



Windows PowerShell Get-Help on Cmdlet 'New-AzStorageContext'

PS:\>Get-HELP New-AzStorageContext -Full

NAME

New-AzStorageContext

SYNOPSIS

Creates an Azure Storage context.

SYNTAX

New-AzStorageContext [-StorageAccountName] <System.String> -Anonymous [-Endpoint <System.String>] [-Protocol {Http | Https}] [<CommonParameters>]

New-AzStorageContext [-StorageAccountName] <System.String> -Anonymous -Environment <System.String> [-Protocol {Http | Https}] [<CommonParameters>]

New-AzStorageContext -Anonymous [-BlobEndpoint <System.String>] [-FileEndpoint <System.String>] [-QueueEndpoint <System.String>] [-TableEndpoint <System.String>]
[<CommonParameters>]

New-AzStorageContext [-StorageAccountName] <System.String> [-StorageAccountKey] <System.String> -BlobEndpoint <System.String> [-FileEndpoint <System.String>]

`[-QueueEndpoint <System.String>] [-TableEndpoint <System.String>] [<CommonParameters>]`

`New-AzStorageContext [-BlobEndpoint <System.String>] [-FileEndpoint <System.String>] [-QueueEndpoint
<System.String>] -SasToken <System.String> [-TableEndpoint
<System.String>] [<CommonParameters>]`

`New-AzStorageContext [-BlobEndpoint <System.String>] [-EnableFileBackupRequestIntent] [-FileEndpoint
<System.String>] [-QueueEndpoint <System.String>] [-TableEndpoint
<System.String>] [-UseConnectedAccount] [<CommonParameters>]`

`New-AzStorageContext -ConnectionString <System.String> [<CommonParameters>]`

`New-AzStorageContext [-StorageAccountName] <System.String> [-EnableFileBackupRequestIntent] [-Endpoint
<System.String>] [-Protocol {Http | Https}]
[-UseConnectedAccount] [<CommonParameters>]`

`New-AzStorageContext [-StorageAccountName] <System.String> [-EnableFileBackupRequestIntent] -Environment
<System.String> [-Protocol {Http | Https}]
[-UseConnectedAccount] [<CommonParameters>]`

`New-AzStorageContext [-StorageAccountName] <System.String> [-StorageAccountKey] <System.String> [-Endpoint
<System.String>] [-Protocol {Http | Https}]
[<CommonParameters>]`

`New-AzStorageContext [-StorageAccountName] <System.String> [-Endpoint <System.String>] [-Protocol {Http | Https}]
-SasToken <System.String> [<CommonParameters>]`

`New-AzStorageContext [-StorageAccountName] <System.String> [-StorageAccountKey] <System.String> -Environment
<System.String> [-Protocol {Http | Https}]
[<CommonParameters>]`

`New-AzStorageContext [-StorageAccountName] <System.String> -Environment <System.String> -SasToken
<System.String> [<CommonParameters>]`

New-AzStorageContext -Local [<CommonParameters>]

DESCRIPTION

The New-AzStorageContext cmdlet creates an Azure Storage context. The default Authentication of a Storage Context is OAuth (Microsoft Entra ID), if only input Storage account name. See details of authentication of the Storage Service in <https://learn.microsoft.com/rest/api/storageservices/authorization-for-the-azure-storage-services>.

PARAMETERS

-Anonymous <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet creates an Azure Storage context for anonymous logon.

Required?	true
Position?	named
Default value	False
Accept pipeline input?	False
Accept wildcard characters?	false

-BlobEndpoint <System.String>

Azure storage blob service endpoint

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

-ConnectionString <System.String>

Specifies a connection string for the Azure Storage context.

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

-EnableFileBackupRequestIntent <System.Management.Automation.SwitchParameter>

Required parameter to use with OAuth (Microsoft Entra ID) Authentication for Files. This will bypass any file/directory level permission checks and allow access, based on the allowed data actions, even if there are ACLs in place for those files/directories.

Required? false
Position? named
Default value False
Accept pipeline input? False
Accept wildcard characters? false

-Endpoint <System.String>

Specifies the endpoint for the Azure Storage context.

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

-Environment <System.String>

Specifies the Azure environment. The acceptable values for this parameter are: AzureCloud and AzureChinaCloud. For more information, type ``Get-Help`

`Get-AzEnvironment``.

Required? true
Position? named

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-FileEndpoint <System.String>

Azure storage file service endpoint

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Local <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet creates a context by using the local development storage account.

Required? true

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-Protocol <System.String>

Transfer Protocol (https/http).

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-QueueEndpoint <System.String>

Azure storage queue service endpoint

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

-SasToken <System.String>

Specifies a Shared Access Signature (SAS) token for the context.

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

-StorageAccountKey <System.String>

Specifies an Azure Storage account key. This cmdlet creates a context for the key that this parameter specifies.

Required? true
Position? 1
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-StorageAccountName <System.String>

Specifies an Azure Storage account name. This cmdlet creates a context for the account that this parameter specifies.

Required? true
Position? 0
Default value None
Accept pipeline input? False
Accept wildcard characters? false

-TableEndpoint <System.String>

Azure storage table service endpoint

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-UseConnectedAccount <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet creates an Azure Storage context with OAuth (Microsoft Entra ID) Authentication. The cmdlet will use OAuth Authentication by default, when other authentication not specified.

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

OUTPUTS

NOTES

Example 1: Create a context by specifying a storage account name and key

```
New-AzStorageContext -StorageAccountName "ContosoGeneral" -StorageAccountKey "< Storage Key for  
ContosoGeneral ends with == >"
```

This command creates a context for the account named ContosoGeneral that uses the specified key.

Example 2: Create a context by specifying a connection string

```
New-AzStorageContext -ConnectionString  
"DefaultEndpointsProtocol=https;AccountName=ContosoGeneral;AccountKey=< Storage Key for ContosoGeneral ends with  
== >,"
```

This command creates a context based on the specified connection string for the account ContosoGeneral.

- Example 3: Create a context for an anonymous storage account -

```
New-AzStorageContext -StorageAccountName "ContosoGeneral" -Anonymous -Protocol "http"
```

This command creates a context for anonymous use for the account named ContosoGeneral. The command specifies HTTP as a connection protocol.

Example 4: Create a context by using the local development storage account

```
New-AzStorageContext -Local
```

This command creates a context by using the local development storage account. The command specifies the Local parameter.

Example 5: Get the container for the local developer storage account

```
New-AzStorageContext -Local | Get-AzStorageContainer
```

This command creates a context by using the local development storage account, and then passes the new context to the Get-AzStorageContainer cmdlet by using the pipeline operator. The command gets the Azure Storage container for the local developer storage account.

----- Example 6: Get multiple containers -----

```
$Context01 = New-AzStorageContext -Local
```

```
$Context02 = New-AzStorageContext -StorageAccountName "ContosoGeneral" -StorageAccountKey "< Storage Key for  
ContosoGeneral ends with == >"
```

```
($Context01, $Context02) | Get-AzStorageContainer
```

The first command creates a context by using the local development storage account, and then stores that context in the \$Context01 variable. The second command creates a context for the account named ContosoGeneral that uses the specified key, and then stores that context in the \$Context02 variable. The final command gets the containers for the contexts stored in \$Context01 and \$Context02 by using Get-AzStorageContainer .

----- Example 7: Create a context with an endpoint -----

```
New-AzStorageContext -StorageAccountName "ContosoGeneral" -StorageAccountKey "< Storage Key for  
ContosoGeneral ends with == >" -Endpoint  
"contosoaccount.core.windows.net"
```

This command creates an Azure Storage context that has the specified storage endpoint. The command creates the context for the account named ContosoGeneral that uses the specified key.

--- Example 8: Create a context with a specified environment ---

```
New-AzStorageContext -StorageAccountName "ContosoGeneral" -StorageAccountKey "< Storage Key for  
ContosoGeneral ends with == >" -Environment "AzureChinaCloud"
```

This command creates an Azure storage context that has the specified Azure environment. The command creates the context for the account named ContosoGeneral that uses the specified key.

----- Example 9: Create a context by using an SAS token -----

```
$SasToken = New-AzStorageContainerSASToken -Name "ContosoMain" -Permission "rad"  
$Context = New-AzStorageContext -StorageAccountName "ContosoGeneral" -SasToken $SasToken  
$Context | Get-AzStorageBlob -Container "ContosoMain"
```

The first command generates an SAS token by using the New-AzStorageContainerSASToken cmdlet for the container named ContosoMain, and then stores that token in the

\$SasToken variable. That token is for read, add, update, and delete permissions. The second command creates a context for the account named ContosoGeneral that uses

the SAS token stored in \$SasToken, and then stores that context in the \$Context variable. The final command lists all the blobs associated with the container named

ContosoMain by using the context stored in \$Context.

Example 10: Create a context by using the OAuth Authentication

Connect-AzAccount

```
$Context = New-AzStorageContext -StorageAccountName "myaccountname" -UseConnectedAccount
```

This command creates a context by using the OAuth (Microsoft Entra ID) Authentication.

Example 11: Create a context by specifying a storage account name, storage account key and custom blob endpoint

```
New-AzStorageContext -StorageAccountName "myaccountname" -StorageAccountKey "< Storage Key for  
myaccountname ends with == >" -BlobEndpoint  
"https://myaccountname.blob.core.windows.net/"
```

This command creates a context for the account named myaccountname with a key for the account, and specified blob endpoint.

Example 12: Create a context for an anonymous storage account with specified blob endpoint

```
New-AzStorageContext -Anonymous -BlobEndpoint "https://myaccountname.blob.core.windows.net/"
```

This command creates a context for anonymous use for the account named myaccountname, with specified blob endpoint.

Example 13: Create a context by using an SAS token with specified endpoints

```
$SasToken = New-AzStorageContainerSASToken -Name "MyContainer" -Permission "rad"
```

```
New-AzStorageContext -SasToken $SasToken -BlobEndpoint "https://myaccountname.blob.core.windows.net/"  
-TableEndpoint "https://myaccountname.table.core.windows.net/"  
-FileEndpoint "https://myaccountname.file.core.windows.net/" -QueueEndpoint
```

"https://myaccountname.queue.core.windows.net/"

The first command generates an SAS token by using the New-AzStorageContainerSASToken cmdlet for the container named MyContainer, and then stores that token in the

\$SasToken variable. The second command creates a context that uses the SAS token and a specified blob endpoint, table endpoint, file endpoint, and queue endpoint.

Example 14: Create a context by using the OAuth Authentication with a specified blob endpoint

```
New-AzStorageContext -UseConnectedAccount -BlobEndpoint "https://myaccountname.blob.core.windows.net/"
```

This command creates a context by using the OAuth authentication with a specified blob endpoint.

Example 15: Create a context by using the OAuth Authentication on File service

```
New-AzStorageContext -StorageAccountName "myaccountname" -UseConnectedAccount  
-EnableFileBackupRequestIntent
```

This command creates a context to use the OAuth (Microsoft Entra ID) authentication on File service. Parameter '-EnableFileBackupRequestIntent' is required to use

OAuth (Microsoft Entra ID) Authentication for File service. This will bypass any file/directory level permission checks and allow access, based on the allowed data

actions, even if there are ACLs in place for those files/directories.

RELATED LINKS

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

Get-AzStorageBlob

New-AzStorageContainerSASToken

