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

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

NAME

New-SqlColumnEncryptionKey

SYNOPSIS

Creates a column encryption key object in the database.

SYNTAX

New-SqlColumnEncryptionKey [-Name] <String> [-InputObject] <Database> [-AccessToken <PSObject>] -ColumnMasterKeyName <String> [-Encrypt {Mandatory | Optional | Strict}] [-EncryptedValue <String>] [-HostNameInCertificate <String>] [-KeyVaultAccessToken <String>] [-ManagedHsmAccessToken <String>] [-ProgressAction <ActionPreference>] [-Script] [-TrustServerCertificate] [<CommonParameters>]

New-SqlColumnEncryptionKey [-Name] <String> [[-Path] <String>] [-AccessToken <PSObject>] -ColumnMasterKeyName <String> [-Encrypt {Mandatory | Optional | Strict}] [-EncryptedValue <String>] [-HostNameInCertificate <String>] [-KeyVaultAccessToken <String>] [-ManagedHsmAccessToken <String>] [-ProgressAction <ActionPreference>] [-Script] [-TrustServerCertificate] [<CommonParameters>]

DESCRIPTION

The New-SqlColumnEncryptionKey cmdlet creates a column encryption key object in the database. A column encryption key object encapsulates an encrypted value of a

symmetric cryptographic key that can be subsequently used to encrypt database columns using the Always Encrypted feature.

This cmdlet supports two modes of operation:

- If the encrypted value of a column encryption key is specified, the cmdlet simply creates a new column encryption key object encapsulating the specified encrypted

value.

- If the encrypted value of a column encryption key is not specified, the cmdlet first generates a plaintext key value, encrypts it with the specified column master

key, and then creates a new column encryption key object encapsulating the generated encrypted value. In this mode, the cmdlet communicates with a key store

holding the column master key. If the key is stored in Azure, you need to specify a valid authentication token for a key vault or a managed HSM holding the key.

Alternatively, you can authenticate to Azure with Add-SqlAzureAuthenticationContext before calling this cmdlet.

> `Module requirements: version 21+ on PowerShell 5.1; version 22+ on PowerShell 7.x.`

PARAMETERS

-AccessToken <PSObject>

The access token used to authenticate to SQL Server, as an alternative to user/password or Windows Authentication.

This can be used, for example, to connect to `SQL Azure DB` and `SQL Azure Managed Instance` using a `Service Principal` or a `Managed Identity`.

The parameter to use can be either a string representing the token or a `PSAccessToken` object as returned by running `Get-AzAccessToken -ResourceUrl

https://database.windows.net`.

> This parameter is new in v22 of the module.

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

-ColumnMasterKeyName <String>

Specifies the name of the column master key that was used to produce the specified encrypted value of the column encryption key, or the name the column master key

that is used to produce the new encrypted value.

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

-Encrypt <String>

The encryption type to use when connecting to SQL Server.

This value maps to the `Encrypt` property `SqlConnectionEncryptOption` on the SqlConnection object of the Microsoft.Data.SqlClient driver.

In v22 of the module, the default is `Optional` (for compatibility with v21). In v23+ of the module, the default value will be 'Mandatory', which may create a

breaking change for existing scripts.

> This parameter is new in v22 of the module.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-EncryptedValue <String>

Specifies a hexadecimal string that is an encrypted column encryption key value.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-HostNameInCertificate <String>

The host name to be used in validating the SQL Server TLS/SSL certificate. You must pass this parameter if your SQL Server instance is enabled for Force

Encryption and you want to connect to an instance using hostname/shortname. If this parameter is omitted then passing the Fully Qualified Domain Name (FQDN) to

-ServerInstance is necessary to connect to a SQL Server instance enabled for Force Encryption.

> This parameter is new in v22 of the module.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InputObject <Database>

Specifies the SQL database object, for which this cmdlet runs the operation.

Required? true

Position? 2

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-KeyVaultAccessToken <String>

Specifies an access token for key vaults in Azure Key Vault. Use this parameter if the column master key you want to use to encrypt the new column encryption key

is stored in a key vault in Azure Key Vault.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ManagedHsmAccessToken <String>

Specifies an access token for managed HSMs in Azure Key Vault. Use this parameter if the column master key you want to use to encrypt the new column encryption

key is stored in a managed HSM in Azure Key Vault.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Name <String>

Specifies the name of the column encryption key object to be created.

Required? true

Position? 1

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Path <String>

Specifies the path of the SQL database, for which this cmdlet runs the operation. If you do not specify a value for this parameter, the cmdlet uses the current

working location.

Required? false

Position? 2

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ProgressAction <ActionPreference>

Determines how PowerShell responds to progress updates generated by a script, cmdlet, or provider, such as the progress bars generated by the Write-Progress

cmdlet. The Write-Progress cmdlet creates progress bars that show a command's status.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Script [<SwitchParameter>]

Indicates that this cmdlet runs a Transact-SQL script that performs the operation.

Required? false

Position? named

Default value False

Accept pipeline input? False

Accept wildcard characters? false

-TrustServerCertificate [<SwitchParameter>]

Indicates whether the channel will be encrypted while bypassing walking the certificate chain to validate trust.

In v22 of the module, the default is `$true` (for compatibility with v21). In v23+ of the module, the default value will be '$false', which may create a breaking

change for existing scripts.

> This parameter is new in v22 of the module.

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

Microsoft.SqlServer.Management.Smo.Database

OUTPUTS

SqlColumnEncryptionKey

NOTES

--- Example 1: Generate and encrypt a column encryption key ---

New-SqlColumnEncryptionKey -Name 'CEK1' -ColumnMasterKeyName 'CMK1'

This command generates a plaintext value of a column encryption key, encrypts the plaintext value with the specified master key, and then creates a column encryption

key object, encapsulating the generated encrypted value in the database.

Example 2: Generate and encrypt a column encryption key using a column master key stored in a key vault in Azure Key Vault.

#?Connect?to?Azure?account.

Import-Module?Az.Accounts?-MinimumVersion?2.2.0

Connect-AzAccount

#?Obtain?the?access?token.

$keyVaultAccessToken?=?(Get-AzAccessToken?-ResourceUrl?https://vault.azure.net).Token

#?Pass?the?token?to?the?cmdlet.?It?will?use?the?token?to?communicate?with?the?key vault containing the column master key.

New-SqlColumnEncryptionKey -Name 'CEK1' -ColumnMasterKeyName 'CMK1' -KeyVaultAccessToken $keyVaultAccessToken

Example 3: Create a column encryption key object for an existing encrypted value of a column encryption key.

New-SqlColumnEncryptionKey -Name 'CEK1' -ColumnMasterKeyName 'CMK1' -EncryptedValue '0x01700000016C006F00630061006C006D0061006300680069006E0065002F006D0079002F00320066

00610066006400380031003200310034003400340065006200310061003200650030003600390033003400380061003500
64003400300032003300380065006600620063006300610031006300284FC4316518C

F3328A6D9304F65DD2CE387B79D95D077B4156E9ED8683FC0E09FA848275C685373228762B02DF2522AFF6D66178
2607B4A2275F2F922A5324B392C9D498E4ECFC61B79F0553EE8FB2E5A8635C4DBC0224D5A7F

1B136C182DCDE32A00451F1A7AC6B4492067FD0FAC7D3D6F4AB7FC0E86614455DBB2AB37013E0A5B8B5089B180
CA36D8B06CDB15E95A7D06E25AACB645D42C85B0B7EA2962BD3080B9A7CDB805C6279FE7DD694

1E7EA4C2139E0D4101D8D7891076E70D433A214E82D9030CF1F40C503103075DEEB3D64537D15D244F503C2750CF
940B71967F51095BFA51A85D2F764C78704CAB6F015EA87753355367C5C9F66E465C0C66BAD

EDFDF76FB7E5C21A0D89A2FCCA8595471F8918B1387E055FA0B816E74201CD5C50129D29C015895CD073925B6EA
87CAF4A4FAF018C06A3856F5DFB724F42807543F777D82B809232B465D983E6F19DFB572BEA7

B61C50154605452A891190FB5A0C4E464862CF5EFAD5E7D91F7D65AA1A78F688E69A1EB098AB42E95C674E234173
CD7E0925541AD5AE7CED9A3D12FDFE6EB8EA4F8AAD2629D4F5A18BA3DDCC9CF7F352A892D4B

EBDC4A1303F9C683DACD51A237E34B045EBE579A381E26B40DCFBF49EFFA6F65D17F37C6DBA54AA99A65D5573D
4EB5BA038E024910A4D36B79A1D4E3C70349DADFF08FD8B4DEE77FDB57F01CB276ED5E676F1EC
973154F86'

RELATED LINKS

Online Version: https://learn.microsoft.com/powershell/module/sqlserver/new-sqlcolumnencryptionkey

Get-SqlColumnEncryptionKey

Remove-SqlColumnEncryptionKey