



Windows PowerShell Get-Help on Cmdlet 'Import-AzKeyVaultCertificate'

PS:\>Get-HELP Import-AzKeyVaultCertificate -Full

NAME

Import-AzKeyVaultCertificate

SYNOPSIS

Imports a certificate to a key vault.

SYNTAX

```
Import-AzKeyVaultCertificate [-VaultName] <System.String> [-Name] <System.String> [-CertificateCollection]
<System.Security.Cryptography.X509Certificates.X509Certificate2Collection> [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-PolicyObject
<Microsoft.Azure.Commands.KeyVault.Models.PSKeyVaultCertificatePolicy>] [-PolicyPath <System.String>] [-Tag
<System.Collections.Hashtable>] [-Confirm] [-WhatIf]
[<CommonParameters>]

Import-AzKeyVaultCertificate [-VaultName] <System.String> [-Name] <System.String> -CertificateString <System.String>
[-ContentType <System.String>] [-DefaultProfile
<Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] [-Password
<System.Security.SecureString>] [-PolicyObject
<Microsoft.Azure.Commands.KeyVault.Models.PSKeyVaultCertificatePolicy>] [-PolicyPath <System.String>] [-Tag
```

<System.Collections.Hashtable>] [-Confirm] [-WhatIf]

[<CommonParameters>]

Import-AzKeyVaultCertificate [-VaultName] <System.String> [-Name] <System.String> [-DefaultProfile
 <Microsoft.Azure.Commands.Common.Authentication.Abstractions.Core.IAzureContextContainer>] -FilePath
<System.String> [-Password <System.Security.SecureString>]
 [-PolicyObject <Microsoft.Azure.Commands.KeyVault.Models.PSKeyVaultCertificatePolicy>] [-PolicyPath
<System.String>] [-Tag <System.Collections.Hashtable>] [-Confirm]
[-WhatIf] [<CommonParameters>]

DESCRIPTION

The Import-AzKeyVaultCertificate cmdlet imports a certificate into a key vault. You can create the certificate to import by using one of the following methods: - Use

`Add-AzKeyVaultCertificate` to create a certificate signing request and submit it to a certificate authority. See <https://learn.microsoft.com/azure/key-vault/certificates/create-certificate-signing-request>

- Use an existing certificate package file, such as a .pfx or .p12 file, which contains both the certificate and private key.

PARAMETERS

-CertificateCollection <System.Security.Cryptography.X509Certificates.X509Certificate2Collection>

Specifies the certificate collection to add to a key vault.

Required? true

Position? 2

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-CertificateString <System.String>

Base64 encoded representation of the certificate object to import. This certificate needs to contain the private key.

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

-ContentType <System.String>

Specifies the type of the certificate to be imported. Regards certificate string as PFX format by default.

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

-FilePath <System.String>

Specifies the path of the certificate file that this cmdlet imports.

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

-Name <System.String>

Specifies the certificate name. This cmdlet constructs the fully qualified domain name (FQDN) of a certificate from key vault name, currently selected environment, and certificate name.

Required? true

Position? 1

Default value None

Accept pipeline input? True (ByPropertyName)

Accept wildcard characters? false

-Password <System.Security.SecureString>

Specifies the password for a certificate file.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-PolicyObject <Microsoft.Azure.Commands.KeyVault.Models.PSKeyVaultCertificatePolicy>

An in-memory object to specify management policy for the certificate. Mutual-exclusive to PolicyPath.

Required? false

Position? named

Default value None

Accept pipeline input? True (ByValue)

Accept wildcard characters? false

-PolicyPath <System.String>

A file path to specify management policy for the certificate that contains JSON encoded policy definition. Mutual-exclusive to PolicyObject.

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

-Tag <System.Collections.Hashtable>

Key-value pairs in the form of a hash table. For example: @{key0="value0";key1=\$null;key2="value2"}

Required? false
Position? named
Default value None
Accept pipeline input? True (ByPropertyName)
Accept wildcard characters? false

-VaultName <System.String>

Specifies the key vault name into which this cmdlet imports certificates. This cmdlet constructs the fully qualified domain name (FQDN) of a key vault based on the name and currently selected environment.

Required? true
Position? 0
Default value None
Accept pipeline input? True (ByPropertyName)
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

System.Security.Cryptography.X509Certificates.X509Certificate2Collection

System.Collections.Hashtable

OUTPUTS

Microsoft.Azure.Commands.KeyVault.Models.PSKeyVaultCertificate

NOTES

----- Example 1: Import a key vault certificate -----

```
$Password = ConvertTo-SecureString -String "123" -AsPlainText -Force
```

```
Import-AzKeyVaultCertificate -VaultName "ContosoKV01" -Name "ImportCert01" -FilePath  
"C:\Users\contosoUser\Desktop\import.pfx" -Password $Password
```

Name : importCert01

Certificate : [Subject]

CN=contoso.com

[Issuer]

CN=contoso.com

[Serial Number]

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[Not Before]

2/8/2016 3:11:45 PM

[Not After]

8/8/2016 4:21:45 PM

[Thumbprint]

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Thumbprint : XX

Tags :

Enabled : True

Created : 2/8/2016 11:50:43 PM

Updated : 2/8/2016 11:50:43 PM

The first command uses the ConvertTo-SecureString cmdlet to create a secure password, and then stores it in the \$Password variable. The second command imports the certificate named ImportCert01 into the CosotosoKV01 key vault.

Example 2: Import a key vault certificate by CertificateString

```
$Password = ConvertTo-SecureString -String "123" -AsPlainText -Force
```

```
$Base64String = [System.Convert]::ToBase64String([System.IO.File]::ReadAllBytes("import.pfx"))
```

```
Import-AzKeyVaultCertificate -VaultName "ContosoKV01" -Name "ImportCert01" -CertificateString $Base64String  
-Password $Password
```

Name : importCert01

Certificate : [Subject]

CN=contoso.com

[Issuer]

CN=contoso.com

[Serial Number]

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[Not Before]

2/8/2016 3:11:45 PM

[Not After]

8/8/2016 4:21:45 PM

[Thumbprint]

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Thumbprint : XX

Tags :

Enabled : True

Created : 2/8/2016 11:50:43 PM

Updated : 2/8/2016 11:50:43 PM

The first command uses the ConvertTo-SecureString cmdlet to create a secure password, and then stores it in the \$Password variable. The second command reads a

certificate as a Base64 encoded representation. The third command imports the certificate named ImportCert01 into the CosotosoKV01 key vault.

-- Example 3: Import a key vault certificate with PolicyFile --

```
$Password = ConvertTo-SecureString -String "123" -AsPlainText -Force  
Import-AzKeyVaultCertificate -VaultName "ContosoKV01" -Name "ImportCert01" -FilePath  
"C:\Users\contosoUser\Desktop\import.pfx" -Password $Password -PolicyPath  
"C:\Users\contosoUser\Desktop\policy.json"
```

Name : importCert01

Certificate : [Subject]

CN=contoso.com

[Issuer]

CN=contoso.com

[Serial Number]

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

[Not Before]

2/8/2016 3:11:45 PM

[Not After]

8/8/2016 4:21:45 PM

[Thumbprint]

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

KeyId : https://ContosoKV01.vault.azure.net/keys/ImportCert01/XXXXXXXXXXXXXXXXXXXXXXXXXXXX

SecretId : https://ContosoKV01.vault.azure.net/secrets/ImportCert01/XXXXXXXXXXXXXXXXXXXXXXXXXXXX

Thumbprint : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Policy :

Secret Content Type: application/x-pkcs12

Issuer Name : Unknown

Created On : 3/22/2023 6:00:52 AM

Updated On : 4/27/2023 9:52:53 AM

...

RecoveryLevel : Recoverable+Purgeable

Enabled : True

Expires : 6/9/2023 6:20:26 AM

NotBefore : 3/11/2023 6:20:26 AM

Created : 4/24/2023 9:05:51 AM

Updated : 4/24/2023 9:05:51 AM

Tags : {}

VaultName : ContosoKV01

Name : ImportCert01

Version : XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Id : https://ContosoKV01.vault.azure.net/certificates/ImportCert01/XXXXXXXXXXXXXXXXXXXXXXXXXXXX

The first command uses the ConvertTo-SecureString cmdlet to create a secure password, and then stores it in the \$Password variable. The second command imports the certificate named ImportCert01 into the CosotosoKV01 key vault with a policy defined by file.

RELATED LINKS

Online Version: <https://learn.microsoft.com/powershell/module/az.keyvault/import-azkeyvaultcertificate>

[Remove-AzKeyVaultCertificate](#)

[Creating and merging CSR in Key Vault](#)

<https://learn.microsoft.com/azure/key-vault/certificates/create-certificate-signing-request>