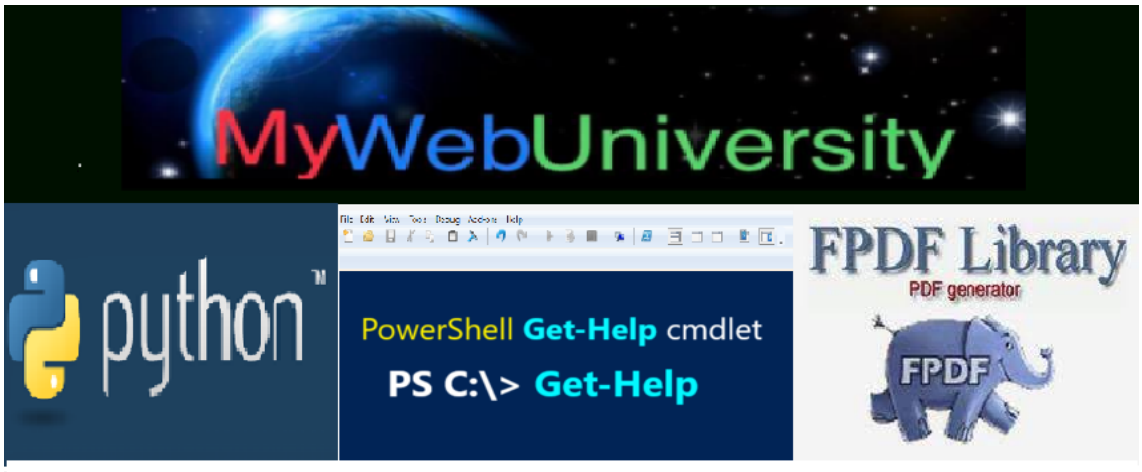

Full credit is given to all the above companies including the Operating System that this PDF file was generated!

Windows PowerShell Get-Help on Cmdlet 'Set-AzDataLakeGen2ItemAclObject'

PS:\>Get-HELP Set-AzDataLakeGen2ItemAclObject -Full

NAME

Set-AzDataLakeGen2ItemAclObject

SYNOPSIS

Creates/Updates a DataLake gen2 item ACL object, which can be used in Update-AzDataLakeGen2Item cmdlet.

SYNTAX

Set-AzDataLakeGen2ItemAclObject -AccessControlType {User | Group | Mask | Other} [-DefaultScope] [-EntityId <System.String>] [-InputObject

<Microsoft.WindowsAzure.Commands.Storage.Model.ResourceModel.PSPathAccessControlEntry[]>] -Permission <System.String> [<CommonParameters>]

DESCRIPTION

The Set-AzDataLakeGen2ItemAclObject cmdlet creates/updates a DataLake gen2 item ACL object, which can be used in Update-AzDataLakeGen2Item cmdlet. If the new ACL

entry with same AccessControlType/EntityId/DefaultScope not exist in the input ACL, will create a new ACL entry, else update permission of existing ACL entry.

PARAMETERS

-AccessControlType <Azure.Storage.Files.DataLake.Models.AccessControlType>

There are four types: "user" grants rights to the owner or a named user, "group" grants rights to the owning group or a named group, "mask" restricts rights

granted to named users and the members of groups, and "other" grants rights to all users not found in any of the other entries.

Required? true

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-DefaultScope <System.Management.Automation.SwitchParameter>

Set this parameter to indicate the ACE belongs to the default ACL for a directory; otherwise scope is implicit and the ACE belongs to the access ACL.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-EntityId <System.String>

The user or group identifier. It is omitted for entries of AccessControlType "mask" and "other". The user or group identifier is also omitted for the owner and

owning group.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

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

If input the PSPathAccessControlEntry[] object, will add the new ACL as a new element of the input PSPathAccessControlEntry[] object.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Permission <System.String>

The permission field is a 3-character sequence where the first character is 'r' to grant read access, the second character is 'w' to grant write access, and the

third character is 'x' to grant execute permission. If access is not granted, the '-' character is used to denote that the permission is denied. 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.

Required? true

Position? named

Default value None

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

None

OUTPUTS

Microsoft.WindowsAzure.Commands.Storage.Model.ResourceModel.PSPathAccessControlEntry

NOTES

Example 1: Create an ACL object with 3 ACL entry, and update ACL on a directory

```
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType user -Permission rwx -DefaultScope
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType group -Permission rw- -InputObject $acl
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType other -Permission "rw-" -InputObject $acl
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 creates an ACL object with 3 ACL entries (use -InputObject parameter to add acl entry to existing acl object), and updates ACL on a directory.

Example 2: Create an ACL object with 4 ACL entries, and update permission of an existing ACL entry

```
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType user -Permission rwx -DefaultScope
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType group -Permission rw- -InputObject $acl
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType other -Permission "rwt" -InputObject $acl
$acl = Set-AzDataLakeGen2ItemAclObject -AccessControlType user -EntityId $id -Permission rwx -InputObject $acl
$acl

DefaultScope AccessControlType EntityId                         Permissions
------------ ----------------- --------                         -----------
True         User                                               rwx
False        Group                                               rw-
False        Other                                              rwt
False        User            ********-****-****-****-************ rwx

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

DefaultScope AccessControlType EntityId                         Permissions
------------ ----------------- --------                         -----------
True         User                                               rwx
False        Group                                               rw-
False        Other                                              rw-
False        User            ********-****-****-****-************ r-x
```

This command first creates an ACL object with 4 ACL entries, then run the cmdlet again with different permission but same AccessControlType/EntityId/DefaultScope of an existing ACL entry. Then the permission of the ACL entry is updated, but no new ACL entry is added.

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/az.storage/set-azdatalakegen2itemaclobject