace	georgewallace
Container Registry	George Wallace	georgewallace
Azure Cosmos DB	Jason Howell	jasonwhowell
Cost Management	Jason Howell	jasonwhowell
Cost Management by Cloudyn	Jason Howell	jasonwhowell
D		
Data Box	Femila Anilkumar	femila
Data Catalog	Jason Howell	jasonwhowell
Data Factory	Jason Howell	jasonwhowell
Data Lake Analytics	Jason Howell	jasonwhowell

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Data Lake Storage Gen2	Tessa Wooley	twooley
Data Lake Storage Gen1 (was Data Lake Store)	Tessa Wooley	twooley
Data Share	Jason Howell	jasonwhowell
Azure Data Science Virtual Machines	Carolyn Gronlund	cjgronlund
Azure Databricks	Jason Howell	jasonwhowell
Database Migration Service	Jason Roth, Craig Guyer	rothja, craigg-msft
DDos Protection	Kumud Dwivedi	KumudD
Dedicated HSM	Rachel Karlin	rkarlin, mbaldwin
Deployment Manager	Carmon Mills	carmonm
Azure Device Certification	Phil Meadows	philmea
Device Update for IoT Hub	Phil Meadows	philmea
Azure Dev Spaces	George Wallace	georgewallace
Azure DevOps (all)	Mike Jacobs, Kathryn Elliott, Steve Danielson	mijacobs, KathrynEE, Steved0x
Azure DevOps Project	Mike Jacobs	mijacobs
Azure DevOps Starter	Mike Jacobs	mijacobs
Azure DevOps - Agile/Work	Mike Jacobs	mijacobs
Azure DevOps - CI/CD	Mike Jacobs	mijacobs
Azure DevOps - Collaboration	Mike Jacobs	mijacobs
Azure DevOps - Ecosystem & Integration	Mike Jacobs	mijacobs
Azure DevOps - Git	Mike Jacobs	mijacobs
Azure DevOps - Package	Mike Jacobs	mijacobs
Azure DevOps - SCS	Mike Jacobs	mijacobs
Azure DevOps - Test	Mike Jacobs	mijacobs
Digital Twins	Elizabeth Ross	lizross

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Disk Storage	Tessa Wooley	twooley
Azure DNS	Kumud Dwivedi	KumudD
E		
Azure Edge Zones	Kumud Dwivedi	KumudD
Event Grid	Jason Howell	jasonwhowell
Event Hubs	Jason Howell	jasonwhowell
ExpressRoute	Kumud Dwivedi	KumudD
F		
Azure Firewall	Kumud Dwivedi	KumudD
Azure Firewall Manager	Kumud Dwivedi	KumudD
Azure Front Door Service	Kumud Dwivedi	KumudD
Azure Functions	George Wallace	georgewallace
Azure FXT Edge Filer	Femila Anilkumar	femila
FHIR Server for Azure	dseven	-
Files	Tessa Wooley	twooley
G		
Genomics	Carolyn Gronlund	cjgronlund
Azure GO	Barbara Kess	barbkess
Azure Government	Femila Anilkumar	femila
Azure Governance	Maggie Evans	MaggiePucciEvans
H		
Azure HDInsight	Jason Howell	jasonwhowell
HPC Cache	Femila Anilkumar	femila
Azure Hub Page (articles/index.md)	Martin Ekuan, Mike Tillman, Craig Guyer	martinekuan, mtillman, craigg-msft
I		

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Internet Analyzer	Kumud Dwivedi	KumudD
Intune	Doug Eby	dougeby
Intune for Education	Doug Eby	dougeby
IoT Central	Elizabeth Ross	lizross
IoT Device Developer	Phil Meadows	philmea
IoT Device Provisioning	Elizabeth Ross	lizross
IoT Edge	Elizabeth Ross	lizross
IoT Fundamentals	Elizabeth Ross	lizross
IoT Hub	Elizabeth Ross	lizross
IoT PnP	Elizabeth Ross	lizross
J		
Azure Java	Barbara Kess	barbkess
Azure JavaScript	Barbara Kess	barbkess
K		
Key Vault	Rachel Karlin, Matthew Baldwin	rkarlin, mbaldwin
Kinect DK	Martin Ekuan	martinekuan
Azure Kubernetes Service (formerly Azure Container Service)	George Wallace	georgewallace
Kusto/Data Explorer	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
L		
Azure Lab Services	Femila Anilkumar	femila
Azure Lighthouse	Maggie Evans	MaggiePucciEvans
Linux Virtual Machines	Mary McCready	mamccrea
Load Balancer	Kumud Dwivedi	KumudD
Log Analytics	David Berry	DavidCBerry13

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Logic Apps	Mary McCready	mamccrea
M		
M365 Business	Doug Eby	dougeby
M365 Enterprise	Doug Eby	dougeby
Azure Machine Learning	Carolyn Gronlund, Larry Franks	cjgronlund, blackmist
Managed Disks	Tessa Wooley	twooley
Managed Applications	Maggie Evans	MaggiePucciEvans
Management Groups	Maggie Evans	MaggiePucciEvans
Azure Maps	Phil Meadows	philmea
Azure Database for MariaDB	Jason Roth	rothja
Marketplace	Jason Howell	jasonwhowell
Media Services	Femila Anilkumar	femila
Microsoft Identity Manager	Mike Tillman	mtillman
Microsoft Information Protection	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Azure Migrate	Maggie Evans	MaggiePucciEvans
Mixed Reality	Martin Ekuan	martinekuan
Azure Mobile App	Mike Tillman, Kate Follis	mtillman, mgblythe
Mobile Engagement	Matt Soucoup	codemillmatt
Mobile Services	Matt Soucoup	codemillmatt
Azure Monitor	David Berry	DavidCBerry13
Mooncake / Azure in China	Femila Anilkumar	femila
Multi-Factor Authentication	Mike Tillman	mtillman
Azure Database for MySQL	Jason Roth	rothja
N		
Azure NetApp Files	Tessa Wooley	twooley

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Network Watcher	Kumud Dwivedi	KumudD
Azure Notebooks	Joshua Partlow	JoshuaPartlow
Notification Hubs	Femila Anilkumar	femila
O		
Azure Orbital	Kumud Dwivedi	KumudD
Open Datasets	Carolyn Gronlund	cjgronlund
OMS	David Berry	DavidCBerry13
P		
Peering Service	Kumud Dwivedi	KumudD
Azure Percept	Phil Meadows	philmea
Azure Pipelines	Mike Jacobs	mijacobs
Azure portal	Mike Tillman, Kate Follis	mtillman, mgblythe
Azure Policy	Maggie Evans	MaggiePucciEvans
Azure Database for PostgreSQL	Jason Roth	rothja
Azure Powershell	Barbara Kess	barbkess
Power BI	Rachel Karlin, Kate Follis	rkarlin, kfollis
Power BI Embedded	Rachel Karlin, Kate Follis	rkarlin, kfollis
Private Link	Kumud Dwivedi	KumudD
Azure Purview	Jason Howell	jasonwhowell
Azure Python	Barbara Kess	barbkess
Q		
Azure Quantum	Carolyn Gronlund, Kitty Yeung	cjgronlund, KittyYeungQ
Queue	Tessa Wooley	twooley
R		
Azure Red Hat Open Shift	George Wallace	georgewallace

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
RemoteApp	Femila Anilkumar	femila
Azure Remote Rendering	Florian Born	florianborn71
Azure Repos	Mike Jacobs	mijacobs
Reserved Instances	Mary McCready	mamccrea
Azure Resource Graph	Maggie Evans	MaggiePucciEvans
Resource Health	Carmon Mills	carmonmills
Resource Manager	Maggie Evans	MaggiePucciEvans
Rights Management	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Role-based access control (RBAC)	Mike Tillman	mtillman
Azure Route Server	Kumud Dwivedi	KumudD
S		
Scheduler	Mary McCready	mamccrea
Search	Heidi Steen, Nitin Mehrotra	HeidiSteen, nitinme
Security	Rachel Karlin	rkarlin
Security Center	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Security Center for IoT	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Azure Sentinel	Rachel Karlin, Matthew Baldwin, Shlomo Sagir	rkarlin, mbaldwin, shsagir
Server Management	Carmon Mills	carmonmills
Service Bus Messaging	Jason Howell	jasonwhowell
Service Bus Relay	Jason Howell	jasonwhowell
Service Fabric	George Wallace	georgewallace
Service Fabric Mesh	George Wallace	georgewallace
Service Health	David Berry	DavidCBerry13
Service Map	Carmon Mills	carmonmills

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
SignalR	Eliot Graff	EliotSeattle
Site Recovery	Maggie Evans	MaggiePucciEvans
Azure Spring Cloud	Barbara Kess	barbkess
SQL Data Warehouse	Jason Roth	rothja
SQL Database	Jason Roth, steve stein	rothja, stevestein
SQL Server Stretch Database	Jason Roth	rothja
SQL Virtual Machines	Jason Roth	rothja
Azure Stack	Femila Anilkumar	femila
Static Web Apps	George Wallace	georgewallace
Storage	Tessa Wooley	twooley
Storage - Blob	Tessa Wooley	twooley
Storage - Disk Storage	Tessa Wooley	twooley
Storage - Files	Tessa Wooley	twooley
Storage - Queue	Tessa Wooley	twooley
StorSimple	Femila Anilkumar	femila
Stream Analytics	Jason Howell	jasonwhowell
Azure Supportability	Michael Blythe	mgblythe
Synapse Analytics	Jason Roth	rothja
System Center	Carmon Mills	carmonmills
T		
Azure Test Plans	Mike Jacobs	mijacobs
Time Series Insights	Ornat Spodek	orspodek
Traffic Manager	Kumud Dwivedi	KumudD
U		
V		

SERVICE/PRODUCT (A-Z)	BUSINESS APPROVER	BUSINESS APPROVER'S GITHUB ALIAS
Virtual Machine Scale Sets	Mary McCready	mamccrea
Virtual Machines	Mary McCready	mamccrea
Virtual Network	Kumud Dwivedi	KumudD
Virtual WAN	Kumud Dwivedi	KumudD
Azure VMware Solution by Virtustream	George Wallace	georgewallace
VPN Gateway	Kumud Dwivedi	KumudD
W		
Web Application Firewall	Kumud Dwivedi	KumudD
Web Apps	George Wallace	georgewallace
Web Apps on Linux (Containers)	George Wallace	georgewallace
Windows Virtual Machines	Mary McCready	mamccrea
Windows Virtual Desktop	Femila Anilkumar	femila
X		
Y		
Z		

Cheat sheet: basic Git command-line steps

4/2/2021 • 3 minutes to read

This cheat sheet is an abridged version of the steps outlined in Tutorial: Make changes to an article published on docs.microsoft.com. As a prerequisite, you must have already forked and cloned a repository (repo).

Create a new article, or update an article

These steps use MicrosoftDocs/azure-docs-pr as an example, but you can substitute any org/repo combination you've cloned. These steps also assume your fork is at the remote named `origin` and the upstream published repo is at the remote named `upstream`.

1. Launch your Git command-line tool (for example, Git Bash).
2. Change to your local repository:

   ```
   cd azure-docs-pr
   ```

3. Check out the default branch:

   ```
   git checkout master
   ```

4. Create a new local working branch derived from the master branch:

   ```
   git fetch upstream master
   git checkout -b <working branch> master
   ```

5. Let your fork know you created the local working branch:

   ```
   git push origin <working branch>
   ```

6. Create your new article or make changes to an existing article.
7. After you create or modify your article and images, add and commit the changes you made:

   ```
   git add .
   git commit -m "<comment>"
   ```

   Or, if you wish to only add the specific files you modified:

   ```
   git add <file path>
   git commit -m "<comment>"
   ```

8. Push the changes to your fork on GitHub:

   ```
   git push origin <working branch>
   ```

9. When you're ready to submit your content to the upstream master branch for staging, validation, and/or

publishing, go to the repo in GitHub (for example, https://github.com/MicrosoftDocs/azure-docs-pr) and create a pull request (PR) from your fork to the master branch.

10. If you're an employee working in the private repository, the changes you submit are automatically staged and a staging link is written to the PR. Review your staged content and sign off in a PR comment by adding the `#sign-off` comment. This indicates the changes are ready for review and to be published live.

11. If your update is small in scope, the PR may qualify for automatic merge. If not, the PR review team reviews your PR, provides feedback, and merges it if the PR meets the minimum quality criteria.

Work in multiple working branches

When you create more than one working branch, and you want to switch back-and-forth between them to iterate on different sets of work, do the following to prevent moving commits from one working branch to another working branch.

> **IMPORTANT**
> If you unintentionally move commits from one working branch to another, you will co-mingle files and edits from the first working branch into the second working branch.

1. When you're in a working branch and you want to get out of it, check if you have untracked files (changes that aren't staged for a commit):

   ```
   git status
   ```

   a. If you have untracked files, in the Git Bash output they're shown in red font: go to Step 2.

   b. If you don't have untracked files, in the Git Bash output no files are shown in red font: go to Step 4.

2. Add and commit the changes in the untracked files:

   ```
   git add .
   git commit -m "<comment>"
   ```

3. Push the changes to your fork on GitHub:

   ```
   git push origin <working branch>
   ```

4. Check out the other pre-existing working branch:

   ```
   git checkout <other working branch>
   ```

5. Create a new article or make changes to an existing article. Add, commit, and push the changes to your fork on GitHub.

> **NOTE**
> If you do the procedure above and more than about an hour goes by, the prompt in the GitHub UI to create a PR disappears. To create a PR, see Create a pull request.

Next steps

- See these steps in full detail: Tutorial: Make changes to an article published on docs.microsoft.com
- Partner quick reference
- Learn more about Git

Create and publish sample code

6/8/2021 • 6 minutes to read

Learn how to create a sample code repository in the `Azure-Samples` GitHub organization for backing your articles on docs.microsoft.com.

> **IMPORTANT**
> These steps are for Azure sample repos for programming languages like Java, Node.js, or .NET, not scripting environments such as Azure CLI and Azure PowerShell. For samples guidance for those scripting languages, see Getting started with samples.

Prerequisites

- A GitHub account.
- A development environment to develop your sample.
- Review the Governance for Code samples at Microsoft before creating the content.

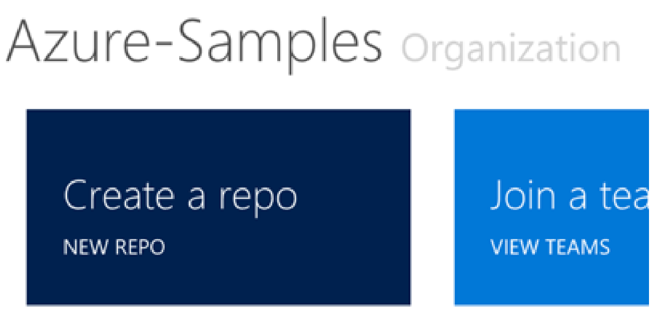
Associate your GitHub identity with your Microsoft identity

To request a new repo in the Azure-Samples organization and manage its permissions, you must first associate your GitHub identity with your Microsoft domain identity. Follow the steps the tutorial to associate your GitHub identity with your Microsoft one and use the Open Source portal to join the `Azure-Samples` organization.

Request a new repo

To create a new sample repo in the `Azure-Samples` organization:

1. Sign in to the Microsoft Open Source Hub
2. Click "Join Azure-Samples".
3. Select **GitHub** in the top menu.
4. Select the **Azure-Samples** link, then select the **Create a repo** box.



You're presented with a wizard-like experience for creating the new repo. Use the following values when prompted:

- For **Release registration**, choose **New release registration**.
- For **New release registration**, choose **Yes** to confirm it's a public open-source project.

> **NOTE**
>
> You can create private repos and make them public later when you're ready to release the sample.

Select **Sample code** for project type, **MIT** license, and fill in the appropriate answers for:

- Other source/binary code usage
- Whether the sample collects any personal data, including data sent in HTTP headers to REST API calls to Azure services.
- Confirm that the sample is less than 5,000 lines of code.

Select **Next** to confirm your choices.

Name your repo

On the **Create a new repository** page, enter a name for your new repo. Use the following rules to accurately describe your sample:

- Avoid using terms like "sample" or "getting started" in your repo name.
- All lowercase, words separated with dashes.
- Name should start with the service slug, as it is represented in docs.microsoft.com/azure. The easiest way to know your service slug is to go to the service marketing on azure.microsoft.com (example: https://azure.microsoft.com/services/search/ so the slug is `search`).
- After the slug, there should be a language or platform: dotnet, java, node, python
- Next, describe in two or three words what the sample does when run. It's often easiest to write out a sentence describing the repo, then extract the key terms in the sentence to create the rest of the name.

Examples:

- `mysql-java-create-todo-app`
- `cosmosdb-nodejs-create-mongodb-rest-api`

Add a repo description

The description text is visible on the repo page on GitHub. Limit the description to ~120 characters. Include what the sample is (web app, API, application template), the Azure services the sample uses, and the languages and/or frameworks used in the sample.

Examples:

- `Java bookshelf app using SQL Database and Azure Storage`
- `.NET REST API with Cosmos DB's SQL API`

Assign permissions

Assign at least one team to manage the repo settings for the sample repo. If you already belong to a team, that team name will be presented here. If you're not part of a team, create a new one in the Azure-Samples organization, then refresh this page.

Select the name of the team with Admin permissions for the repo, then select **Next**.

Select additional teams with write permissions to your repo. These teams can accept PRs and push commits to your repo. It's recommended that you add an Azure content development team to the write permissions. If problems with the repo are reported in the associated docs, the content team can correct and merge the changes.

Assign any Read Permissions in the same way.

In the **Contributor License Agreement**, leave **Microsoft** as the Legal Entity and select **Next**. Select an appropriate prepopulated `.gitignore` based on the language used in the sample, then select **Next**.

Review and submit the application for your repo.

Complete the README

Once the repo request is approved and the repo is created, the following files are added:

- README.md - Template for your sample's README. **Complete all sections of the README**. Do not alter the *Contributing* section.
- CODE_OF_CONDUCT.md - Contains links to Microsoft's open-source code of conduct documentation.
- LICENSE.md - Azure samples standardize around the MIT license.
- SECURITY.md - Information for reporting security issues found in the sample.

README title & description

Title - The main title (H1) of the README file (the line of text after the first `#` in markdown) is used as the title of the sample on both the details page and the docs.microsoft.com samples browser. Limit it to 120 characters or less.

Description - Under the title (H1), add a short explanation describing your sample. Show a screenshot of your sample running if it's web page or other interactive application. If it's a console application, include a formatted code block tagged with `console` that shows its usage and output.

- Link back to the appropriate quickstart or tutorial content on docs.microsoft.com if the sample is used in product documentation. If the sample is not used directly in content, link back to the Azure docs landing page for the service or developer center for that developer language.
- If using REST APIs or Azure client libraries in your sample, make sure to link both to the reference on docs.microsoft.com for the library and the GitHub where the client library is developed.

README content

The rest of the Markdown describes your sample. Complete each section of the README template:

- **Contents** - Outline the file contents of the repository.
- **Prerequisites** - Required components and tools to build and/or run the sample.
- **Setup** - How to prepare to run the sample.
- **Running the sample** - Step-by-step instructions to execute the sample and see its output. Include instructions on modifying the sample code for use in another project.
- **Key concepts** - Context on the tools and services used in the sample.
  - Add learn more links that go out to deeper content tutorials on docs.microsoft.com or other relevant samples on GitHub.
  - Point out which lines of code in the sample contain interesting Azure code, and make a short table with descriptions linking directly to those lines of code.

README metadata

The metadata at the top of the readme file helps the docs.microsoft.com platform associate the sample with Azure services and languages/platforms and improves their discoverability. Having the metadata section is required for a sample to show up on the docs.microsoft.com samples browser.

For detailed information about completing the YAML frontmatter metadata, see:

Onboarding Samples to docs.microsoft.com/samples

Best practices for sample code

- Sample code that creates billable Azure service instances and resources should clean up those resources after it finishes execution and the user terminates the sample. Don't remove the resources without confirming with the user. If possible, clean up resources even if an unhandled exception occurs at runtime.
- If you need access keys or connection strings for a service, point out to use the Azure CLI, PowerShell, or Cloud Shell to do so. If Azure CLI or PowerShell can't be used, point to content on docs.microsoft.com that shows how to find the values in the Azure portal.
- When mentioning class names in Azure client libraries in your README, link straight to the class in the API reference on docs.microsoft.com.

Accessibility tools and references

5/6/2021 • 4 minutes to read

Accessibility is an important part of contributing towards Microsoft's goal of inclusivity, and our mission to empower every person and every organization on the planet to do more.

IMPORTANT

Every team at Microsoft must design its products with every user in mind

-- Satya Nadella

TIP

Official guidance is available on the intranet site: Microsoft Accessibility.

We recommend reviewing the Accessibility 101 training to get a baseline understanding of the accessibility needs of our customers.

Accessibility tools

There are many tools available to assist you in your evaluation of content to accessibility standards. The following sections list a series of tools that can assist you with making your content accessible.

Discovery accessibility of content

IMPORTANT

Microsoft uses Accessibility Insights to test accessibility requirements. We need to implement the accessibility features so that both Microsoft Narrator and NonVisual Desktop Access (NVDA) produce the same user experience across both the Microsoft Edge browser and the Chrome browser.

Accessibility Insights for Web

Accessibility Insights is a browser plugin that you can install for Chrome or Microsoft Edge. It helps developers find and fix accessibility issues in web apps and sites.

WINDOWS	MACOS	COMMAND
`Ctrl + Shift + K`	`Shift + ⌘ + K`	Activate the extension
`Ctrl + Shift + 1`	`Shift + ⌘ + 1`	Toggle Automated checks
`Ctrl + Shift + 2`	`Shift + ⌘ + 2`	Toggle Landmarks
`Ctrl + Shift + 3`	`Shift + ⌘ + 3`	Toggle Headings

Test accessible content

Windows Narrator

Narrator lets you use your PC without a mouse to finish common tasks if you're a person who is blind or has low vision. It reads and interacts with things on the screen, like text and buttons. Use Narrator to read and write email, browse the internet, and work with documents.

Shortcut keys

Shortcut keys will help you go to a page when using Narrator. If you would like a full list of keyboard shortcuts, review Appendix B: Narrator keyboard commands and touch gestures.

KEYBOARD SHORTCUT	NOTES
`Windows + Ctrl + Enter`	Start/Stop Narrator
`Esc`	Exit Narrator
Narrator[1] + ←	Change Views in Narrator (previous)
Narrator[1] + →	Change Views in Narrator (next)

[1] By default the Narrator key is set up as both `Caps Lock` and `Insert`

NVDA

NVDA is a free and open-source screen reader for Windows. You can download it from the NVDA Download page. If you would like to learn more about how to use NVDA, review their User Guide.

NAME	DESKTOP KEY	DESCRIPTION
Stop speech	control	Instantly stops speaking.
Pause Speech	shift	Instantly pauses speech. Pressing shift again will continue speech from where it left off (if the current synthesizer supports pausing).
NVDA Menu	NVDA+n	Pops up the NVDA menu to allow you to access preferences, tools, help, and so on.
Toggle Speech Mode	NVDA+s	Toggles speech mode between speech, beeps, and off.
Toggle Input Help Mode	NVDA+1	Pressing any key in this mode will report the key, and the description of any NVDA command associated with it.
Quit NVDA	NVDA+q	Exits NVDA.

Create accessible experiences

ColorBox

Created by Lyft, ColorBox ensures that hue, saturation, and luminosity applied on a color is distinguishable to people with vision disabilities.

Accessibility guidelines

Microsoft put Accessibility Guidelines in place so there's a consistent experience for people who have disabilities. Microsoft is a global company and as such has to meet government compliance with accessibility standards in different nations.

VERSION	STANDARD	GOVERNING ENTITY
WCAG 2.1	Web Content Accessibility Guidelines	W3C
MAS 2.1	Microsoft Accessibility Standards	Microsoft

The Microsoft Accessibility Standards are a superset of the Web Content Accessibility Guidelines. Encompassing the W3C guidelines ensures our standards meet compliance needs in all regions around the world. The following table, assists you in determining the priority of an accessibility bug in your content.

Accessibility Severity Level

Storefront Priority Definitions

Severity 1: Not MAS compliant.

This issue is a showstopper and constitutes a block-ship bug. There are no known workarounds. Must be fixed immediately.

Priority 0: Must fix immediately - blocking with no workaround

Severity 2: Not MAS compliant.

This issue is either non-blocking for core user tasks or blocking for non-core user tasks. Remediation action needed ASAP or within three months post release with an approved exception!

Priority 1: Must fix as soon as possible - blocking some areas;

OR, depending on actual experience,

Priority 2: Should fix soon - before product release.

Severity 3: Not MAS compliant, but low user impact.

Remediation action required in the next major release or the next time the site is updated, whichever occurs first.

Priority 3: Fix if time

Severity 4: Technically a MAS violation, but negligible user impact.

For bugs with this lowest impact, the division accessibility driver can approve a “won’t fix” designation.

Priority 3: Fix if time

Disability etiquette

Disability etiquette is a set of voice principles to reduce the risk of offending people with disabilities. Microsoft specifically adheres to the United States standards of people first language (PFL).

> **IMPORTANT**
> You should consider Disability Etiquette when describing how the customer is interacting with web technologies. To ensure that you are using input-neutral terms, review Describing interactions with UI.

*Disability terminology reference guide

Green (preferred/recommended language)	Yellow (acceptable language)	Red (insensitive/offensive language)
Person with a cognitive disability	Developmentally disabled	Retarded/Mental Retardation/Brain damaged
Person with a learning disability	Trending green Dyslexic person	Reading problems/SPED/Special Ed
Person who is Deaf or Hard of Hearing	Trending green Hearing disability/ Deaf person	Mute/Hearing Impaired
Person with a physical disability or person with motor disability	Physically disabled	Wheelchair Bound/Crippled
Person with Autism	Trending green Autistic person	Slow/Dumb/Weird
Person who is blind	Trending green Blind person	Sight challenged
Person with a mental health disability	Suffers from mental illness	Crazy/Insane/Nuts/Psycho
Person with a disability	Disabled person	Handicapped
Everybody, including people with disabilities	People of all abilities	Differently abled
Person without a disability	Non-disabled person	Normal person/Healthy person

Accessibility guidelines for images and videos

11/2/2020 • 7 minutes to read

This page summarizes the accessibility guidelines for images, videos, and animated GIFs.

Alt text requirements for images

Alternative text (alt text) that follows these guidelines is required for images that convey information like diagrams and screenshots.

- A minimum length of 40 characters and a maximum of 150.
- For complex images, supplement alt text with adjacent text in the content or a long description.
- Begin with the type of graphic, for example "Screenshot of..." or "Diagram showing...".
- Don't start with "Image..." or "Graphic...". Screen readers say this automatically.
- End with a period so the screen reader can identify the end of the alt text.
- Omit extraneous elements like figure numbers or bold/italic formatting.
- Add spaces to acronyms where each character should be read (e.g. "H T M L") but not for words (e.g. "NASA").
- Decorative images (images that do not convey information) and icons do **not** include alt text. Use the `:::image:::` Markdown extension with `type="icon"` which will output an empty alt attribute.

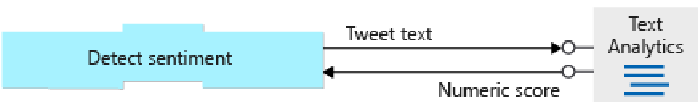
> **TIP**
>
> Use the `:::image:::` Markdown extension for all images. It supports simple images with alt text, complex images with both alt text and a long description, and decorative images / icons having no alt text.

Alt text best-practices for images

The following recommendations show example alt text for several technical images. For more information, see Microsoft's non-text content training.

Simple images

Consider the following image:



GUIDELINE	EXAMPLE
Express the core idea of the image	Appropriate alt text: *"Diagram showing a Logic App using the detect-sentiment action to invoke the Text Analytics service."*
Don't describe the image literally	Inappropriate alt text: *"Diagram of Text Analytics service usage. Lines with arrows connect the two elements and show text flowing to the service and a numeric score being returned."*
Don't write in the style of a title or caption	Inappropriate alt text: *"Use of the Text Analytics service."*

GUIDELINE	EXAMPLE
Don’t duplicate content in the alt text and elsewhere on the page	Lead-in sentence preceding the image: *"The following image shows a Logic App invoking the Text Analytics service:"* Appropriate alt text that is distinct from lead-in: *"Diagram showing a Logic App passing a tweet to the Text Analytics service and receiving a sentiment score in return."*

Complex images

Consider the following image:

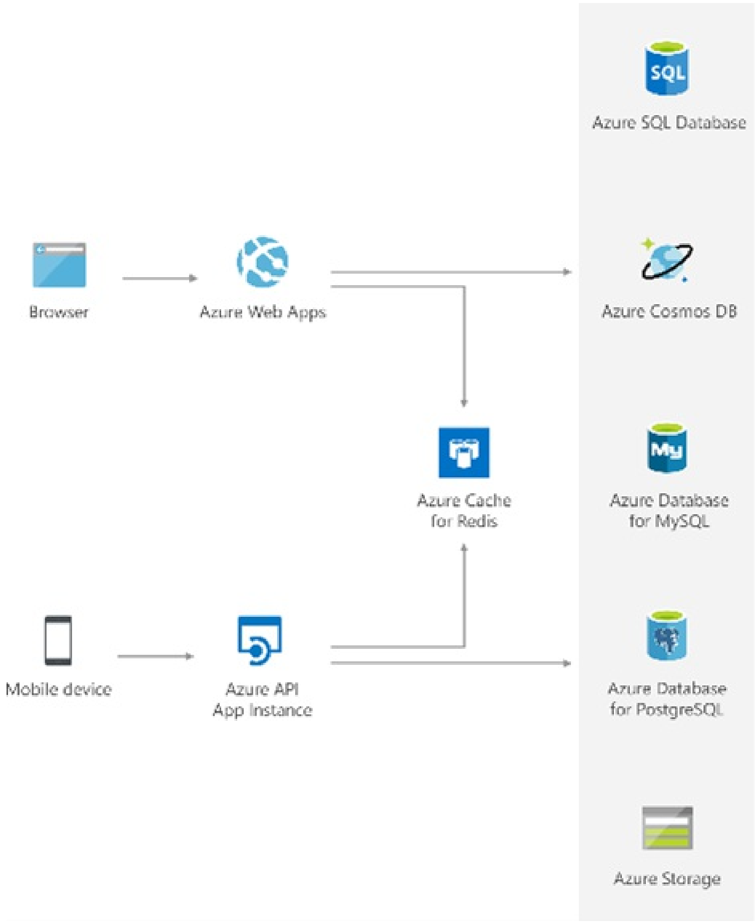


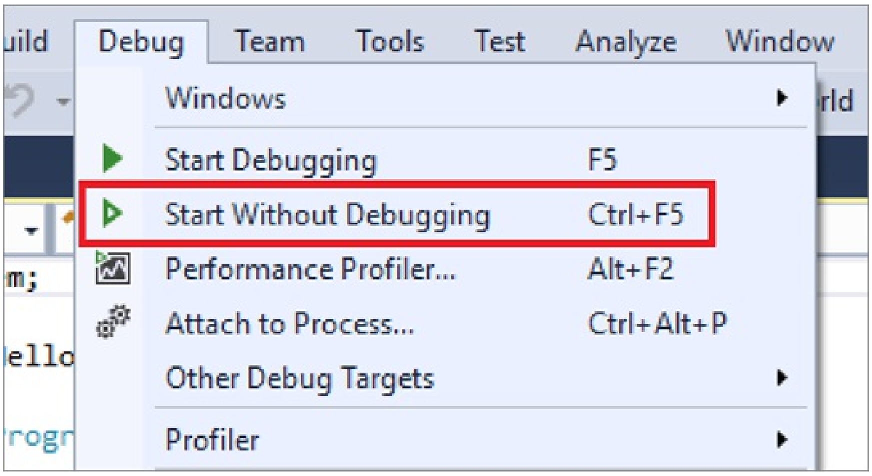
Diagram showing a typical caching architecture using an Azure Cache for Redis in a web application. The architecture has three layers: the client-access layer and the data tier with an Azure Cache for Redis between them. The client layer shows the access pattern for two types of clients: customers using a web browser interact with Azure Web Apps while customers on a mobile device interact with an Azure API App Instance. Both clients pull data from the Azure Cache for Redis when the needed data is in the cache. If the data isn't in the cache, the clients query the data tier directly. The data tier contains five storage locations: Azure SQL Database, Azure Cosmos DB, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Storage.

GUIDELINE	EXAMPLE

GUIDELINE	EXAMPLE
Provide both alternative text and a long description	Appropriate alt text: *"Diagram of a caching architecture with Azure Cache for Redis positioned between the client-access layer and the data tier."* Appropriate long description: *"Diagram showing a typical caching architecture using an Azure Cache for Redis in a web application. The architecture has three layers: the client-access layer and the data tier with an Azure Cache for Redis between them. The client layer shows the access pattern for two types of clients: customers using a web browser interact with Azure Web Apps while customers on a mobile device interact with an Azure API App Instance. Both clients pull data from the Azure Cache for Redis when the needed data is in the cache. If the data isn't in the cache, the clients query the data tier directly. The data tier contains five storage locations: Azure SQL Database, Azure Cosmos DB, Azure Database for MySQL, Azure Database for PostgreSQL, and Azure Storage."*

Screenshots of user interface

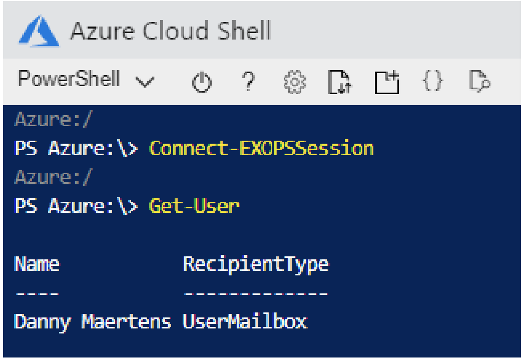
Consider the following image:



GUIDELINE	EXAMPLE
Begin with *"Screenshot of"*	*"**Screenshot of** the Visual Studio Debug menu. The menu entry titled Start Without Debugging and its keyboard shortcut Ctrl+F5 are highlighted."*
Identify the product	*"Screenshot of the **Visual Studio** Debug menu. The menu entry titled Start Without Debugging and its keyboard shortcut Ctrl+F5 are highlighted."*
Identify highlighted areas	*"Screenshot of the Visual Studio Debug menu. The menu entry titled **Start Without Debugging** and its keyboard shortcut Ctrl+F5 are highlighted."*
Identify keyboard shortcuts	*"Screenshot of the Visual Studio Debug menu. The menu entry titled Start Without Debugging **and its keyboard shortcut Ctrl+F5** are highlighted."*
Include the location of UI elements, state of UI controls, and any relevant values in data-entry fields	*"Screenshot of the Visual Studio **Debug menu**. The menu entry titled Start Without Debugging and its keyboard shortcut Ctrl+F5 are highlighted."*

Screenshots of command-line interface

Consider the following image:



Screenshot of the Azure Cloud Shell. The screenshot shows the PowerShell environment being used to execute two commands. The first command is Connect-EXOPSSession, which launches a session. This command has no output. The second command is Get-User, which yields basic information about the current user. The output shown for this command is formatted as a table: the first row lists the column headers of Name and RecipientType. The second row consists of dashes used as a separator between the column headers and the data. The third row gives the user information, the value in the Name column is Danny Maertens, and the RecipientType is UserMailbox.

GUIDELINE	EXAMPLE
Provide both alternative text and a long description	Appropriate alt text: *"Screenshot of the Azure Cloud Shell using the PowerShell environment to launch a session and execute the Get-User command."* Appropriate long description: *"Screenshot of the Azure Cloud Shell. The screenshot shows the PowerShell environment being used to execute two commands. The first command is Connect-EXOPSSession, which launches a session. This command has no output. The second command is Get-User, which yields basic information about the current user. The output shown for this command is formatted as a table: the first row lists the column headers of Name and RecipientType. The second row consists of dashes used as a separator between the column headers and the data. The third row gives the user information, the value in the Name column is Danny Maertens, and the RecipientType is UserMailbox."*
The long description should include all relevant text shown in the screenshot	Appropriate long description: *"Screenshot of the Azure Cloud Shell. The screenshot shows the PowerShell environment being used to execute two commands. The first command is* ***Connect-EXOPSSession****, which launches a session. This command has no output. The second command is* ***Get-User****, which yields basic information about the current user. The output shown for this command is formatted as a table: the first row lists the column headers of* ***Name*** *and* ***RecipientType****. The second row consists of* ***dashes*** *used as a separator between the column headers and the data. The third row gives the user information,* ***the value in the Name column is Danny Maertens, and the RecipientType is UserMailbox****."*

GUIDELINE	EXAMPLE
The long description should differentiate between each command and the output of those commands	Appropriate long description: *"Screenshot of the Azure Cloud Shell. The screenshot shows the PowerShell environment being used to execute two commands. The* ***first command is Connect-EXOPSSession****, which launches a session. This command* ***has no output****. The* ***second command is Get-User****, which yields basic information about the current user. The* ***output shown for this command is*** *formatted as a table: the first row lists the column headers of Name and RecipientType. The second row consists of dashes used as a separator between the column headers and the data. The third row gives the user information, the value in the Name column is Danny Maertens, and the RecipientType is UserMailbox."*

Close caption requirements for video

Videos require closed captions:

- Writers must make sure that videos they add to content include closed captions. Most Channel 9 and Red Tiger videos already include closed captions. (Red Tiger is a Microsoft publishing system for videos. It's the preferred system because it supports both accessibility and localization.)
- Writers should contact Channel 9 if closed captions are missing. Channel 9 can usually supply them in a few days.

For more information, see Create and publish a video.

Playback control requirements for animated GIFs

Animated GIFs require a pause before autoplay. This pause capability is in place on docs.microsoft.com (Docs). Writers should make sure the pause is working after their article is published.

GDPR references in new or existing technical documentation articles

5/10/2021 • 9 minutes to read

Technical documentation cannot use the terms "General Data Protection Regulation" or "GDPR" outside the context of the CELA-approved includes listed in this article.

General Data Protection Regulation (GDPR) is about protecting and enabling the privacy rights of individuals in the European Union (EU). GDPR establishes strict global privacy requirements governing how organizations manage and protect personal data while respecting individual choice - no matter where data is sent, processed, or stored.

Enforcement of this regulation begins on May 25, 2018. Organizations that fail to comply, whether intentionally or unintentionally, can be fined up to €20 million or 4% of the company's global annual turnover of the previous financial year, whichever is higher based on the type of infraction incurred.

All new GDPR-related content requires frontline CELA review and approval before publication. To find your appropriate CELA representative, see: https://fyc-prod.azurewebsites.net/. We also recommend asking your product group to check with their frontline CELA for review and sign-off, as needed.

Mandatory GDPR guidance for all articles

The following are mandatory guidelines that apply to any feature or functionality impacted by the GDPR requirements.

You must NEVER:

- Interpret the GDPR regulations (by saying things like, "Article 20 of the GDPR says...")
- Imply or promise any action will lead to compliance (by saying things like, "Turning off this feature will keep you compliant..."
- Imply or promise that Microsoft will be compliant. Microsoft will be, but those commitments need to come from contracts and formal statements, not the documentation.
- Use the term GDPR in file names, titles, introduction, or procedural text. All general discussions about GDPR, data deletion through the portal, and breach notification information should be handled through the Service Trust Portal and Compliance sites. Not using this term helps to avoid potential legal implications and future maintenance of content.
- Duplicate content. If content already exists about this feature/functionality, you should link to the existing information.
- Use Microsoft internal data classifications, such as EUII, EUPI, PII, customer content, or telemetry. Instead, you should use the terms: personal data, customer data, customer-authored data, and system-generated logs.

Where does the GDPR content appear?

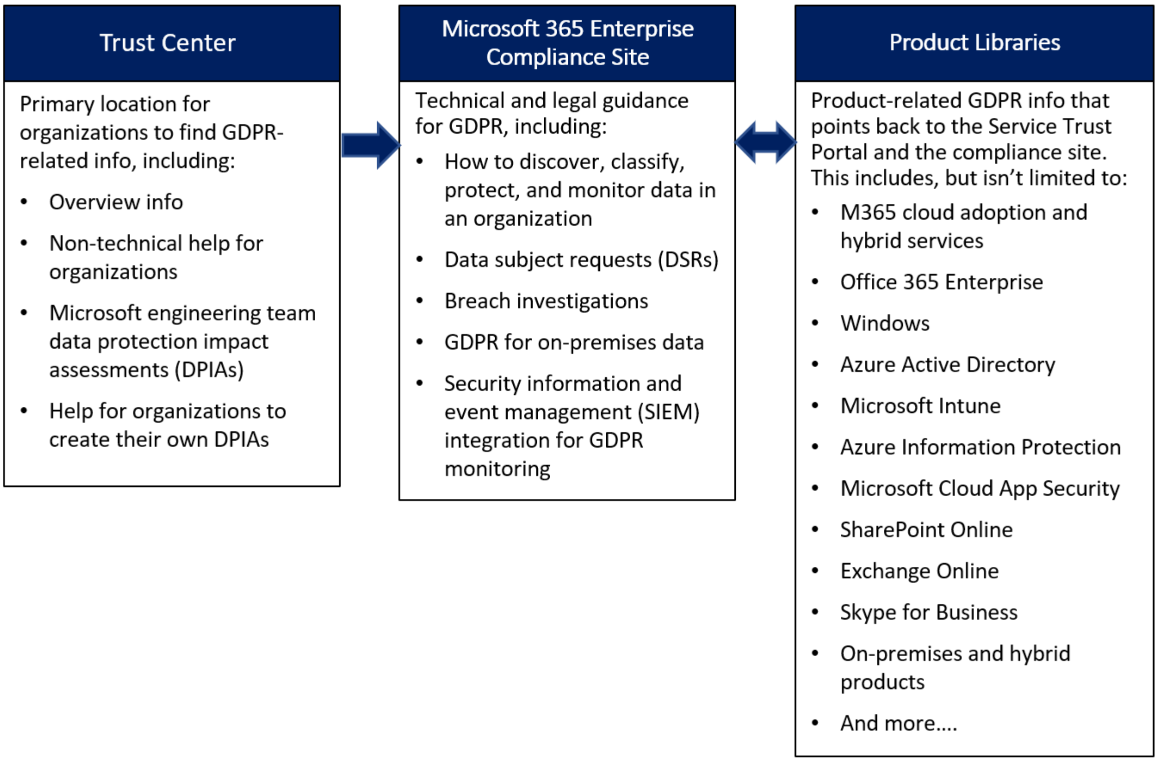
At a high-level, this diagram points to where various GDPR content is located:

- Microsoft Trust Center. Includes general GDPR information, along with information about GDPR compliance best practices, and documentation helpful to your GDPR accountability, such as Data Protection Impact Assessments (DPIAs), Data Subject Requests (DSRs), and data breach notification. This site is targeted towards business decision makers and marketing, to determine the compliance capabilities of our products.

- [Microsoft 365 Enterprise GDPR Compliance site](). Includes legal and technical guidance, based on product groups, pointing to the individual product library articles for more information.

- [Service Trust portal](). Includes all the global GDPR and all-up Microsoft information, pointing into the Microsoft 365 Enterprise Compliance site and the individual product library articles for more information. This site requires authentication to view, and is targeted to technical analyst, auditors, and compliance people.

- **Product libraries (various).** Provides the specific details about features and functionalities, pointing into both the Service Trust portal and the Microsoft 365 Enterprise Compliance site.



Add GDPR info and Include references based on scenario

Typically, the product group or a PM will contact you to let you know that a feature or functionality is impacted by GDPR. This notification means that you'll need to add content about how to export and delete a user's personal data. Currently there are three scenarios:

Scenario 1: Delete and/or export customer data and system-generated logs through the Azure portal

Any GDPR-impacted feature or functionality, where the delete and export of system-generated logs happens through the Azure portal. This scenario includes any feature or functionality that uses the primary Azure portal delete or export functionality (https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-azure#step-5-delete and https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-azure#step-6-export). Anything that is deleted or exported in a different way should be considered “in-product” for both instructions and the Include reference.

You must:

- Document the feature or functionality in place, where you would talk about the feature or functionality normally. However, since the feature or functionality uses the global Azure portal delete and export functionality, you should point to the Microsoft Trust Center and the Service Trust Portal for instructions instead of rewriting the steps in your article.

- Add the following Include file (azure-docs-pr\includes\gdpr-dsr-and-stp-note.md) to the end of your introduction paragraph.

Example of Include in markdown: `[!INCLUDE [GDPR-related guidance](../../includes/gdpr-dsr-and-stp-note.md)]` with the `../` references matching the hierarchy of your article.

Include file text (CELA-approved):

> **NOTE**
>
> For information about viewing or deleting personal data, see Azure Data Subject Requests for the GDPR. For more information about GDPR, see the GDPR section of the Microsoft Trust Center and the GDPR section of the Service Trust portal.

Scenario 2: Delete and/or export customer data and system-generated logs through an in-product solution

Any GDPR-impacted feature or functionality, where the delete and export of system-generated logs and customer data happens in-product, using the feature or functionality itself. This scenario includes any feature or functionality that uses anything outside of the global Azure portal delete or export functionality (https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-azure#step-5-delete and https://docs.microsoft.com/microsoft-365/compliance/gdpr-dsr-azure#step-6-export), including any feature or functionality in the Azure portal that uses a different delete or export process.

You must:

- Document the feature or functionality in place, where you would talk about the feature or functionality normally.
- Add the following Include file (azure-docs-pr\includes\gdpr-intro-sentence.md) to the end of your introduction paragraph.

  Example of Include in markdown: `[!INCLUDE [GDPR-related guidance](../../includes/gdpr-intro-sentence.md)]` with the `../` references matching the hierarchy of your article.

  Include file text (CELA-approved):

  > **NOTE**
  >
  > This article provides steps about how to delete personal data from the device or service and can be used to support your obligations under the GDPR. For general information about GDPR, see the GDPR section of the Microsoft Trust Center and the GDPR section of the Service Trust portal.

Scenario 3: Deleting customer data requires deleting an underlying Windows component

Any GDPR-impacted feature or functionality, where the user is asked to delete the Windows event log or any other underlying Windows component, to delete customer data.

You must:

- Update an existing feature or functionality article to point users to the Microsoft corporate policy about this type of data removal. The official statement will appear in the Compliance Manager, with secondary content appearing in the M365 Enterprise compliance site (both statements are going to appear on 5/25). This scenario appears to be hybrid-only at this point.
- Add the following Include file (azure-docs-pr\includes\gdpr-hybrid-note.md) to the end of your introduction paragraph.

  Example of Include in markdown: `[!INCLUDE [GDPR-related guidance](../../includes/gdpr-hybrid-note.md)]` with the `../` references matching the hierarchy of your article

Include file text (CELA-approved):

> **NOTE**
> For information about viewing or deleting personal data, please review Microsoft's guidance on the Windows data subject requests for the GDPR site. For general information about GDPR, see the GDPR section of the Microsoft Trust Center and the GDPR section of the Service Trust portal.

Use multiple Include references together

If your feature or functionality uses both the global Azure portal and an in-product delete or export functionality, you should use both Include references:

- **Global Azure portal reference (gdpr-dsr-and-stp-note.md).** Use this reference first, after your introduction to the entire article.
- **In-product reference (gdpr-intro-sentence.md).** Use this reference second, after the introduction to the procedural section of the article.

Add includes to other repos

For repos other than azure-docs-pr, create an identical set of include files in the repo and use them as needed in that repo's content. In the gdpr-dsr-and-stp-note.md file, change the Azure Data Subject Requests for the GDPR link to point to the DSR page for your product area. Contact Liz Ross to let her know you're using the include files in your repo (in case something needs to change) or if you have any questions about their use.

Recommendations for writing about personal data

Within individual articles, you should think about covering features and functionality, based on the following types of information.

- **Inventory.** Include information about what personal data exists and where, along with any steps about how to use the feature or functionality to search for and identify personal data.
- **Protect.** Include detailed instructions or links to existing articles about how customers can protect their personal data. This content can also cover how the feature or functionality helps to facilitate data classification to help satisfy customer data subject requests, along with how to use the feature or functionality to display a custom privacy statement.
- **Respond.** Include information about any resources the customer can use to respond to data subject requests. This content should include step-by-step information about how to use the in-product feature or functionality to update personal data, to delete personal data, or how to export personal data. See the Delete and/or export customer data and system-generated logs through the Azure portal and the Delete and/or export customer data and system-generated logs through an in-product solution sections of this article.
- **Audit and reporting.** Include information and relevant links about how a customer can verify they're providing the proper information for data subject requests. This content can include any auditing or reporting features or functionality, which the customer can use to show compliance with data subject request fulfillment.

Exceptions

If you believe you need an exception to this guidance for Content & Learning-sponsored content, you must obtain written approval. Contact lizross for an initial review of your exception. They will determine if additional Azure Operations or CELA review is required.

Pull request review and GDPR mentions

In Content & Learning repositories covered by PRmerger and the PR review process, all pull requests that contain content that mentions GDPR are flagged by PRmerger for human review. This includes articles where the updates in the pull request do not affect the sections of the article that mention GDPR.

The PR review team will review the new or updated parts of the article for compliance with the GDPR content guidelines described in this article. If the new or updated parts of the article do not appear to align to the guidance, the PR review team will send mail to:

- The pull request author
- The content manager for the Azure service or the business approver for the repository (outside Azure)

Subject: GDPR review needed for PR

In the body, the reviewer needs to clearly list the file that has an issue and clearly describe or show how the article does not align to the guidance.

The review must include this text in the message - some authors just need a pointer to bring their content into alignment:

> The CELA-approved guidance for our tech docs is published here - https://review.docs.microsoft.com/en-us/help/contribute/contribute-add-gdpr-references?branch=master

The reviewer closes with text that allows the author to explain possible reasons that might justify an exception. If no justification exists, the author needs to fix the article to align to the guidance:

> If you have reasons or legal approvals for not aligning with the guidance, let the folks on this thread know. Otherwise, the article should be brought into alignment with the GDPR content guidance.

Terminology and inclusive language

3/5/2021 • 6 minutes to read

These guidelines address inclusive language in content. They are part of a long-term effort to make our content more inclusive and will be added to over time. The branch name change in GitHub is a separate effort but part of the same commitment.

The guidelines were reviewed across Microsoft: Global D+I, Global Readiness (CELA), Cloud Style Guide, C+L term leads, and members of marketing, engineering, and localization. We also try to align with partners in the industry.

Inclusive language guidelines

The guidelines recommend specific term updates while allowing for some flexibility, depending on context. The goal is consistency throughout the user experience. The note included in the guidelines for references to *slave* in content or code we can't change should be used in technical documentation. The guidelines should otherwise be applied across Microsoft.

TERM	STATUS	OPS VALIDATION STATUS	GUIDELINES
master (git default branch)	Ready to deploy	Enabled	GitHub (which includes the Azure DevOps team) is changing the name of the default repo branch name from master to main.
master/detail	In review	Not enabled	Master/detail refers to a control or control pattern that combines a list with a details view, such as the mail list and mail view in Outlook. Replacements under consideration: list/detail, collection/detail, parent/detail.

TERM	STATUS	OPS VALIDATION STATUS	GUIDELINES
master	Ready to deploy	Not enabled	Microsoft recognizes that the term *master* has numerous meanings and is used in various contexts. Don't use *master* in a master/slave context: **Examples that must be changed**: - *master server* (which implies a slave client). Change to *primary server* or other appropriate term. - *master in, slave out* (for hardware signal names). Change to *serial data in/out* or other appropriate term. **Examples that are OK, where master describes proficiency or an original**: - *master list*, *master slide*, *master plan*, *master detail*, etc. Teams can consider alternatives to *master* in these examples, such as *primary*, *main*, or *template*, but make the change consistently throughout the experience.
slave	Ready to deploy	Enabled (Suggestion)	Use *replica* or alternatives such as *secondary*, *subordinate*, *worker*, *target*, etc., depending on context. - Don't use *slave* in new code or content. - For legacy or third-party content or code that we can't change, add this note before the reference in documentation: **Note**: This article contains references to the term *slave*, a term that Microsoft no longer uses. When the term is removed from the software, we'll remove it from this article.

TERM	STATUS	OPS VALIDATION STATUS	GUIDELINES
master/slave	Ready to deploy	Not enabled	Use *primary/replica* or alternatives such as *primary/secondary*, *primary/subordinate*, *principal/agent*, *parent/child*, *controller/worker*, etc., depending on context. - Don't use *master/slave* in new code or content. - For legacy or third-party content or code that we can't change, add this note before the reference in documentation: **Note**: This article contains references to the term *slave*, a term that Microsoft no longer uses. When the term is removed from the software, we'll remove it from this article.
whitelist	Ready to deploy	Enabled (Suggestion)	Use *allowlist* as the general reference. More specific usage like *safelist* can be used in the appropriate context. - *allowlist* is one word.
blacklist	Ready to deploy	Enabled (Suggestion)	Use *blocklist* as the general reference. More specific usage like *blocked senders list* can be used in the appropriate context. - *blocklist* is one word.

Here is example markdown for the note you can use if needed:

```
>[!NOTE]
> This article contains references to the term *slave*, a term that Microsoft no longer uses. When the term
is removed from the software, we’ll remove it from this article.
```

Specific uses of master that we're checking

To minimize false positives, the docs build system checks for very specific uses of the word *master*:

```
master branch
*master* branch
**master** branch
checkout master
checkout *master*
checkout **master**
checkout@master
pull master
pull *master*
pull **master**
push master
push *master*
push **master**
fetch master
fetch *master*
fetch **master**
master server
*master* server
**master** server
master target server
*master* target server
**master** target server
branch: [master]
git pull azure master
git pull azure *master*
git pull azure **master**
git push azure master
git push azure *master*
git push azure **master**
git fetch azure master
git fetch azure *master*
git fetch azure **master**
git pull origin master
git pull origin *master*
git pull origin **master**
git push origin master
git push origin *master*
git push origin **master**
git fetch origin master
git fetch origin *master*
git fetch origin **master**
```

Request an exemption

To request an exemption, send mail to terminologytaskforce@microsoft.com. Let them know the term, the context, the repo, and the paths of the files affected.

Terminology tools

We're developing new tools and updating existing tools to help with terminology updates.

TOOL	STATUS	DESCRIPTION
Sensitive term checking via OPS builds	Deployed	
Multi-search-and-replace	Available	Uses a configuration file to replace multiple terms. To use the tool on your repo, please contact Alex Buck (abuck).

TOOL	STATUS	DESCRIPTION
TermFixer for links	Available	A command-line tool Mike Jacobs developed, looks for GitHub links in your articles. For each one it finds, it queries GitHub to check whether the default branch has been renamed yet. If it has, it updates the link to point to the new default branch name. TermFixer usage and installation instructions
Acrolinx	Update pending	The Acrolinx tool is being updated with the latest term guidelines.
Policheck	Update pending	The Policheck terms list is being updated with the latest term guidelines.

Frequent asked questions (FAQ)

Q: Can I go ahead and replace sensitive terms in my content right now?

A: Yes! Just make sure there's an approved replacement. If the term's status is "Ready to deploy," you can make the replacement. If the status is "In review," hold off for now.

Q: What's our deadline?

A: We'd like to get these terms fixed as soon as possible. That said, there are a lot of them in a lot of repos. Please make fixing these terms a part of your Q3 planning. As we get closer to fixing all the terms, we'll set a deadline for turning the sensitive-language build validations from suggestions into warnings.

Q: I found a term that's not on the list... what should I do?

A: Send mail to terminologytaskforce@microsoft.com and tell us about the term and where it appears.

Q: My Doc repo's default branch is named "master"--when I can change it?

A: The Docs build system now supports "main" as a default branch. Khairun Jamal's team has developed tooling to automate changing the default branch from "master" to "main" in our publishing repos. Please wait until you hear from Khairun before changing the default branch name in your Docs repos.

If you need to change a default branch name before that, reach out to terminologytaskforce@microsoft.com.

Q: I found a sensitive term in my content that is actually in the product, how do I fix it?

A: If you are the content owner for the product, reach out to the usual engineering or PM contact on the product team. Provide a link to the page https://aka.ms/terminologytaskforce, that explains the goal and scope of the project, as well as guidance for changing the terms. Also include the link to the content the term was found in, for reference, and a description of where in the product the term is located, if possible. For any further questions, feel free to direct them to terminologytaskforce@microsoft.com.

If you don't know the product owner, email terminologytaskforce@microsoft.com for help.

See also

Here are other sources for additional information on the change to *master*:

- GitHub guidance for renaming branches
- Open Source Hardware Association
- Wired

- ZDNet re: MySQL
- ZDNet re: GitHub
- ZDNet re: Linux
- Drupal.org
- Washington Post

Legal content

3/5/2021 • 2 minutes to read

Documentation in large repos such as azure-docs-pr and sql-docs-pr is subject to machine translation, public feedback, and automatic merging of internal contributions if the changes are within the PR Merger threshold. This can pose a risk to the company if legal or policy documents are incorrectly translated or unintentionally modified.

To mitigate this risk, we isolate all content with potential legal implications in a private, locked-down repo: MicrosoftDocs/DocsLegal-pr. This content must receive applicable CELA reviews and follow a special localization process. Each product team is responsible for maintaining the integrity of content in their subfolder of this repo.

NOTE

Before you add content to the DocsLegal-pr repo, review existing legal and licensing webpages to see if they apply to your scenario. If you determine you need additional documentation, consult your CELA representative to confirm the need.

File and folder requirements

Be aware of these requirements before you contribute to the DocsLegal-pr repo:

- Individual product teams are responsible for completing necessary CELA reviews and updates.
- Individual product teams are responsible for confirming localization needs with CELA, and then coordinating translations and publication to localized repos. This includes funding those translations. See Localization of legal content for detailed instructions.
- Each product or service must create and use a subfolder in the repo.
- Filenames must be lowercase with hyphen separators.
- All files must include the required metadata for Developer Relations. The DocsLegal-pr repo uses the Developer Relations rule set to enforce these values.
  - If your file is not product or service specific, use `ms.prod: legal`
  - All files should use the generic topic type `ms.topic: article`

We recommend that all files use an additional metadata value: `layout: ContentPage`. This creates a flat-page experience with no TOC. The exception to this is groups who want to create a contextual TOC to make legal articles appear within the TOC of a product or service guide. *Do not* create a regular TOC within the DocsLegal-pr repo: the root TOC and breadcrumbs are out of context and will confuse readers.

Publication process

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

1. Draft your Markdown topic in a forked copy of MicrosoftDocs/DocsLegal-pr. Refer to the requirements above.

2. Create a pull request (PR) to the master branch of the upstream repo.

3. Review the staged pages from the **Preview URL** links in the PR and ensure the content is equal to the CELA-approved document. Complete any other reviews your team requires. For example, some product

groups have their CELA representatives review staged content.

4. Confirm that the PR built successfully, address any validation errors or warnings, then add the `#sign-off` comment if you are ready for the content to go live.

    - If you require special (timed) or rushed publication, do not #sign-off. Instead, add a comment to the PR and explain the request.

5. Content Production Services (CPS) monitors the repo for new PRs, and will comment if the PR is not ready to be merged. You must address these comments.

6. CPS publishes all merged PRs to the live site once a day.

After your content goes live in the main (en-us) repo, you can start to coordinate your localization requests, if needed. It is important that the English content is published **before** you move forward with updates to translated content.

Use PoliCheck to identify sensitive language

11/2/2020 • 2 minutes to read

Policheck is a content-scanning tool designed to check for sensitive geopolitical terms, profanity, and trademark terms in Microsoft products.

Learn more: https://microsoft.sharepoint.com/sites/globalreadiness/SitePages/PoliCheck.aspx

When to use PoliCheck

- During localization
- As an IC, if you want immediate scan results on a repo

> **NOTE**
>
> As of July 2020, the Content Production Services team is phasing into a pilot where they run Policheck on this scope and frequency.

Prerequisites

1. Install PoliCheck from http://toolbox/policheck.
2. Ensure that your local git repository is up to date.
3. Verify that you are in the repository that you want to test in GitBash.

Run PoliCheck

You can run PoliCheck against content in a GitHub repository. You can scan an entire repository, a folder, or a single markdown file. Or, repository administrators can periodically run PoliCheck on the local copy of a repo.

1. Launch PoliCheck.
2. Click **Options** > **File Type Settings**.
3. In the **File Type Extensions** box, make sure **Pure Text Files** is selected. The markdown extension (.md) is part of this group of file extensions.
4. Close the File Type Settings dialog box - if you are prompted to save, click **Yes**.
5. Click **Task** > **New**, and then select **File Scan** in the **Task Type** dialog box.
6. On the **Step 1** tab, name the task. Then, if you want to test a folder in your Git repository, select Folder in the Target Area. If you want to test a specific file, select File in the Target area. Specify the location in which you want to save the results. Click **Next**.
7. On the **Step 2** tab, select **English**, all severities, and all term classes. Click **Next**.
8. On the **Step 3** tab, select **Default File**, and click **OK**.
9. In the Policheck console, right-click the scan task you just created, and then click **Start**.

Policheck displays the results in the right pane. To export the results to an XML file that you can then open in Excel, click **Actions** > **Export Scan Results** > **File**.

Rerun PoliCheck

To run the same task again, right-click the existing task and select `Start`.

Contributor guide operating principles

5/10/2021 • 2 minutes to read

We believe our contributor guides are content that deserve the same care and attention we give to all our work. Our decisions on what to include and how it's organized are informed by the principles of **Find, Trust, Transfer**. These principles are intentionally oriented from the guide consumer's point-of-view. We want contributors to quickly find the right article, trust that it's up to date, and apply what they learn to their own writing.

Find

We have the right content and it's easy to navigate.

- We reuse and link content, we don't duplicate.
- No more than three layers deep in the TOC where possible.
- Emphasize consistency in organization and format.
- When a user finds a gap in the guidance it's easy to fill that gap, and get approval for the change.
- It's clear what each set of guidance does and doesn't apply to. By default, guidance applies globally unless otherwise indicated.

Trust

Our content is kept up to date and shows approved procedures. Errors are fixed quickly.

- Anyone can suggest a change to an article, but page owners approve or get approval for changes as needed.
- Ensure appropriate notice to users before making substantive changes. Make clear to those contributors who propose changes what the response timeline (SLA) is likely to be.
- Clearly distinguish between recommended guidelines and required rules.
- Clearly distinguish which organizations or types of content a set of guidelines applies to.
- We have clear, current contacts for each article.

Transfer

We include a mix of procedures, concepts, and examples to enable our consumers to apply the ideas to their own work.

- An article is for one user doing one thing.
- Focus on the task and support it with necessary conceptual information and examples.
- Document the preferred/supported way to do something, not all the possible options.

Contributor guides content standards

3/5/2021 • 2 minutes to read

The content on docs.microsoft.com should speak with one voice, but different situations call for different conventions. Striking a balance between speaking with one voice and speaking to different communities requires clear decision-making processes. By establishing content standards that scope responsibility to specific organizational levels, we maintain consistency and avoid revisiting the same decisions eternally.

The goal of the contributor guide content standards is the consistent appearance and function of internal-facing contributor guidance across content sets and modalities. This consistency standardizes the experience from one content set to another.

Federated content standards

The approach uses a government analogy to distinguish three hierarchical levels for implementing content standards across the docs.microsoft.com platform: federal, state, and local.

Federal level

Standards at the federal level apply to all users of the docs.microsoft.com platform.

Federal: platform

State level

Standards at the state level support the ways that individual business organizations use the platform. The guidance for one organization wouldn't necessarily be the same as for another since the audiences may differ in their needs.

State: business org

Local level

Standards at the local level apply to variations for a service/product. For example, this level of guidance aligns with the "repo-specific" section of the Docs contributor guide TOC.

Local: product/service

Templates and style guides

Style guides help keep a content set consistent when there are different products, contributors, and editors. They provide distributed governance for things like terminology, grammar, and usage (example: footnotes).

Templates help keep a content set consistent by providing a pre-patterned starting point for a specific content format.

Both tools support consistency and reduce the decision-making load on contributors when creating content.

List of available contributor guides

6/9/2021 • 2 minutes to read

C&L

GUIDE	CONTACT	GITHUB ALIAS
Docs contributor guide (Internal)	Jim Parker	Jim-Parker
Docs contributor guide (External)	Jim Parker	Jim-Parker
Learn contributor guide	Jim Parker	Jim-Parker
Repo admin guide	Jim Parker	Jim-Parker
Localization guide	Jason Card	JasonCard
Tools & process guide	Jim Parker	Jim-Parker
Content product service (CPS) guide		

Other

GUIDE	CONTACT	GITHUB ALIAS
Apex Publishing Desk Documentation		
Applied AI Docs	Erik Hopf	erhopf
Azure Databricks		
Business Applications Content Experience Team (BACX) guide		
Cloud Studio		
Contributor guide for Azure Data Explorer (Kusto)		
Dev Div		
Developer relations team guide		
Docs style guide		
Engineering		
Engineering: new-hope		

GUIDE	CONTACT	GITHUB ALIAS
Experimentation platform		
DevRel UX	Adam Kinney	
Event Learning Experience		
MARVEL Authoring Guide for Docs		
Microsoft open source policies & strategy guide		
Microsoft open source program tools guide		
PowerShell Contributor Guide	Sean Wheeler	sdwheeler
Product Launch Readiness (PLR) guide		
Windows authoring guide		
World Wide Learning (WWL) guide		
PLR Launch Criteria		

Old (now 404)

- https://review.docs.microsoft.com/en-us/opsdocs/?branch=master
- https://review.docs.microsoft.com/en-us/learn-live/docs/general-workitem-tracking?branch=master