

# Cmdlet Reference for Windows Azure Pack Configuration Cmdlets

---

Microsoft Corporation

Published: November 1, 2013

## Applies To

Windows Azure Pack for Windows Server

## Feedback

Send suggestions and comments about this document to [sc2012docs@microsoft.com](mailto:sc2012docs@microsoft.com).

# Copyright

---

This document is provided "as-is". Information and views expressed in this document, including URL and other Internet website references, may change without notice.

Some examples depicted herein are provided for illustration only and are fictitious. No real association or connection is intended or should be inferred.

This document does not provide you with any legal rights to any intellectual property in any Microsoft product. You may copy and use this document for your internal, reference purposes. You may modify this document for your internal, reference purposes.

© 2013 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Bing, Excel, Hyper-V, Internet Explorer, Silverlight, SQL Server, Windows, Windows Intune, Windows PowerShell, Windows Server, and Windows Vista are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners.

## Revision History

Release Date	Changes
November 1, 2013	Initial release of this document.

---

## Contents

Add-MgmtSvcAdminUser .....	5
Add-MgmtSvcDatabaseUser .....	10
Add-MgmtSvcResourceProviderConfiguration .....	15
Get-MgmtSvcAdminUser .....	20
Get-MgmtSvcDatabaseSetting .....	24
Get-MgmtSvcDefaultDatabaseName .....	29
Get-MgmtSvcEndpoint.....	31
Get-MgmtSvcFeature .....	34
Get-MgmtSvcFqdn.....	36
Get-MgmtSvcNamespace.....	40
Get-MgmtSvcNotificationSubscriber .....	42
Get-MgmtSvcResourceProviderConfiguration .....	46
Get-MgmtSvcSchema.....	49
Get-MgmtSvcSetting.....	51
Initialize-MgmtSvcFeature .....	53
Initialize-MgmtSvcProduct .....	58
Install-MgmtSvcDatabase.....	62
New-MgmtSvcMachineKey.....	66
New-MgmtSvcPassword.....	69
New-MgmtSvcResourceProviderConfiguration .....	71
New-MgmtSvcSelfSignedCertificate.....	84
Protect-MgmtSvcConfiguration.....	87
Remove-MgmtSvcAdminUser .....	90
Remove-MgmtSvcDatabaseUser .....	95
Remove-MgmtSvcNotificationSubscriber .....	100

---

Remove-MgmtSvcResourceProviderConfiguration .....	104
Reset-MgmtSvcPassphrase .....	108
Set-MgmtSvcCeip .....	113
Set-MgmtSvcDatabaseSetting .....	115
Set-MgmtSvcDatabaseUser .....	121
Set-MgmtSvcFqdn .....	126
Set-MgmtSvcIdentityProviderSettings .....	130
Set-MgmtSvcNotificationSubscriber .....	136
Set-MgmtSvcPassphrase .....	145
Set-MgmtSvcRelyingPartySettings .....	149
Set-MgmtSvcSetting .....	154
Test-MgmtSvcDatabase .....	158
Test-MgmtSvcPassphrase .....	162
Test-MgmtSvcProtectedConfiguration .....	167
Uninstall-MgmtSvcDatabase .....	169
Unprotect-MgmtSvcConfiguration .....	173
Update-MgmtSvcV1Data .....	175

# Add-MgmtSvcAdminUser

---

## Add-MgmtSvcAdminUser

Adds an administrative principal to the database.

### Syntax

Parameter Set: ConnectionParameters

```
Add-MgmtSvcAdminUser [-Principal] <String> [-Database <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Add-MgmtSvcAdminUser [-Principal] <String> [-ConnectionString <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Add-MgmtSvcAdminUser** cmdlet adds an administrative user or group to the database. The cmdlet grants the principal permissions to access the management portal for administrators and Windows Azure Pack administrator API.

If this cmdlet is run on the computer on which the Admin API service is installed and the Web.config file contains values for EncryptionKey and EncryptionAlgorithm, then those values are used.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database. If a value is not provided for this parameter, the cmdlet uses the management database (Microsoft.MgmtSvc.Store).

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionAlgorithm<String>**

Specifies an encryption algorithm.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionKey<String>**

Specifies an encryption key, as a hexadecimal string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Principal<String>**

Specifies an administrative user or group.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies a server. If a value is not provided for this parameter, the cmdlet uses localhost (or '.').

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-UserName<String>**

Specifies a user name. If a value is not provided for this parameter, the cmdlet uses integrated security.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Add a principal by using a connection string

This command adds the specified user as an administrative principal to the database. The command uses the specified connection string to connect to the database.

```
PS C:\> Add-MgmtSvcAdminUser -Principal "PattiFuller@Contoso.com" -ConnectionString  
'Server=.\sqlexpress;Initial Catalog=Microsoft.MgmtSvc.Store;User Id=sa;Password=PassWord;'
```

## Related topics

[Get-MgmtSvcAdminUser](#)

[Remove-MgmtSvcAdminUser](#)

---

# Add-MgmtSvcDatabaseUser

---

## Add-MgmtSvcDatabaseUser

Adds SQL authentication credentials for a user to the database.

### Syntax

Parameter Set: Default

```
Add-MgmtSvcDatabaseUser -Schema <String> -User <String> -UserPassword <String> [-  
ConnectionString <String> ] [-Database <String> ] [-Password <String> ] [-RoleName <String>  
] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Add-MgmtSvcDatabaseUser** cmdlet adds SQL authentication credentials for a specified user in the database. Specify the user name and password. You can also specify a user role for the account. Note that the **Initialize-MgmtSvcFeature** cmdlet adds the necessary users to Windows Azure Pack for Windows Server. You only need to use **Add-MgmtSvcDatabaseUser** to add additional users.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-RoleName<String>**

Specifies the name of a role for the added account. Roles are used to grant or deny user access to database objects. Roles and database access are specific to a schema.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Schema<String>**

Specifies a schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-User<String>**

Specifies the user name of the principal. The cmdlet adds this account to the database for the specified schema.

Aliases	none
Required?	true
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserPassword<String>**

Specifies a password for the principal account specified by the *User* parameter. The cmdlet adds this account to the database for the specified schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Add a user to a database for a schema**

This command adds the user named user to the database named Contoso.MgmtSvc.Store for the schema SqlServer. The command specifies the computer that runs SQL Server. The command also specifies a password and the role DBO for the new user.

```
PS C:\> Add-MgmtSvcDatabaseUser -Schema "SqlServer" -User "user" -UserPassword "Sand3289" -RoleName "DBO" -Database "Contoso.MgmtSvc.Store" -Server "ContosoSQLServer"
```

## **Related topics**

[Remove-MgmtSvcDatabaseUser](#)

[Set-MgmtSvcDatabaseUser](#)

[Get-MgmtSvcSchema](#)

[Install-MgmtSvcDatabase](#)

---

# Add-MgmtSvcResourceProviderConfiguration

---

## Add-MgmtSvcResourceProviderConfiguration

Adds a resource provider to Windows Azure Pack.

### Syntax

Parameter Set: Default

```
Add-MgmtSvcResourceProviderConfiguration [-ResourceProvider] <ResourceProvider> [-As <String> ] [-ConnectionString <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Add-MgmtSvcResourceProviderConfiguration** cmdlet adds a resource provider to Windows Azure Pack for Windows Server. You can run this cmdlet from any computer in the deployment. If you run this cmdlet on the computer on which the Admin API service is installed and the Web.config file contains values for *ConnectionString*, *EncryptionKey*, and *EncryptionAlgorithm*, then those values are used.

### Parameters

#### **-As<String>**

Specifies an output format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionAlgorithm<String>**

Specifies an encryption algorithm.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionKey<String>**

Specifies an encryption key, as a hexadecimal string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Force**

Updates an existing resource provider when one is found with the same name and instance ID of the one provided with this cmdlet. If there is no existing resource provider with the provided name and instance ID, then the resource provider is added, and this parameter is ignored.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ResourceProvider<ResourceProvider>**

Specifies a resource provider.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Add a resource provider

The first five commands set variables to use when creating a new resource provider.

The sixth command creates a resource provider and stores the resulting resource provider object in the \$RP variable. For information about creating a resource provider, see the **New-MgmtSvcResourceProviderConfiguration** cmdlet.

The last command adds the resource provider stored in \$RP.

```
PS C:\> $ConnectionString = ""
PS C:\> $EncryptionKey = "D576FCB3740049D44183C8BD6AB7979FB68DF253A1AFAB1BEDD987907358397D"
PS C:\> $EncryptionAlgorithm = "AES"
PS C:\> $UserName = "PattiFuller"
PS C:\> $Password = "passw0rd"
PS C:\> $RP = New-MgmtSvcResourceProviderConfiguration -Name 'RP01' `
-DisplayName 'Resource Provider 01' `
-AdminForwardingAddress "https://$Env:ComputerName:30010/" `
-AdminAuthenticationMode 'Basic' `
-AdminAuthenticationUserName $UserName `
-AdminAuthenticationPassword $Password `
-TenantForwardingAddress "https://$Env:ComputerName:30010/subscriptions" `
-TenantAuthenticationMode 'Basic' `
-TenantAuthenticationUserName $UserName `
-TenantAuthenticationPassword $Password `
-TenantSourceUriTemplate '{subid}/services/sqlservers/{*path}' `
-TenantTargetUriTemplate '{subid}/{*path}' `
-UsageForwardingAddress "https://$Env:ComputerName:30010/" `
-UsageAuthenticationMode 'Basic' `
-UsageAuthenticationUserName $UserName `
```

---

```
-UsageAuthenticationPassword $Password `
-NotificationForwardingAddress "https://$Env:ComputerName`:30010/" `
-NotificationAuthenticationMode 'Basic' `
-NotificationAuthenticationUserName $UserName `
-NotificationAuthenticationPassword $Password
PS C:\> Add-MgmtSvcResourceProviderConfiguration -ResourceProvider $RP
```

## Related topics

[New-MgmtSvcResourceProviderConfiguration](#)

[Get-MgmtSvcResourceProviderConfiguration](#)

[Remove-MgmtSvcResourceProviderConfiguration](#)

---

# Get-MgmtSvcAdminUser

---

## Get-MgmtSvcAdminUser

Gets an administrative principal from the database.

### Syntax

Parameter Set: ConnectionParameters

```
Get-MgmtSvcAdminUser [[-Principal] <String[]> ] [-Database <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Get-MgmtSvcAdminUser [[-Principal] <String[]> ] [-ConnectionString <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcAdminUser** cmdlet gets an administrative user from the database.

If this cmdlet is run on the computer on which the Admin API service is installed and the Web.config file contains values for EncryptionKey and EncryptionAlgorithm, then those values are used.

You must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EncryptionAlgorithm<String>**

Specifies an encryption algorithm.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EncryptionKey<String>**

Specifies an encryption key, as a hexadecimal string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Principal<String[]>**

Specifies an administrative user or group.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	true

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **1: Get a principal by using a connection string**

This command gets the principal named Patti Fuller by using a connection string to connect to the database.

```
PS C:\> Get-MgmtSvcAdminUser -Principal "PattiFuller@Contoso.com" -ConnectionString  
'Server=.\sqlexpress;Initial Catalog=Contoso.MgmtSvc.Store;User Id=sa;Password=PassWord;'
```

## **Related topics**

[Add-MgmtSvcAdminUser](#)

[Remove-MgmtSvcAdminUser](#)

---

# Get-MgmtSvcDatabaseSetting

---

## Get-MgmtSvcDatabaseSetting

Reads settings from a configuration database.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcDatabaseSetting [-Namespace] <String> [-Name] <String> [-ConnectionString  
<String> ] [-Database <String> ] [-Passphrase <String> ] [-Password <String> ] [-Server  
<String> ] [-UserName <String> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcDatabaseSetting** cmdlet reads the configuration settings of a management service component from the database. A setting consists of a namespace, a name, and a value. If you protected a setting by encrypting it, you need to supply a value for the *Passphrase* parameter to decrypt the setting.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Database<String>**

Specifies a database name. The Config schema must be installed in this database.

By default, the Config schema is installed in the following databases:

- Microsoft.MgmtSvc.Config
- Microsoft.MgmtSvc.PortalConfigStore
- Microsoft.MgmtSvc.Store

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies an array of names for a setting.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	true

## **-Namespace<String>**

Specifies an array of namespaces for a setting.

Aliases	none
Required?	true
Position?	1

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Passphrase<String>**

Specifies a passphrase. If a setting was previously encrypted, provide a passphrase to decrypt the setting. A passphrase must be provided if the setting requires it.

By default, only Microsoft.MgmtSvc.Config settings require a passphrase.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a password for the user to connect to the database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
---------	------

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account used to connect to the database.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **Example 1: Get a setting**

The first command stores a connection string in the \$ConnectionString variable.

The second command gets the setting that has the specified namespace, name, and value. A passphrase is supplied to decrypt the setting.

```
PS C:\> $ConnectionString = 'Data Source=rd-source04;Initial
Catalog=Microsoft.MgmtSvc.Config;User ID=SA;Password=PassWord01'
PS C:\> Get-MgmtSvcDatabaseSetting -Namespace "TenantSite" -Name "machineKey.decryptionKey"
-ConnectionString $ConnectionString -Passphrase "PassPhrase01"
```

---

## Related topics

[Set-MgmtSvcDatabaseSetting](#)

[Test-MgmtSvcPassphrase](#)

[Reset-MgmtSvcPassphrase](#)

[Get-MgmtSvcDefaultDatabaseName](#)

---

# Get-MgmtSvcDefaultDatabaseName

---

## Get-MgmtSvcDefaultDatabaseName

Gets the names of the default databases.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcDefaultDatabaseName [[-Name] <String[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcDefaultDatabaseName** cmdlet gets the names of the default databases. If you use this cmdlet without specifying a parameter, all default database names are returned. To get a specific default database name, use the *Name* parameter.

### Parameters

#### **-Name<String[]>**

Specifies an array of names for default databases.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	true

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Examples

### Example 1: Get a default database name

This example returns information about the default databases named Microsoft.MgmtSvc.Config and Microsoft.MgmtSvc.Store.

```
PS C:\> Get-MgmtSvcDefaultDatabaseName -Name  
"Microsoft.MgmtSvc.Config", "Microsoft.MgmtSvc.Store"
```

---

# Get-MgmtSvcEndpoint

---

## Get-MgmtSvcEndpoint

Gets management service component endpoints.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcEndpoint [[-Namespace] <String[]> ] [[-Name] <String[]> ] [[-Fqdn] <String> ] [[-Port] <Int32> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcEndpoint** cmdlet gets one or more management service component endpoints. You must run this cmdlet on the computer that hosts the desired namespace.

### Parameters

#### **-Fqdn<String>**

Specifies a fully qualified domain name (FQDN). This FQDN identifies the computer that hosts the Admin API service.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

#### **-Name<String[]>**

Specifies an array of endpoint names.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

### **-Namespace<String[]>**

Specifies an array of endpoint namespaces.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

### **-Port<Int32>**

Specifies a port number.

Aliases	none
Required?	false
Position?	4
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).



---

## Examples

### Example 1: Get management service component endpoints

This command gets endpoints that are configured for the feature in the namespace named AdminAPI.

```
PS C:\> Get-MgmtSvcEndpoint -Namespace "AdminAPI"
```

### Related topics

[Get-MgmtSvcFeature](#)

[Get-MgmtSvcNamespace](#)

[Get-MgmtSvcSchema](#)

---

# Get-MgmtSvcFeature

---

## Get-MgmtSvcFeature

Gets management service features.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcFeature [-Name] <String[]> ] [ <CommonParameters>
```

### Detailed Description

The **Get-MgmtSvcFeature** cmdlet returns a list of installed management service features, including whether the feature has been configured.

### Parameters

#### **-Name<String[]>**

Specifies an array of names of management service features. You can use wildcards.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	true

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### Example 1: Get management service features

This command gets all management service features that are installed on the local computer. The command also returns the configuration status of the computer.

```
PS C:\> Get-MgmtSvcFeature
```

## Related topics

[Initialize-MgmtSvcFeature](#)

[Get-MgmtSvcEndpoint](#)

[Get-MgmtSvcNamespace](#)

[Get-MgmtSvcSchema](#)

---

# Get-MgmtSvcFqdn

---

## Get-MgmtSvcFqdn

Gets the FQDN for an Admin or Tenant site.

### Syntax

Parameter Set: ConnectionParameters

```
Get-MgmtSvcFqdn [-Namespace] <String[]> [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Get-MgmtSvcFqdn [-Namespace] <String[]> [-ConnectionString <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcFqdn** cmdlet gets the Fully Qualified Domain Name (FQDN) for an Admin or Tenant site.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Namespace<String[]>**

Specifies an array of namespaces. Valid values are: AdminSite, TenantSite.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Get an FQDN for a Tenant Site

This command gets the FQDN for a tenant site.

```
PS C:\> Get-MgmtSvcFqdn -Namespace TenantSite -Password "PassWord01!" -Server "Computer01" -
UserName "PattiFuller"
```

## Related topics

[Set-MgmtSvcFqdn](#)

---

# Get-MgmtSvcNamespace

---

## Get-MgmtSvcNamespace

Gets registered management service namespaces.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcNamespace [[-Namespace] <String[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcNamespace** cmdlet returns a list of registered management service namespaces. A namespace groups a set of related configuration settings. For example, a set of connectionStrings and App settings in the web.config file of a feature constitutes a namespace.

### Parameters

#### **-Namespace<String[]>**

Specifies an array of namespaces of management service settings.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).



---

## Examples

### Example 1: Get management service namespaces

This command gets the management service namespaces for the local computer.

```
PS C:\> Get-MgmtSvcNamespace
```

MySQL

SQLServer

Monitoring

UsageService

UsageCollector

WebAppGallery

Notification

AdminAPI

TenantAPI

TenantPublicAPI

AdminSite

TenantSite

### Related topics

[Get-MgmtSvcFeature](#)

[Get-MgmtSvcEndpoint](#)

[Get-MgmtSvcSchema](#)

---

# Get-MgmtSvcNotificationSubscriber

---

## Get-MgmtSvcNotificationSubscriber

Gets a notification subscriber.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcNotificationSubscriber [[-Name] <String[]> ] [-ConnectionString <String> ] [-Database <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcNotificationSubscriber** cmdlet gets one or more notification subscribers from the usage service. By default, all notification subscribers are returned. To get a specific notification subscriber, you can provide its name.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

This cmdlet also assumes that the usage service is installed on the local computer. If the usage service is installed on another computer, you must use the *EncryptionKey* and *EncryptionAlgorithm* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EncryptionAlgorithm<String>**

Specifies an encryption algorithm.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EncryptionKey<String>**

Specifies an encryption key, as a hexadecimal string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Name<String[]>**

Specifies an array of notification subscriber names. You can use wildcards.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Get a notification subscriber**

This command gets the notification subscriber named Billing. In this example, the database is on the local computer, therefore, the command does not specify the server, database, user name, or password parameters.

```
PS C:\> Get-MgmtSvcNotificationSubscriber -Name "Billing"
```

## **Related topics**

[Remove-MgmtSvcNotificationSubscriber](#)

[Set-MgmtSvcNotificationSubscriber](#)

---

# Get-MgmtSvcResourceProviderConfiguration

---

## Get-MgmtSvcResourceProviderConfiguration

Gets a resource provider configuration.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcResourceProviderConfiguration [[-Name] <String[]> ] [-As <String> ] [-  
ConnectionString <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [  
<CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcResourceProviderConfiguration** cmdlet gets a resource provider configuration. If you run this cmdlet without specifying any parameters, all resource provider configurations are returned. To get a specific resource provider configuration, specify one or more values for the *Name* parameter.

### Parameters

#### **-As<String>**

Specifies an output format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-ConnectionString<String>**

Specifies an SQL connection string.

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EncryptionAlgorithm<String>**

Specifies an encryption algorithm.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EncryptionKey<String>**

Specifies an encryption key, as a hexadecimal string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String[]>**

Specifies an array of names of resource providers. You can use wildcards.

---

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Get all resource provider configurations

This command gets all resource provider configurations and returns their information in XML format.

```
PS C:\> Get-MgmtSvcResourceProviderConfiguration -As XmlString
```

### Example 2: Get specific resource provider configurations by their name

This command gets only the resource provider configurations named SqlServers and marketplace, and returns information about the configurations to the user.

```
PS C:\> Get-MgmtSvcResourceProviderConfiguration -Name "SqlServers","marketplace"
```

## Related topics

[New-MgmtSvcResourceProviderConfiguration](#)

[Add-MgmtSvcResourceProviderConfiguration](#)

[Remove-MgmtSvcResourceProviderConfiguration](#)



---

# Get-MgmtSvcSchema

---

## Get-MgmtSvcSchema

Gets registered schemas.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcSchema [[-Schema] <String[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcSchema** cmdlet returns a list of registered management service schemas. A schema groups a set of related SQL scripts. A schema represents the database objects that a resource provider uses or that are shared by features.

### Parameters

#### **-Schema<String[]>**

Specifies an array of component schemas. You can use wildcards.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

#### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

---

## Examples

### Example 1: Get all registered schemas

This command gets all of the management service schemas that are registered on the local computer.

```
PS C:\> Get-MgmtSvcSchema
```

MySQL

SQLServer

Usage

WebAppGallery

Notification

Management

Config

PortalAspNet

PortalNotification

### Related topics

[Get-MgmtSvcFeature](#)

[Get-MgmtSvcNamespace](#)

[Get-MgmtSvcEndpoint](#)

[Remove-MgmtSvcDatabaseUser](#)

[Install-MgmtSvcDatabase](#)

[Add-MgmtSvcDatabaseUser](#)

[Set-MgmtSvcDatabaseUser](#)

[Test-MgmtSvcDatabase](#)

---

# Get-MgmtSvcSetting

---

## Get-MgmtSvcSetting

Gets the local setting values for the management service component.

### Syntax

Parameter Set: Default

```
Get-MgmtSvcSetting [[-Namespace] <String[]> ] [[-Name] <String[]> ] [ <CommonParameters>]
```

### Detailed Description

The **Get-MgmtSvcSetting** cmdlet gets the local setting values for the management service component. The cmdlet returns the namespace, name, and value of the management service component.

### Parameters

#### **-Name<String[]>**

Specifies an array of names of management service settings. You can use wildcards.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	true

#### **-Namespace<String[]>**

Specifies an array of namespaces of management service settings. You can use wildcards.

Aliases	none
---------	------

---

Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Get the machine key setting

This command gets the machine key on the tenant site on the local computer.

```
PS C:\> Get-MgmtSvcSetting -Namespace "TenantSite" -Name "machineKey.decryptionKey"
```

## Related topics

[Set-MgmtSvcSetting](#)

[Protect-MgmtSvcConfiguration](#)

[Unprotect-MgmtSvcConfiguration](#)

[Get-MgmtSvcDatabaseSetting](#)

[Set-MgmtSvcDatabaseSetting](#)

---

# Initialize-MgmtSvcFeature

---

## Initialize-MgmtSvcFeature

Configures a management service feature.

### Syntax

Parameter Set: ConnectionParameters

```
Initialize-MgmtSvcFeature [-Name] <String> [[-Settings] <Hashtable> ] [-EnableCeip {No | None | Yes} ] [-Passphrase <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Initialize-MgmtSvcFeature [-Name] <String> [[-Settings] <Hashtable> ] [-ConnectionString <String> ] [-EnableCeip {No | None | Yes} ] [-Passphrase <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Initialize-MgmtSvcFeature** cmdlet configures a management service feature.

The feature can be initialized only if it is installed locally.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string. This parameter overrides Settings["connectionString"].

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-EnableCeip<EnableCeip>**

Enables, when set to Yes, participation in the Customer Experience Improvement Program (CEIP). To disable participation in CEIP, set this value to No. This parameter overrides Settings["enableCeip"].

The acceptable values for this parameter are:

No	
None	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a management service feature.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Passphrase<String>**

Specifies a passphrase.

The passphrase must have more than eight characters and contain at least one non-alphanumeric character. Remember to protect the passphrase. You cannot recover the passphrase. This parameter overrides Settings["configStorePassphrase"].

---

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a password. This parameter overrides Settings["dbAdminPassword"].

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides. This parameter overrides Settings["dbServer"].

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Settings<Hashtable>**

Specifies a collection of key/value pairs for the feature settings. When possible, you should use the named parameters instead of this parameter.

Aliases	none
Required?	false
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account. This parameter overrides Settings["dbAdminUserName"].

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Configure a management service feature

The first command specifies the settings for a management service feature, and stores the settings in the \$Settings variable.

The second command configures the management service feature named AdminSite by using the settings stored in \$Settings for the feature.

```
PS C:\> $Settings =  
@{dbServer='mysqlservermc';dbAdminUserName='sa';dbAdminPassword='#####';configStorePassphr  
ase='#####';}  
PS C:\> Initialize-MgmtSvcFeature -Name "AdminSite" -Settings $Settings -EnableCeip Yes
```

## Related topics

[Get-MgmtSvcFeature](#)

---

# Initialize-MgmtSvcProduct

---

## Initialize-MgmtSvcProduct

Configures features for the management service.

### Syntax

Parameter Set: ConnectionParameters

```
Initialize-MgmtSvcProduct -Passphrase <String> [-EnableCeip {No | None | Yes} ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Initialize-MgmtSvcProduct -Passphrase <String> [-ConnectionString <String> ] [-EnableCeip {No | None | Yes} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Initialize-MgmtSvcProduct** cmdlet configures each feature of a management service installation on a single computer by calling the **Initialize-MgmtSvcFeature** cmdlet for each feature installed on the local computer.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EnableCeip<EnableCeip>**

Enables, when set to Yes, participation in the Customer Experience Improvement Program (CEIP). To disable participation in CEIP, set this value to No.

The acceptable values for this parameter are:

No	
None	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Passphrase<String>**

Specifies the passphrase on the configuration store.

The passphrase should have more than eight characters and contain at least one non-alphanumeric character. Remember to protect the passphrase. You cannot recover the passphrase.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## -Confirm

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Configure features for the management service

This command initializes all features on the computer named ContosoSQL01.

```
PS C:\> Initialize-MgmtSvcProduct -Server "ContosoSQL01" -UserName "sa" -Password '^^^^^^' -Passphrase '#####' -EnableCeip Yes -Verbose
```

## Related topics

[Initialize-MgmtSvcFeature](#)

---

# Install-MgmtSvcDatabase

---

## Install-MgmtSvcDatabase

Creates a schema and database objects in a database.

### Syntax

Parameter Set: Default

```
Install-MgmtSvcDatabase -Schema <String> [-ConnectionString <String> ] [-Database <String> ]  
[-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The **Install-MgmtSvcDatabase** cmdlet creates schema and associated database objects. The cmdlet creates the database if it does not already exist. The cmdlet installs a database schema on the specific server from scripts consumed during portal installation.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

You can run this cmdlet repeatedly on the existing database without destroying data. You can specify a new or an existing database, and, depending on the schemas, you can install multiple schemas in the same database.

The **Initialize-MgmtSvcFeature** cmdlet calls this cmdlet to configure the required databases. Therefore, it is uncommon to call **Install-MgmtSvcDatabase** directly.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Schema<String>**

Specifies a schema. The cmdlet installs this schema.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none



---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Add a schema to the database**

This command adds the schema Management to the database.

```
PS C:\> Install-MgmtSvcDatabase -Schema "Management" -ConnectionString "Data Source=$env:ComputerName;Initial Catalog=Management;Integrated Security=SSPI"
```

## **Related topics**

[Test-MgmtSvcDatabase](#)

[Uninstall-MgmtSvcDatabase](#)

---

# New-MgmtSvcMachineKey

---

## New-MgmtSvcMachineKey

Creates a machine key element.

### Syntax

Parameter Set: Default

```
New-MgmtSvcMachineKey [-Base64] [-Decryption <String> ] [-DecryptionKeySize <Int32> ] [-Validation <String> ] [ <CommonParameters>]
```

### Detailed Description

The **New-MachineKey** cmdlet creates a <machineKey> configuration element for use in the web.config file. The **Initialize-MgmtSvcFeature** cmdlet calls this cmdlet to generate the initial keys stored during configuration.

A machine key can be a validation key to confirm the integrity of data, or a decryption key to encrypt or decrypt forms authentication data. This cmdlet generates a value in memory. It is recommended that you periodically rotate the machine keys. For example, once per year.

### Parameters

#### -Base64

Indicates that the validation and decryption values are Base64 encoded.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Decryption<String>**

Specifies an algorithm to encrypt and decrypt forms authentication data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-DecryptionKeySize<Int32>**

Specifies a key size, in bits, of the algorithm used to encrypt and decrypt forms authentication data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Validation<String>**

Specifies a hash algorithm used to validate data.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Create a machine key

This command creates a machine key configuration element by using the Hash-based Message Authentication Code (HMAC) SHA256 (HMACSHA256) for validation and the Advanced Encryption Standard (AES) encryption method for decryption.

```
PS C:\> ([xml](New-MgmtSvcMachineKey)).OuterXml
```

---

# New-MgmtSvcPassword

---

## New-MgmtSvcPassword

Creates a random password.

### Syntax

Parameter Set: Default

```
New-MgmtSvcPassword [-Length <Int32> ] [ <CommonParameters>]
```

### Detailed Description

The **New-MgmtSvcPassword** cmdlet generates a cryptographically strong password for use with other settings. This cmdlet generates a value in memory. The `Initialize-MgmtSvcFeature` cmdlet calls this cmdlet to generate the passwords used across Windows Azure Pack for Windows Server components. For example, passwords used for database connection strings and access to resource providers.

Use this cmdlet to generate a cryptographically secure random password for use in setting basic authorization credentials. You can use the returned string as the value for the *Value* parameter for the **Set-MgmtSvcSetting** cmdlet, or as the password value in a connection string.

### Parameters

#### **-Length<Int32>**

Specifies a password length. Choose a value between 8 and 128, in multiples of 4. The default length is 32 characters.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Create a password

This command creates a random password that contains 64 characters.

```
PS C:\> New-MgmtSvcPassword -Length 64
```

## Related topics

[Set-MgmtSvcSetting](#)

---

# New-MgmtSvcResourceProviderConfiguration

---

## New-MgmtSvcResourceProviderConfiguration

Creates a resource provider.

### Syntax

Parameter Set: Default

```
New-MgmtSvcResourceProviderConfiguration [[-InstanceId] <String> ] [[-Name] <String> ] [-AdminAuthenticationMode <AuthenticationMode> ] [-AdminAuthenticationPassword <String> ] [-AdminAuthenticationUserName <String> ] [-AdminForwardingAddress <String> ] [-AllowAnonymousAccess] [-AllowMultipleInstances] [-As <String> ] [-DisplayName <String> ] [-From <String> ] [-Input <String> ] [-InstanceDisplayName <String> ] [-NotificationAuthenticationMode <AuthenticationMode> ] [-NotificationAuthenticationPassword <String> ] [-NotificationAuthenticationUserName <String> ] [-NotificationForwardingAddress <String> ] [-ResourceProvider <ResourceProvider> ] [-TenantAuthenticationMode <AuthenticationMode> ] [-TenantAuthenticationPassword <String> ] [-TenantAuthenticationUserName <String> ] [-TenantForwardingAddress <String> ] [-TenantSourceUriTemplate <String> ] [-TenantTargetUriTemplate <String> ] [-Type <ResourceProviderType> ] [-UsageAuthenticationMode <AuthenticationMode> ] [-UsageAuthenticationPassword <String> ] [-UsageAuthenticationUserName <String> ] [-UsageForwardingAddress <String> ] [ <CommonParameters>]
```

### Detailed Description

The **New-MgmtSvcResourceProviderConfiguration** cmdlet creates a resource provider in memory. To add the resource provider, use the **Add-MgmtSvcResourceProviderConfiguration** cmdlet.

### Parameters

#### **-AdminAuthenticationMode<AuthenticationMode>**

Specifies the administrative authentication mode of a resource provider. Valid values for this parameter are:

- None
- Basic
- Windows

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AdminAuthenticationPassword<String>**

Specifies an administrator password to connect to a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AdminAuthenticationUserName<String>**

Specifies an administrator user name to connect to a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AdminForwardingAddress<String>**

Specifies an administrator forwarding address for a resource provider.

Aliases	none
---------	------



---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AllowAnonymousAccess**

Indicates that anonymous access is allowed to a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AllowMultipleInstances**

Indicates that the cmdlet allows multiple instances of a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-As<String>**

Specifies an output format.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DisplayName<String>**

Specifies the display name of a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-From<String>**

Specifies the sender address.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Input<String>**

Specifies input to a resource provider.

Aliases	none
---------	------

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-InstanceDisplayName<String>**

Specifies a display name for an instance of a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-InstanceId<String>**

Specifies an ID for an instance of a resource provider.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a resource provider

Aliases	none
---------	------

Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NotificationAuthenticationMode<AuthenticationMode>**

Specifies the notification authentication mode for a resource provider. Valid values for this parameter are:

- None
- Basic
- Windows

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-NotificationAuthenticationPassword<String>**

Specifies a notification password to connect to a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-NotificationAuthenticationUserName<String>**

Specifies a notification user name to connect to a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NotificationForwardingAddress<String>**

Specifies the notification forwarding address of a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-ResourceProvider<ResourceProvider>**

Specifies a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

---

## **-TenantAuthenticationMode<AuthenticationMode>**

Specifies the tenant authentication mode of a resource provider. Valid values for this parameter are:

- None
- Basic
- Windows

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-TenantAuthenticationPassword<String>**

Specifies the tenant password to connect to a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-TenantAuthenticationUserName<String>**

Specifies the tenant user name to connect to a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-TenantForwardingAddress<String>**

Specifies the tenant forwarding address of a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TenantSourceUriTemplate<String>**

Specifies the tenant source URI template of a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-TenantTargetUriTemplate<String>**

Specifies the tenant target URI template of a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Type<ResourceProviderType>**

Specifies the type of the resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UsageAuthenticationMode<AuthenticationMode>**

Specifies the usage authentication mode of a resource provider. Valid values for this parameter are:

- None
- Basic
- Windows

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UsageAuthenticationPassword<String>**

Specifies the usage password to connect to a resource provider.

Aliases	none
Required?	false



---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UsageAuthenticationUserName<String>**

Specifies the usage user name to connect to a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UsageForwardingAddress<String>**

Specifies the notification forwarding address of a resource provider.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### Example 1: Create a resource provider

The first command stores a connection string in the `$ConnectionString` variable.

The second command stores an encryption key in the `$EncryptionKey` variable.

The third command specifies the encryption algorithm AES and stores the value in the `$EncryptionAlgorithm` variable.

The fourth command stores a user name in the `$UserName` variable.

The fifth command stores a password in the `$Password` variable.

The sixth command uses the provided information to create a resource provider for SQL servers, and stores the resource provider object in the `$RP` variable.

The last command displays information about the new resource provider to the user. You can also use this variable to pass the resource provider object to other cmdlets, such as **Add-MgmtSvcResourceProviderConfiguration**.

```
PS C:\> $ConnectionString = ""
PS C:\> $EncryptionKey = "D576FCB3740049D44183C8BD6AB7979FB68DF253A1AFAB1BEDD987907358397D"
PS C:\> $EncryptionAlgorithm = "AES"
PS C:\> $UserName = "PattiFuller"
PS C:\> $Password = "passw0rd"
PS C:\> $RP = New-MgmtSvcResourceProviderConfiguration -Name 'RP01' `
-DisplayName 'Resource Provider 01' `
-AdminForwardingAddress "https://$Env:ComputerName`:30010/" `
-AdminAuthenticationMode 'Basic' `
-AdminAuthenticationUserName $UserName `
-AdminAuthenticationPassword $Password `
-TenantForwardingAddress "https://$Env:ComputerName`:30010/subscriptions" `
-TenantAuthenticationMode 'Basic' `
-TenantAuthenticationUserName $UserName `
-TenantAuthenticationPassword $Password `
-TenantSourceUriTemplate '{subid}/services/sqlservers/{*path}' `
-TenantTargetUriTemplate '{subid}/{*path}' `
-UsageForwardingAddress "https://$Env:ComputerName`:30010/" `
-UsageAuthenticationMode 'Basic' `
-UsageAuthenticationUserName $UserName `
-UsageAuthenticationPassword $Password `
-NotificationForwardingAddress "https://$Env:ComputerName`:30010/" `
-NotificationAuthenticationMode 'Basic' `
-NotificationAuthenticationUserName $UserName `
-NotificationAuthenticationPassword $Password
PS C:\> $RP
```

---

## **Related topics**

[Add-MgmtSvcResourceProviderConfiguration](#)

[Get-MgmtSvcResourceProviderConfiguration](#)

[Remove-MgmtSvcResourceProviderConfiguration](#)

---

# New-MgmtSvcSelfSignedCertificate

---

## New-MgmtSvcSelfSignedCertificate

Creates a self-signed certificate.

### Syntax

Parameter Set: Default

```
New-MgmtSvcSelfSignedCertificate -StoreLocation {CurrentUser | LocalMachine} -StoreName {AddressBook | AuthRoot | CertificateAuthority | Disallowed | My | Root | TrustedPeople | TrustedPublisher} -Subject <String> [-Password <SecureString> ] [ <CommonParameters>]
```

### Detailed Description

The **New-MgmtSvcSelfSignedCertificate** cmdlet creates a self-signed certificate.

### Parameters

#### **-Password<SecureString>**

Specifies a password as a secure string object.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### **-StoreLocation<StoreLocation>**

Specifies the store location for the certificate. Valid values for this parameter are:

- CurrentUser
- LocalMachine

---

The acceptable values for this parameter are:

CurrentUser	
LocalMachine	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-StoreName<StoreName>**

Specifies the store name for the certificate. Valid values for this parameter are:

- AddressBook
- AuthRoot
- CertificateAuthority
- Disallowed
- My
- Root
- TrustedPeople
- TrustedPublisher

The acceptable values for this parameter are:

AddressBook	
AuthRoot	
CertificateAuthority	
Disallowed	
My	
Root	
TrustedPeople	
TrustedPublisher	

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Subject<String>**

Specifies a subject for the certificate.

Aliases	none
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, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

```
PS C:\> $Password = ConvertTo-SecureString "PassWord01!" -AsPlainText -Force
PS C:\> New-MgmtSvcSelfSignedCertificate -StoreLocation "LocalMachine" -StoreName "Root" -
Subject "CN=Admin,DC=Contoso,DC=com" -Password $Password
```

---

# Protect-MgmtSvcConfiguration

---

## Protect-MgmtSvcConfiguration

Encrypts sections of the web.config file for a namespace.

### Syntax

Parameter Set: Default

```
Protect-MgmtSvcConfiguration [-Namespace] <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The **Protect-MgmtSvcConfiguration** cmdlet encrypts sections of the web.config file for a specified namespace. The cmdlet encrypts the sections of the web.config file that contain secrets, such as connectionStrings, appSettings, and machineKey. To access these settings, use the passphrase parameters described in the **Get-MgmtSvcSetting** and **Set-MgmtSvcSetting** cmdlets. However, you do not need to access the settings by using the cmdlets to unprotect the configuration. This is only necessary to access the web.config file in an editor.

Run this cmdlet on the computer that hosts the web.config file.

### Parameters

#### **-Namespace<String>**

Specifies a namespace.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

---

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Encrypt the web.config file for a namespace**

This command encrypts the sections of the web.config file that contains secrets, such as connectionStrings, appSettings, and machineKey, for the namespace AdminSite.

```
PS C:\> Protect-MgmtSvcConfiguration -Namespace "AdminSite"
```



---

## Related topics

[Unprotect-MgmtSvcConfiguration](#)

[Test-MgmtSvcProtectedConfiguration](#)

[Get-MgmtSvcSetting](#)

[Set-MgmtSvcSetting](#)

---

# Remove-MgmtSvcAdminUser

---

## Remove-MgmtSvcAdminUser

Removes an administrative principal from the database.

### Syntax

Parameter Set: ConnectionParameters

```
Remove-MgmtSvcAdminUser [-Principal] <String> [-Database <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set:ConnectionString

```
Remove-MgmtSvcAdminUser [-Principal] <String> [-ConnectionString <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Remove-MgmtSvcAdminUser** cmdlet removes an administrative user from the database.

If this cmdlet is run on the computer on which the Admin API service is installed and the Web.config file contains values for EncryptionKey and EncryptionAlgorithm, then those values are used.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionAlgorithm<String>**

Specifies an encryption algorithm.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionKey<String>**

Specifies an encryption key, as a hexadecimal string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Principal<String>**

Specifies an administrative user or group.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Remove a principal by using a connection string

This command removes the principal named Patti Fuller by using a connection string to connect to the database.

```
PS C:\> Remove-MgmtSvcAdminUser -Principal "PattiFuller@Contoso.com" -ConnectionString  
'Server=.\sqlexpress;Initial Catalog=Contoso.MgmtSvc.Store;User Id=sa;Password=PassWord;'
```

## Related topics

[Add-MgmtSvcAdminUser](#)

[Get-MgmtSvcAdminUser](#)

---

# Remove-MgmtSvcDatabaseUser

---

## Remove-MgmtSvcDatabaseUser

Removes a principal from the database.

### Syntax

Parameter Set: Default

```
Remove-MgmtSvcDatabaseUser -Schema <String> -User <String> [-ConnectionString <String> ] [-Database <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Remove-MgmtSvcDatabaseUser** cmdlet removes a principal account for a specified schema from the database. Specify the user name and schema.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

Use this cmdlet when uninstalling Windows Azure Pack for Windows Server. When you uninstall the product, you must uninstall the databases and remove the database users.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Schema<String>**

Specifies a schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-User<String>**

Specifies the user name of a principal. The cmdlet removes this account from the database for the specified schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **Example 1: Remove a user account**

This command removes the user named MgmtSvc-Usage from the Usage schema in the database named Contoso.MgmtSvc.Usage. The command specifies the computer that runs SQL server as ContosoSQLServer. The command also specifies a user named SysAdmin, which has permissions to make this change, along with the password for that account.

---

```
PS C:\> Remove-MgmtSvcDatabaseUser -Schema "Usage" -User "MgmtSvc-Usage" -Database
"Microsoft.MgmtSvc.Usage" -Password "PassWord!" -Server "ContosoSQLServer" -UserName
"SysAdmin"
```

## Related topics

[Add-MgmtSvcDatabaseUser](#)

[Set-MgmtSvcDatabaseUser](#)

---

# Remove-MgmtSvcNotificationSubscriber

---

## Remove-MgmtSvcNotificationSubscriber

Removes a notification subscriber.

### Syntax

Parameter Set: Default

```
Remove-MgmtSvcNotificationSubscriber [[-Name] <String> ] [-ConnectionString <String> ] [-Database <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Remove-MgmtSvcNotificationSubscriber** cmdlet removes a notification subscriber from the database.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a notification subscriber.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Remove a notification subscriber

This command removes the notification subscriber named Billing from the usage service.

```
PS C:\> Remove-MgmtSvcNotificationSubscriber -Name "Billing"
```

## Related topics

[Get-MgmtSvcNotificationSubscriber](#)

[Set-MgmtSvcNotificationSubscriber](#)

---

# Remove-MgmtSvcResourceProviderConfiguration

---

## Remove-MgmtSvcResourceProviderConfiguration

Removes a resource provider from a management store database.

### Syntax

Parameter Set: Default

```
Remove-MgmtSvcResourceProviderConfiguration [-Name] <String> [-InstanceId] <String> [-As <String> ] [-ConnectionString <String> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Remove-MgmtSvcResourceProviderConfiguration** cmdlet removes an entry for a resource provider from the management store database. You can run this cmdlet from any computer in the deployment.

If this cmdlet is run on the computer on which the Admin API service is installed and Web.config file contains values for ConnectionString, EncryptionKey, and EncryptionAlgorithm, then those values are used.

### Parameters

#### **-As<String>**

Specifies an output format.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionAlgorithm<String>**

Specifies an encryption algorithm.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionKey<String>**

Specifies an encryption key, as a hexadecimal string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-InstanceId<String>**

Specifies an ID for an instance of a resource provider.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a resource provider.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Remove a resource provider

This command removes the resource provider with the name RP01 and instance ID of 0602c550-0853-48fc-bfbb-dc1f84ac08a3.

```
PS C:\> Remove-MgmtSvcResourceProviderConfiguration -Name "RP01" -InstanceId "0602c550-0853-48fc-bfbb-dc1f84ac08a3"
```

## Related topics

[New-MgmtSvcResourceProviderConfiguration](#)

[Add-MgmtSvcResourceProviderConfiguration](#)

[Get-MgmtSvcResourceProviderConfiguration](#)

---

# Reset-MgmtSvcPassphrase

---

## Reset-MgmtSvcPassphrase

Changes the passphrase on a configuration store.

### Syntax

Parameter Set: Default

```
Reset-MgmtSvcPassphrase -NewPassphrase <String> -OldPassphrase <String> [-ConnectionString <String> ] [-Database <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Reset-MgmtSvcPassphrase** cmdlet changes the passphrase on a configuration store. You must provide the old passphrase in addition to the new passphrase.

The passphrase should have more than eight characters and contain at least one non-alphanumeric character. Remember to protect the passphrase. You cannot recover the passphrase.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-NewPassphrase<String>**

Specifies a new passphrase.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-OldPassphrase<String>**

Specifies the old passphrase.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Change the passphrase on a configuration store**

This command changes the passphrase on the configuration store.

```
PS C:\> Reset-MgmtSvcPassphrase -OldPassphrase '$##$##' -NewPassphrase 'p@assw0rd1'
```

---

## Related topics

[Get-MgmtSvcDatabaseSetting](#)

[Set-MgmtSvcDatabaseSetting](#)

[Test-MgmtSvcPassphrase](#)



---

# Set-MgmtSvcCeip

---

## Set-MgmtSvcCeip

Updates CEIP settings for the local computer.

### Syntax

Parameter Set: Default

```
Set-MgmtSvcCeip [-EnableCeip {No | None | Yes} ] [ <CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcCeip** cmdlet updates the Customer Experience Improvement Program (CEIP) settings for the local computer.

**NOTE:** You must run this cmdlet in a Windows PowerShell session that you have run as Administrator.

### Parameters

#### **-EnableCeip<EnableCeip>**

Enables, when set to Yes, participation in CEIP. To disable participation in CEIP, set this value to No.

The acceptable values for this parameter are:

No	
None	
Yes	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?
-----------------------------

false
-------

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Enable participation in CEIP

This command enables participation in CEIP.

```
PS C:\> Set-MgmtSvcCeip -EnableCeip Yes
```

---

# Set-MgmtSvcDatabaseSetting

---

## Set-MgmtSvcDatabaseSetting

Writes settings to the configuration database.

### Syntax

Parameter Set: Default

```
Set-MgmtSvcDatabaseSetting [-Namespace] <String> [-Name] <String> [[-Value] <String> ] [-  
ConnectionString <String> ] [-Database <String> ] [-Force] [-Passphrase <String> ] [-  
Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcDatabaseSetting** cmdlet writes configuration settings for a management service component to the database. A setting consists of a namespace, a name, and a value.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Force**

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Name<String>**

Specifies the name of a setting.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Namespace<String>**

Specifies the namespace of a setting.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Passphrase<String>**

Specifies a passphrase. The database encrypts items in the configuration store by using this passphrase.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Value<String>**

Specifies the value of a setting.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## -Confirm

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### Example 1: Write a setting to the database

The first command creates a machine key by using the **New-MgmtSvcMachineKey** cmdlet, and stores it in the \$MK variable.

The second command stores a connection string in the \$ConnectionString variable.

The last command writes the namespace, name, and value to the database. The command uses the connection string stored in \$ConnectionString and the machine key stored in \$MK. The command also includes a passphrase.

```
PS C:\> $MK = New-MgmtSvcMachineKey
PS C:\> $ConnectionString = 'Data Source=mysqlserver;Initial
Catalog=Microsoft.MgmtSvc.Config;User ID=sa;Password=Book67pp'
```

---

```
PS C:\> Set-MgmtSvcDatabaseSetting -ConnectionString $ConnectionString -Namespace
"TenantSite" -Name "machineKey.decryptionKey" -Value $MK.Attributes("decryptionKey").Value
-Force -Passphrase "PassPhrase01!"
```

## Related topics

[Get-MgmtSvcDatabaseSetting](#)

[Test-MgmtSvcPassphrase](#)

[Reset-MgmtSvcPassphrase](#)

[New-MgmtSvcMachineKey](#)



---

# Set-MgmtSvcDatabaseUser

---

## Set-MgmtSvcDatabaseUser

Modifies a principal user in the database.

### Syntax

Parameter Set: Default

```
Set-MgmtSvcDatabaseUser -Schema <String> -User <String> -UserPassword <String> [-  
ConnectionString <String> ] [-Database <String> ] [-Password <String> ] [-Server <String> ]  
[-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcDatabaseUser** cmdlet modifies a principal account in a schema in the database. You can use this cmdlet to reset the password for the user. The **Initialize-MgmtSvcFeature** cmdlet sets all of the required users for Windows Azure Pack for Windows Server. Therefore you only need to use this cmdlet to set additional users.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Schema<String>**

Specifies a schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-User<String>**

Specifies the user name of a principal. The cmdlet modifies this account in the database for the specified schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-UserPassword<String>**

Specifies a password for the principal account specified by the *User* parameter. The cmdlet modifies this account in the database for the specified schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## Examples

### Example 1: Change password for a user

This command changes the password for a user named MgmtSvc-UsageCollector in the Usage schema.

```
PS C:\> Set-MgmtSvcDatabaseUser -Schema "Usage" -User "MgmtSvc-UsageCollector" -UserPassword  
"PassWord!" -ConnectionString 'Data Source=mysqlserver;Initial  
Catalog=Microsoft.MgmtSvc.Usage;User ID=SysAdmin;Password=Book67%&'
```

## Related topics

[Add-MgmtSvcDatabaseUser](#)

[Remove-MgmtSvcDatabaseUser](#)

---

# Set-MgmtSvcFqdn

---

## Set-MgmtSvcFqdn

Sets the FQDN for an Admin or Tenant site.

### Syntax

Parameter Set: ConnectionParameters

```
Set-MgmtSvcFqdn [-Namespace] <String[]> [-FullyQualifiedDomainName <String> ] [-Password <String> ] [-Port <Int32> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

Parameter Set: ConnectionString

```
Set-MgmtSvcFqdn [-Namespace] <String[]> [-ConnectionString <String> ] [-FullyQualifiedDomainName <String> ] [-Port <Int32> ] [-Confirm] [-WhatIf] [<CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcFqdn** cmdlet sets the Fully Qualified Domain Name (FQDN) for an Admin or Tenant site.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-FullyQualifiedDomainName<String>**

Specifies a Fully Qualified Domain Name (FQDN).

Aliases	FQDN
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Namespace<String[]>**

Specifies an array of namespaces. Valid values are: AdminSite, TenantSite.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Port<Int32>**

Specifies a port number.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## -Confirm

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Set the FQDN of a tenant site.

This command sets the FQDN for a tenant site.

```
PS C:\> Set-MgmtSvcFqdn -Namespace "TenantSite" -FullyQualifiedDomainName  
"Computer01.Contoso.com" -Password "PassWord01!" -Port 30081 -Server "SQLComputer01" -  
UserName "PattiFuller"
```

## Related topics

[Get-MgmtSvcFqdn](#)

---

# Set-MgmtSvcIdentityProviderSettings

---

## Set-MgmtSvcIdentityProviderSettings

Configures an administrator authentication site or a tenant authentication site to use AD FS.

### Syntax

Parameter Set: ConnectionParameters

```
Set-MgmtSvcIdentityProviderSettings [-Target] <String[]> [-MetadataEndpoint] <Uri> [-ConfigureSecondary] [-DisableCertificateValidation] [-ManagementConnectionString <String> ] [-Password <String> ] [-PortalConnectionString <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Set-MgmtSvcIdentityProviderSettings [-Target] <String[]> [-MetadataEndpoint] <Uri> [-ConfigureSecondary] [-ConnectionString <String> ] [-DisableCertificateValidation] [-ManagementConnectionString <String> ] [-PortalConnectionString <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcIdentityProviderSettings** cmdlet configures an administrator authentication site or a tenant authentication site to use Active Directory Federation Services (AD FS). Specify one or more namespaces and an endpoint for metadata.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### -ConfigureSecondary

Indicates that secondary identity provider settings are set.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DisableCertificateValidation**

Indicates that the cmdlet disables certificate validation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ManagementConnectionString<String>**

Specifies a connection string for the Admin Auth Site.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MetadataEndpoint<Uri>**

Specifies an endpoint for identity provider metadata.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PortalConnectionString<String>**

Specifies a connection string for the Tenant Auth Site.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Target<String[]>**

Specifies the target site. Valid values are:

- Membership. This value indicates that the target is the Tenant Auth Site.
- Windows. This value indicates that the target is the Admin Auth Site.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### Example 1: Configure a tenant authentication site

The first command stores a connection string in the \$ConnectionString variable.

The second command configures a tenant authentication. The command uses the connection string stored in the \$ConnectionString variable. The command also specifies an endpoint for the metadata and disables certificate validation.

```
PS C:\> $ConnectionString = 'Data Source=rd-sdfre4;Initial
Catalog=Microsoft.MgmtSvc.Config;User ID=SysAdmin;Password=PassWord'
PS C:\> Set-MgmtSvcIdentityProviderSettings -Target Management -MetadataEndpoint
"https://Server07.Contoso.com/FederationMetadata/2007-06/FederationMetadata.xml" -
PortalConnectionString $ConnectionString -DisableCertificateValidation
```

---

# Set-MgmtSvcNotificationSubscriber

---

## Set-MgmtSvcNotificationSubscriber

Adds or updates a notification subscriber.

### Syntax

Parameter Set: ByProperties

```
Set-MgmtSvcNotificationSubscriber -Endpoint <Uri> -Name <String> [-AuthenticationMode {None | Basic} ] [-AuthenticationPassword <String> ] [-AuthenticationUsername <String> ] [-Enabled <Boolean> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Force] [-SubscriberType {BillingService | MandatoryService | OptionalService} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ByObject

```
Set-MgmtSvcNotificationSubscriber [[-NotificationSubscriber] <NotificationSubscriber> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Force] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionParameters

```
Set-MgmtSvcNotificationSubscriber -Endpoint <Uri> -Name <String> [-AuthenticationMode {None | Basic} ] [-AuthenticationPassword <String> ] [-AuthenticationUsername <String> ] [-Database <String> ] [-Enabled <Boolean> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Force] [-NotificationSubscriber <NotificationSubscriber> ] [-Password <String> ] [-Server <String> ] [-SubscriberType {BillingService | MandatoryService | OptionalService} ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Set-MgmtSvcNotificationSubscriber -Endpoint <Uri> -Name <String> [-AuthenticationMode {None | Basic} ] [-AuthenticationPassword <String> ] [-AuthenticationUsername <String> ] [-ConnectionString <String> ] [-Enabled <Boolean> ] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Force] [-NotificationSubscriber <NotificationSubscriber> ] [-SubscriberType {BillingService | MandatoryService | OptionalService} ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcNotificationSubscriber** cmdlet adds or updates a notification subscriber. If the subscriber does not exist, the cmdlet adds a subscriber. To update an existing notification subscriber, use the *Force* parameter.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*,



---

*UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

This cmdlet also assumes that the usage service is installed on the local computer. If the usage service is installed on another computer, you must use the *EncryptionKey* and *EncryptionAlgorithm* parameters.

## Parameters

### **-AuthenticationMode<NotificationSubscriberAuthMode>**

Specifies the authentication mode. Valid values for this parameter are:

- None
- Basic
- Windows

The acceptable values for this parameter are:

None	
Basic	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-AuthenticationPassword<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-AuthenticationUsername<String>**

Specifies a user name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Enabled<Boolean>**

Indicates whether the notification subscriber is enabled.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EncryptionAlgorithm<String>**

Specifies an encryption algorithm.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-EncryptionKey<String>**

Specifies an encryption key, as a hexadecimal string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Endpoint<Uri>**

Specifies a URI for the notification subscriber.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Force**

Indicates that the cmdlet updates an existing subscriber.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Name<String>**

Specifies the name of a notification subscriber.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false

Accept Wildcard Characters?	false
-----------------------------	-------

### **-NotificationSubscriber<NotificationSubscriber>**

Specifies a notification subscriber object. To obtain a notification subscriber object, use the **Get-MgmtSvcNotificationSubscriber** cmdlet. If you specify a notification subscriber object, the cmdlet changes that object, regardless of the *Name* parameter.

Aliases	none
Required?	false
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByValue)
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-SubscriberType<NotificationSubscriberType>**

Specifies a notification subscriber type. Valid values for this parameter are:

- BillingService
- MandatoryService
- OptionalService

The acceptable values for this parameter are:

BillingService	
MandatoryService	
OptionalService	

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **Example 1: Add a notification subscriber**

This command adds a notification subscriber to the database. The command specifies a name and subscriber type for the notification subscriber.

```
PS C:\> Set-MgmtSvcNotificationSubscriber -AuthenticationMode None -Enabled $False -Endpoint https://localhost/ -Name "Billing" -SubscriberType BillingService
```

---

## Example 2: Modify a notification subscriber

The first command gets the notification subscriber named Billing and stores the object in the \$Subscriber variable.

The second, third, and fourth commands make changes to the object stored in the \$Subscriber variable.

The last command saves the modified notification subscriber object. Using the *Force* parameter indicates that the existing subscriber is updated.

```
PS C:\> $Subscriber = Get-MgmtSvcNotificationSubscriber -Name "Billing"PS C:\>
$Subscriber.AuthenticationMode = 'Basic'PS C:\> $Subscriber.AuthenticationUsername =
'MyUser'PS C:\> $Subscriber.AuthenticationPassword = 'PassWord'PS C:\> Set-
MgmtSvcNotificationSubscriber -NotificationSubscriber $Subscriber -Force
```

## Related topics

[Get-MgmtSvcNotificationSubscriber](#)

[Remove-MgmtSvcNotificationSubscriber](#)



---

# Set-MgmtSvcPassphrase

---

## Set-MgmtSvcPassphrase

Sets the passphrase for a configuration database.

### Syntax

Parameter Set: ConnectionParameters

```
Set-MgmtSvcPassphrase -Passphrase <String> [-Database <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Set-MgmtSvcPassphrase -Passphrase <String> [-ConnectionString <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcPassphrase** cmdlet sets the passphrase that the management service uses to encrypt sensitive data in the configuration database.

A passphrase can only be set once. To replace the passphrase, use the **Reset-MgmtSvcPassPhrase** cmdlet.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none

---

Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Passphrase<String>**

Specifies a passphrase.

The passphrase must contain at least eight characters and contain at least one non-alphanumeric character. Remember to protect the passphrase. You cannot recover the passphrase.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Set a passphrase**

This command sets the passphrase for the configuration store on the computer named Contoso01.

```
PS C:\> Set-MgmtSvcPassphrase -Server "Contoso01" -UserName "sa" -Password '*****' -  
Passphrase '*****'
```

## **Related topics**

[Reset-MgmtSvcPassphrase](#)

[Test-MgmtSvcPassphrase](#)

---

# Set-MgmtSvcRelyingPartySettings

---

## Set-MgmtSvcRelyingPartySettings

Configures the management portal for administrators and management portal for tenants to use AD FS.

### Syntax

Parameter Set: ConnectionParameters

```
Set-MgmtSvcRelyingPartySettings [-Target] <String[]> [-MetadataEndpoint] <Uri> [-DisableCertificateValidation] [-ManagementConnectionString <String> ] [-Password <String> ] [-PortalConnectionString <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

Parameter Set: ConnectionString

```
Set-MgmtSvcRelyingPartySettings [-Target] <String[]> [-MetadataEndpoint] <Uri> [-ConnectionString <String> ] [-DisableCertificateValidation] [-ManagementConnectionString <String> ] [-PortalConnectionString <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcRelyingPartySettings** cmdlet configures the management portal for administrators and management portal for tenants to use Active Directory Federation Services (AD FS). Specify one or more namespaces and an endpoint for metadata.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false

---

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-DisableCertificateValidation**

Indicates that the cmdlet disables certificate validation.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-ManagementConnectionString<String>**

Specifies a connection string for the Admin Site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-MetadataEndpoint<Uri>**

Specifies an endpoint for identity provider metadata.

Aliases	none
Required?	true

---

Position?	2
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-PortalConnectionString<String>**

Specifies a connection string for the Tenant Site.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false

Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Target<String[]>**

Specifies the target site. Valid values are:

-- Admin. This value indicates that the target is the Admin Site, Admin API, and Tenant API.

-- Tenant. This value indicates that the target is the Tenant site and Tenant API.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Confirm**

Prompts you for confirmation before executing the command.



Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Configure a tenant portal**

The first command stores a connection string in the \$ConnectionString variable.

The second command specifies the target as a Tenant and specifies an endpoint. The command uses the connection string stored in the \$ConnectionString variable and disables certificate validation.

```
PS C:\> $ConnectionString = 'Data Source=rd-sdfre4;Initial
Catalog=Microsoft.MgmtSvc.Config;User ID=SysAdmin;Password=Zoom2345'
PS C:\> Set-MgmtSvcRelyingPartySettings -Target Tenant -MetadataEndpoint
"https://Server07.Contoso.com/FederationMetadata/2007-06/FederationMetadata.xml" -
ConnectionString $ConnectionString -DisableCertificateValidation
```

---

# Set-MgmtSvcSetting

---

## Set-MgmtSvcSetting

Changes settings of a management service component in the local web.config file.

### Syntax

Parameter Set: Default

```
Set-MgmtSvcSetting [-Namespace] <String> [-Name] <String> [[-Value] <String> ] [-Encode] [-SkipUpdateHash] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Set-MgmtSvcSetting** cmdlet changes configuration settings for a management service component. You can use this cmdlet to change the values for the specified settings in the specified namespace. You can also make updates to machine keys and credentials, and enable server certificate validation.

You must run this cmdlet on the computer that hosts the management services.

### Parameters

#### -Encode

Indicates that the management service encodes the setting.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### -Name<String>

Specifies the name of the management service setting.

Aliases	none
Required?	true
Position?	2
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

### **-Namespace<String>**

Specifies an array of namespaces of management service settings.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

### **-SkipUpdateHash**

Indicates that the cmdlet does not update the hash table when it modifies a setting.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Value<String>**

Specifies the value of a setting.

Aliases	none
Required?	false
Position?	3
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

---

## Examples

### Example 1: Enable server certificate validation

This command enables server certificate validation for the AdminSite namespace.

```
PS C:\> Set-MgmtSvcSetting -Namespace "AdminSite" -Name DisableCertificateValidation" -Value  
False
```

### Related topics

[Get-MgmtSvcSetting](#)

[Protect-MgmtSvcConfiguration](#)

[Unprotect-MgmtSvcConfiguration](#)

---

# Test-MgmtSvcDatabase

---

## Test-MgmtSvcDatabase

Verifies database installation.

### Syntax

Parameter Set: Default

```
Test-MgmtSvcDatabase -Schema <String> [-ConnectionString <String> ] [-Database <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [ <CommonParameters>]
```

### Detailed Description

The **Test-MgmtSvcDatabase** cmdlet verifies installation of a management service database. The cmdlet returns the version of the installed schema.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Schema<String>**

Specifies a schema. The cmdlet verifies installation of this schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about CommonParameters](#).

## **Examples**

### **Example 1: Verify a schema installation**

This command verifies that the schema named Config is installed.

```
PS C:\> Test-MgmtSvcDatabase -Schema "Config" -ConnectionString 'Data
Source=mysqlserver;Initial Catalog=Microsoft.MgmtSvc.Config;User
ID=SysAdmin;Password=Zoom2345'
```



---

## Related topics

[Install-MgmtSvcDatabase](#)

[Uninstall-MgmtSvcDatabase](#)

---

# Test-MgmtSvcPassphrase

---

## Test-MgmtSvcPassphrase

Validates the passphrase on a configuration database.

### Syntax

Parameter Set: Default

```
Test-MgmtSvcPassphrase -Passphrase <String> [-ConnectionString <String> ] [-Database <String> ] [-Force] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Test-MgmtSvcPassphrase** cmdlet validates the passphrase that the management service uses to encrypt sensitive data in the configuration database. The passphrase is always valid if it doesn't exist and complies with the complexity requirements. The passphrase should have more than eight characters and contain at least one non-alphanumeric character. Additionally, **Test-MgmtSvcPassphrase** notifies you whether the passphrase was found.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false

---

Accept Wildcard Characters?	false
-----------------------------	-------

### **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Force**

Performs the action without a confirmation message.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Passphrase<String>**

Specifies a passphrase.

The passphrase should have more than eight characters and contain at least one non-alphanumeric character. Remember to protect the passphrase. You cannot recover the passphrase. You can use this cmdlet to set the passphrase for the first time in a deployment by specifying the *Force* parameter.

Aliases	none
Required?	true
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

### **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named

---

Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Validate a passphrase**

This command validates the passphrase for the configuration store on the computer named Contoso01.

---

```
PS C:\> Test-MgmtSvcPassphrase -Server "Contoso01" -UserName "sa" -Password '*****' -  
Passphrase '*****'
```

## Related topics

[Get-MgmtSvcDatabaseSetting](#)

[Set-MgmtSvcDatabaseSetting](#)

[Reset-MgmtSvcPassphrase](#)

---

# Test-MgmtSvcProtectedConfiguration

---

## Test-MgmtSvcProtectedConfiguration

Tests whether a configuration is protected.

### Syntax

Parameter Set: Default

```
Test-MgmtSvcProtectedConfiguration [-Namespace] <String> [ <CommonParameters>]
```

### Detailed Description

The **Test-MgmtSvcProtectedConfiguration** cmdlet tests whether the sections of the Web.config file that contain secrets are protected. If the configuration for the namespace is protected, a value of True is returned. If the configuration is not protected, a value of False is returned. The cmdlet displays a warning if the configuration is not protected.

It is recommended that you keep your Web.config file protected. To protect a configuration, use the **Protect-MgmtSvcConfiguration** cmdlet. You can unprotect your configuration to update the Web.config file by using the **Unprotect-MgmtSvcConfiguration**. However, you should then return the configuration to a protected state.

### Parameters

#### **-Namespace<String>**

Specifies a namespace.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

---

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Test a protected configuration

This command tests the protected configuration for the namespace AdminSite.

```
PS C:\> Test-MgmtSvcProtectedConfiguration -Namespace "AdminSite"
```

## Related topics

[Protect-MgmtSvcConfiguration](#)

[Unprotect-MgmtSvcConfiguration](#)



---

# Uninstall-MgmtSvcDatabase

---

## Uninstall-MgmtSvcDatabase

Removes schema and database objects from a database.

### Syntax

Parameter Set: Default

```
Uninstall-MgmtSvcDatabase -Schema <String> [-ConnectionString <String> ] [-Database <String> ] [-Password <String> ] [-Server <String> ] [-UserName <String> ] [-Confirm] [-WhatIf] [ <CommonParameters>]
```

### Detailed Description

The **Uninstall-MgmtSvcDatabase** cmdlet removes the specified schema and associated database objects from the database. This cmdlet does not remove the database.

You can run this cmdlet from any computer in the deployment. However, this cmdlet assumes that the database is on the local computer. If the database is on another computer, you must use the *Server*, *UserName*, *Password*, and *Database* parameters, or a SQL connection string. If you specify a connection string by using the *ConnectionString* parameter, that value takes precedence over the *Server*, *UserName*, *Password*, and *Database* parameters.

Use this cmdlet when uninstalling Windows Azure Pack for Windows Server. When you uninstall the product, you must uninstall the databases and remove the database users.

### Parameters

#### **-ConnectionString<String>**

Specifies an SQL connection string.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Database<String>**

Specifies a database name.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Password<String>**

Specifies a password.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Schema<String>**

Specifies a schema. The cmdlet removes this schema.

Aliases	none
Required?	true
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-Server<String>**

Specifies the name of the computer on which the SQL database resides.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-UserName<String>**

Specifies the name of a user account.

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## -WhatIf

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## <CommonParameters>

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## Examples

### Example 1: Remove a schema from the database

This command removes the schema named Usage and its associated database objects from the database

```
PS C:\> Uninstall-MgmtSvcDatabase -Schema "Usage" -ConnectionString 'Data
Source=mysqlserver;Initial Catalog=Microsoft.MgmtSvc.Usage;User
ID=SysAdmin;Password=Book67pp'
```

## Related topics

[Install-MgmtSvcDatabase](#)

[Test-MgmtSvcDatabase](#)

---

# Unprotect-MgmtSvcConfiguration

---

## Unprotect-MgmtSvcConfiguration

Decrypts sections of the web.config file for a namespace.

### Syntax

Parameter Set: Default

```
Unprotect-MgmtSvcConfiguration [-Namespace] <String> [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Detailed Description

The **Unprotect-MgmtSvcConfiguration** cmdlet decrypts sections of the web.config file for a specified namespace. The cmdlet decrypts the sections of the web.config file that contains secrets, such as connectionStrings, appSettings, and machineKey. To encrypt these sections, use the **Protect-MgmtSvcConfiguration** cmdlet.

Run this cmdlet on the computer that hosts the web.config file.

### Parameters

#### **-Namespace<String>**

Specifies a namespace.

Aliases	none
Required?	true
Position?	1
Default Value	none
Accept Pipeline Input?	True (ByPropertyName)
Accept Wildcard Characters?	false

#### **-Confirm**

Prompts you for confirmation before executing the command.

---

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

### **Example 1: Decrypt the web.config file for a namespace**

This command decrypts the connectionStrings and appSettings sections of the web.config file for the namespace AdminSite.

```
PS C:\> Unprotect-MgmtSvcConfiguration -Namespace "AdminSite"
```

## **Related topics**

[Protect-MgmtSvcConfiguration](#)

[Test-MgmtSvcProtectedConfiguration](#)

[Get-MgmtSvcSetting](#)

[Set-MgmtSvcSetting](#)

---

# Update-MgmtSvcV1Data

---

## Update-MgmtSvcV1Data

### Syntax

Parameter Set: Default

```
Update-MgmtSvcV1Data [-ConnectionString <String> ] [-DisableCertificateValidation] [-EncryptionAlgorithm <String> ] [-EncryptionKey <String> ] [-Confirm] [-WhatIf] [  
<CommonParameters>]
```

### Parameters

#### -ConnectionString<String>

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

#### -DisableCertificateValidation

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

---

## **-EncryptionAlgorithm<String>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-EncryptionKey<String>**

Aliases	none
Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **-Confirm**

Prompts you for confirmation before executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false



---

## **-WhatIf**

Describes what would happen if you executed the command without actually executing the command.

Required?	false
Position?	named
Default Value	none
Accept Pipeline Input?	false
Accept Wildcard Characters?	false

## **<CommonParameters>**

This cmdlet supports the common parameters: -Verbose, -Debug, -ErrorAction, -ErrorVariable, -OutBuffer, and -OutVariable. For more information, see [about\\_CommonParameters](#).

## **Examples**

**PS C:\>**

**PS C:\>**

