



## ***Windows PowerShell Get-Help on Cmdlet 'Get-AzDataLakeGen2ChildItem'***

***PS:\>Get-HELP Get-AzDataLakeGen2ChildItem -Full***

### NAME

Get-AzDataLakeGen2ChildItem

### SYNOPSIS

Lists sub directories and files from a directory or filesystem root.

### SYNTAX

```
Get-AzDataLakeGen2ChildItem [-FileSystem] <System.String> [[-Path] <System.String>] [-AsJob] [-Context
    <Microsoft.Azure.Commands.Common.Authentication.Abstractions.IStorageContext>] [-ContinuationToken
    <System.String>] [-DefaultProfile
    <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-FetchProperty]
    [-MaxCount <System.Nullable`1[System.Int32]>]
    [-OutputUserPrincipalName] [-Recurse] [<CommonParameters>]
```

### DESCRIPTION

The Get-AzDataLakeGen2ChildItem cmdlet lists sub directories and files in a directory or Filesystem in an Azure storage account. 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

-AsJob <System.Management.Automation.SwitchParameter>

Run cmdlet in the background

Required? false

Position? named

Default value False

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

-ContinuationToken <System.String>

Continuation Token.

Required? false

Position? named

Default value None

Accept pipeline input? False

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

-FetchProperty <System.Management.Automation.SwitchParameter>

Fetch the datalake item properties and ACL.

Required? false  
Position? named  
Default value False  
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

-MaxCount <System.Nullable`1[System.Int32]>

The max count of the blobs that can return.

Required? false  
Position? named  
Default value None  
Accept pipeline input? False  
Accept wildcard characters? false

-OutputUserPrincipalName <System.Management.Automation.SwitchParameter>

If specify this parameter, the user identity values returned in the owner and group fields of each list entry will be transformed from Microsoft Entra Object IDs

to User Principal Names. If not specify this parameter, the values will be returned as Microsoft Entra Object IDs. Note that group and application Object IDs are

not translated because they do not have unique friendly names.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Path <System.String>

The path in the specified Filesystem that should be retrieved. Should be a directory, in the format 'directory1/directory2/'.

Required? false

Position? 1

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-Recurse <System.Management.Automation.SwitchParameter>

Indicates if will recursively get the Child Item. The default is false.

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

## OUTPUTS

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

## NOTES

---- Example 1: List the direct sub items from a Filesystem ----

Get-AzDataLakeGen2ChildItem -FileSystem "filesystem1"

FileSystem Name: filesystem1

| Path | IsDirectory | Length | LastModified         | Permissions | Owner       | Group       |
|------|-------------|--------|----------------------|-------------|-------------|-------------|
| ---- | -----       | -----  | -----                | -----       | -----       |             |
| dir1 | True        |        | 2020-03-13 13:07:34Z | rw-r-x---   | \$superuser | \$superuser |

dir2            True            2020-03-23 09:28:36Z rwxr-x---    \$superuser        \$superuser

This command lists the direct sub items from a Filesystem

Example 2: List recursively from a directory, and fetch Properties/ACL

```
Get-AzDataLakeGen2ChildItem -FileSystem "filesystem1" -Path "dir1/" -Recurse -FetchProperty
```

FileSystem Name: filesystem1

| Path               | IsDirectory | Length | LastModified         | Permissions | Owner       | Group       |
|--------------------|-------------|--------|----------------------|-------------|-------------|-------------|
| dir1/dir3          | True        |        | 2020-03-23 09:34:31Z | rwx---rwx   | \$superuser | \$superuser |
| dir1/file1         | False       | 1024   | 2020-03-23 09:29:18Z | rwx---rwx   | \$superuser | \$superuser |
| dir1/testfile_1K_0 | False       | 1024   | 2020-03-23 09:29:21Z | rw-r-----   | \$superuser | \$superuser |

This command lists the direct sub items from a Filesystem

Example 3: List items recursively from a Filesystem in multiple batches

```
$MaxReturn = 1000
```

```
$FileSystemName = "filesystem1"
```

```
$Total = 0
```

```
$Token = $Null
```

```
do
```

```
{
```

```
    $items = Get-AzDataLakeGen2ChildItem -FileSystem $FileSystemName -Recurse -MaxCount $MaxReturn
```

```
-ContinuationToken $Token
```

```
    $Total += $items.Count
```

```
    if($items.Length -le 0) { Break;}
```

```
    $Token = $items[$items.Count -1].ContinuationToken;
```

```
}
```

```
While ($null -ne $Token)
```

```
Echo "Total $Total items in Filesystem $FileSystemName"
```

This example uses the `MaxCount` and `ContinuationToken` parameters to list items recursively from a Filesystem in multiple batches. A small `MaxCount` can limit the items

account returned from single request, may help on operation times out error, and limit the memory usage of Powershell. The first four commands assign values to

variables to use in the example. The fifth command specifies a Do-While statement that uses the `Get-AzDataLakeGen2ChildItem` cmdlet to list items. The statement

includes the continuation token stored in the `$Token` variable. `$Token` changes value as the loop runs. The final command uses the `Echo` command to display the total.

## RELATED LINKS

Online Version: <https://learn.microsoft.com/powershell/module/az.storage/get-azdatalakegen2childitem>