Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Update-AzDataLakeGen2Item'

PS:\>Get-HELP Update-AzDataLakeGen2Item -Full

NAME

Update-AzDataLakeGen2Item

SYNOPSIS

Update a file or directory on properties, metadata, permission, ACL, and owner.

SYNTAX

Update-AzDataLakeGen2Item [-FileSystem] <System.String> [-Acl <Microsoft.WindowsAzure.Commands.Storage.Model.ResourceModel.PSPathAccessControlEntry[]>] [-Context <Microsoft.Azure.Commands.Common.Authentication.Abstractions.IStorageContext>] [-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Group <System.String>] [-Metadata <System.Collections.Hashtable>] [-Owner <System.String>] [-Path <System.String>] [-Permission <System.String>] [-Property <System.Collections.Hashtable>] [-Confirm] [-WhatIf] [<CommonParameters>]

Update-AzDataLakeGen2Item [-Acl <Microsoft.WindowsAzure.Commands.Storage.Model.ResourceModel.PSPathAccessControlEntry[]>] [-Context <Microsoft.Azure.Commands.Common.Authentication.Abstractions.IStorageContext>] [-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Group

<System.String>] -InputObject

<Microsoft.WindowsAzure.Commands.Common.Storage.ResourceModel.AzureDataLakeGen2Item> [-Metadata <System.Collections.Hashtable>] [-Owner <System.String>] [-Permission

<System.String>] [-Property <System.Collections.Hashtable>] [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The Update-AzDataLakeGen2Item cmdlet updates a file or directory on properties, metadata, permission, ACL, and owner. This cmdlet only works if Hierarchical Namespace

is enabled for the Storage account. This kind of account can be created by run "New-AzStorageAccount" cmdlet with "-EnableHierarchicalNamespace $true".

PARAMETERS

-Acl <Microsoft.WindowsAzure.Commands.Storage.Model.ResourceModel.PSPathAccessControlEntry[]>

Sets POSIX access control rights on files and directories. Create this object with New-AzDataLakeGen2ItemAclObject.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Context <Microsoft.Azure.Commands.Common.Authentication.Abstractions.IStorageContext>

Azure Storage Context Object

Required? false

Position? named

Default value None

Accept pipeline input? True (ByPropertyName, ByValue)

Accept wildcard characters? false

-DefaultProfile <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>

The credentials, account, tenant, and subscription used for communication with Azure.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-FileSystem <System.String>

FileSystem name

Required? true

Position? 0

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Group <System.String>

Sets the owning group of the blob.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InputObject <Microsoft.WindowsAzure.Commands.Common.Storage.ResourceModel.AzureDataLakeGen2Item>

Azure Datalake Gen2 Item Object to update

Required? true

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Metadata <System.Collections.Hashtable>

Specifies metadata for the directory or file.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Owner <System.String>

Sets the owner of the blob.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Path <System.String>

The path in the specified Filesystem that should be updated. Can be a file or directory In the format 'directory/file.txt' or 'directory1/directory2/'. Not

specify this parameter will update the root directory of the Filesystem.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Permission <System.String>

Sets POSIX access permissions for the file owner, the file owning group, and others. Each class may be granted read,

write, or execute permission. Symbolic

(rwxrw-rw-) is supported. The sticky bit is also supported and its represented either by the letter t or T in the final character-place depending on whether the

execution bit for the others category is set or unset respectively, absence of t or T indicates sticky bit not set.Invalid in conjunction with ACL.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Property <System.Collections.Hashtable>

Specifies properties for the directory or file. The supported properties for file are: CacheControl, ContentDisposition, ContentEncoding, ContentLanguage,

ContentMD5, ContentType. The supported properties for directory are: CacheControl, ContentDisposition, ContentEncoding, ContentLanguage.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Confirm <System.Management.Automation.SwitchParameter>

Prompts you for confirmation before running the cmdlet.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-WhatIf <System.Management.Automation.SwitchParameter>

Shows what would happen if the cmdlet runs. The cmdlet is not run.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

<CommonParameters>

This cmdlet supports the common parameters: Verbose, Debug,

ErrorAction, ErrorVariable, WarningAction, WarningVariable,

OutBuffer, PipelineVariable, and OutVariable. For more information, see

about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.String

Microsoft.WindowsAzure.Commands.Common.Storage.ResourceModel.AzureDataLakeGen2Item

Microsoft.Azure.Commands.Common.Authentication.Abstractions.IStorageContext

OUTPUTS

Microsoft.WindowsAzure.Commands.Common.Storage.ResourceModel.AzureDataLakeGen2Item

NOTES

Example 1: Create an ACL object with 3 ACL entry, and update ACL to all items in a Filesystem recursively

```
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType user -Permission rwx
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType group -Permission rw- -InputObject $acl
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType other -Permission "rwt" -InputObject $acl
Get-AzDataLakeGen2ChildItem -FileSystem "filesystem1" -Recurse | Update-AzDataLakeGen2Item -ACL $acl
```

FileSystem Name: filesystem1

Path	IsDirectory	Length	LastModified	Permissions	Owner	Group
dir1	True		2020-03-13 13:07:34Z	rwxrw-rwt	$superuser	$superuser
dir1/file1	False	1024	2020-03-23 09:29:18Z	rwxrw-rwt	$superuser	$superuser
dir2	True		2020-03-23 09:28:36Z	rwxrw-rwt	$superuser	$superuser

This command first creates an ACL object with 3 acl entry (use -InputObject parameter to add acl entry to existing acl object), then get all items in a filesystem and
update acl on the items.

-- Example 2: Update all properties on a file, and show them --

```
$file = Update-AzDataLakeGen2Item -FileSystem "filesystem1" -Path "dir1/file1" `
        -Acl $acl `
                -Property @{"ContentType" = "image/jpeg"; "ContentMD5" = "i727sP7HigloQDsqadNLHw==";
"ContentEncoding" = "UDF8"; "CacheControl" = "READ";
"ContentDisposition" = "True"; "ContentLanguage" = "EN-US"} `
        -Metadata @{"tag1" = "value1"; "tag2" = "value2" } `
        -Permission rw-rw-rwx `
        -Owner '$superuser' `
```

```
                -Group '$superuser'

$file

   FileSystem Name: filesystem1

Path              IsDirectory  Length         LastModified         Permissions  Owner              Group
----              -----------  ------         ------------         -----------  -----              -----
dir1/file1        False        1024           2020-03-23 09:57:33Z rwxrw-rw-    $superuser         $superuser

$file.ACL

DefaultScope AccessControlType EntityId Permissions
------------ ----------------- -------- -----------
False        User                       rwx
False        Group                       rw-
False        Other                       rw-

$file.Permissions

Owner       : Execute, Write, Read
Group       : Write, Read
Other       : Write, Read
StickyBit   : False
ExtendedAcls : False

$file.Properties.Metadata

Key  Value
---  -----
tag2 value2
tag1 value1
```

```
$file.Properties
```

```
LastModified          : 3/23/2020 9:57:33 AM +00:00
CreatedOn             : 3/23/2020 9:29:18 AM +00:00
Metadata              : {[tag2, value2], [tag1, value1]}
CopyCompletedOn       : 1/1/0001 12:00:00 AM +00:00
CopyStatusDescription :
CopyId                :
CopyProgress          :
CopySource            :
CopyStatus            : Pending
IsIncrementalCopy     : False
LeaseDuration         : Infinite
LeaseState            : Available
LeaseStatus           : Unlocked
ContentLength         : 1024
ContentType           : image/jpeg
ETag                  : "0x8D7CF109B9878CC"
ContentHash           : {139, 189, 187, 176...}
ContentEncoding       : UDF8
ContentDisposition    : True
ContentLanguage       : EN-US
CacheControl          : READ
AcceptRanges          : bytes
IsServerEncrypted     : True
EncryptionKeySha256   :
AccessTier            : Cool
ArchiveStatus         :
AccessTierChangedOn   : 1/1/0001 12:00:00 AM +00:00
```

This command updates all properties on a file (ACL, permission,owner, group, metadata, property can be updated with any conbination), and show them in Powershell

console.

---------- Example 3: Add an ACL entry to a directory ----------

Get the origin ACL

$acl = (Get-AzDataLakeGen2Item -FileSystem "filesystem1" -Path 'dir1/dir3/').ACL

Update permission of a new ACL entry (if ACL entry with same AccessControlType/EntityId/DefaultScope not exist, will add a new ACL entry, else update permission of

existing ACL entry)

$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType user -EntityId $id -Permission rw- -InputObject $acl

set the new acl to the directory

Update-AzDataLakeGen2Item -FileSystem "filesystem1" -Path 'dir1/dir3/' -ACL $acl

FileSystem Name: filesystem1

Path	IsDirectory	Length	LastModified	Permissions	Owner	Group
dir1/dir3	True		2020-03-23 09:34:31Z	rwxrw-rw-+	$superuser	$superuser

This command gets ACL from a directory, updates/adds an ACL entry, and sets back to the directory. If ACL entry with same AccessControlType/EntityId/DefaultScope not

exist, will add a new ACL entry, else update permission of existing ACL entry.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.storage/update-azdatalakegen2item