



Windows PowerShell Get-Help on Cmdlet 'Invoke-SqlNotebook'

PS:\>Get-HELP Invoke-SqlNotebook -Full

NAME

Invoke-SqlNotebook

SYNOPSIS

Executes a SQL Notebook file (.ipynb) and outputs the materialized notebook.

SYNTAX

Invoke-SqlNotebook [-AccessToken <PSObject>] [-Credential <PSCredential>] [-Database <Object>] [-Encrypt {Mandatory | Optional | Strict}] [-Force]

[-HostNameInCertificate <String>] [-InputFile <Object>] [-InputObject <Object>] [-OutputFile <Object>] [-Password <Object>] [-ServerInstance <Object>]

[-TrustServerCertificate] [-Username <Object>] [<CommonParameters>]

Invoke-SqlNotebook [-ConnectionString <Object>] [-Force] [-InputFile <Object>] [-InputObject <Object>] [-OutputFile <Object>] [<CommonParameters>]

Invoke-SqlNotebook [-Force] -InputFile <Object> [-OutputFile <Object>] [<CommonParameters>]

Invoke-SqlNotebook [-Force] -InputObject <Object> [-OutputFile <Object>] [<CommonParameters>]

DESCRIPTION

The Invoke-SqlNotebook cmdlet executes a SQL Notebook file (.ipynb) and outputs the materialized notebook.

The Notebook will be executed on the ServerInstance and Database provided.

When the cmdlet is run, the resulting Notebook file will be in the location the user defines or in the same directory of the input notebook file.

The cmdlet outfile may be omitted: if that's the case, it will be created with the input file name and the _out appended to the file.

PARAMETERS

`-AccessToken <PSObject>`

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

This can be used, for example, to connect to `SQL Azure DB` and `SQL Azure Managed Instance` using a `Service Principal` or a `Managed Identity` (see references at the bottom of this page)

In common scenarios, this parameter is obtained with something like `(Get-AzAccessToken -ResourceUrl https://database.windows.net).Token` (requires the Az.Account module)

Do not specify UserName , Password , or Credential when using this parameter.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-ConnectionString <Object>

Specifies a connection string to connect to the server.

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

-Credential <PSCredential>

The PSCredential object whose Username and Password fields will be used to connect to the SQL instance.

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

-Database <Object>

This cmdlet connects to this database in the instance that is specified in the ServerInstance parameter.

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

-Encrypt <String>

The encryption type to use when connecting to SQL Server.

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

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

> This parameter is new in v22 of the module.

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

-Force [<SwitchParameter>]

By default, when the cmdlet writes the materialized notebook to a file, a check is performed to prevent the user from accidentally overwriting an existing file.

Use `-Force` to bypass this check and allow the cmdlet to overwrite the existing file.

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

-HostNameInCertificate <String>

The host name to be used in validating the SQL Server TLS/SSL certificate.

> This parameter is new in v22 of the module.

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

Accept wildcard characters? false

-InputFile <Object>

Specifies a Notebook File (.ipynb) that will be executed through the cmdlet.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-InputObject <Object>

Specifies the Notebook as a Json string that will be used as the input notebook.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-OutputFile <Object>

Specifies the desired output Notebook file for which the executed Notebook will be saved.

Required? false

Position? named

Default value None

Accept pipeline input? False

Accept wildcard characters? false

-Password <Object>

Specifies the password for the SQL Server Authentication login ID that was specified in the Username parameter.

parameter instead.

If you specify the Password parameter followed by your password, the password is visible to anyone who can see your monitor.

If you code Password followed by your password in a .ps1 script, anyone reading the script file will see your password.

Assign the appropriate NTFS permissions to the file to prevent other users from being able to read the file.

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

-ServerInstance <Object>

Specifies a character string or SQL Server Management Objects (SMO) object that specifies the name of an instance of the Database Engine.

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

-TrustServerCertificate [<SwitchParameter>]

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

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

> This parameter is new in v22 of the module.

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

-Username <Object>

Specifies the login ID for making a SQL Server Authentication connection to an instance of the Database Engine.

The password must be specified through the Password parameter.

If Username and Password are not specified, this cmdlet attempts a Windows Authentication connection using the Windows account running the Windows PowerShell session. When possible, use Windows Authentication.

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, WarningAction, WarningVariable, OutBuffer, PipelineVariable, and OutVariable. For more information, see [about_CommonParameters \(https://go.microsoft.com/fwlink/?LinkID=113216\)](https://go.microsoft.com/fwlink/?LinkID=113216).

INPUTS

System.Object

OUTPUTS

System.Object

NOTES

A good way to visualize a materialized Notebook is to use Azure Data Studio.

Example 1: Execute a Notebook against the local server (default instance)

```
PS C:\> Invoke-SqlNotebook -ServerInstance localhost -Database TestNotebook -InputFile C:\notebook.ipynb
```

Directory: C:\

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	8/1/2019 1:00 PM	4656	notebook_out.ipynb

This command runs the notebook against the default instance of SQL Server running on the the machine where the cmdlet is executed from. By default, since no

-OutputFile was passed in, the materialized notebook is saved on disk with the same name as the InputFile with a _out as a suffix to the filename (notebook.ipynb ->

notebook_out.ipynb)

Example 2: Execute Notebook on the local server (default instance) and saves the materialized Notebook to a specified file

```
PS C:\> Invoke-SqlNotebook -ServerInstance localhost -Database TestNotebook -InputFile C:\notebook.ipynb -OutputFile C:\new_notebook.ipynb
```


Directory: C:\

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	8/1/2019 1:00 PM	44656	new_notebook.ipynb

This command runs the notebook against the default instance of SQL Server running on the the machine where the cmdlet is executed from. The executed notebook is then

saved to the file specified with the -OutputFile parameter.

Example 3: Execute Notebook using -ConnectionString parameter

```
PS C:\> Invoke-SqlNotebook -ConnectionString 'Server=Localhost;Database=TestNotebook;Trusted_Connection=True;'
-InputFile C:\notebook.ipynb
```

Directory: C:\

Mode	LastWriteTime	Length	Name
----	-----	-----	----
-a----	8/1/2019 1:00 PM	44656	notebook_out.ipynb

This is the same as Example 1, only that the connection to the server is specified via the -ConnectionString parameter.

-- Example 4: Execute Notebook Against All Registered Servers --

```
dir 'SQLSERVER:\SQLRegistration\Database Engine Server Group' |
WHERE { $_.Mode -ne 'd' } |
foreach {
    $datetime = Get-Date -Format yyyyMMddhhmm;
    Get-SqlInstance -ServerInstance $_.Name |
    foreach {
        Invoke-SqlNotebook -ServerInstance $_.Name -Database master -InputFile '$home\Documents\SQL Server
Management Studio\BPCheck.ipynb' `
        -OutputFile "BPCheck_output_$( $_.NetName )_$( $datetime ).ipynb";
```

```
}  
  
}
```

Use Registered Servers or Central Management Server to run Invoke-SqlNotebook against multiple servers. In this example, the `NetName` property of the SQL Server

instance will be included in the name of the output file; the materialized notebook is going to be time-stamped with year-month-day-hour-minute.

Example 5: Execute Notebook, Open Results in Azure Data Studio

```
Invoke-SqlNotebook -ServerInstance ServerA -Database master -InputFile "$home\Documents\SQL Server Management  
Studio\BPCheck.ipynb" |
```

```
Open-EditorFile
```

Note: The `Open-EditorFile` command is only available in the PowerShell Integrated Console.

Example 6: Execute Notebook, Connect to Azure SQL Databases (or Managed Instance) using a Service Principal

```
Import-Module SQLServer
```

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

```
### Obtain the Access Token interactively.
```

```
### Note: help for Invoke-Sqlcmd has examples on other ways to acquire the token.
```

```
Connect-AzAccount
```

```
$access_token = (Get-AzAccessToken -ResourceUrl https://database.windows.net).Token
```

```
# Now that we have the token, we use it to connect to the database 'mydb' on server 'myserver'
```

```
Invoke-SqlNotebook -ServerInstance myserver.database.windows.net -Database master -AccessToken $access_token`  
-InputFile MyNotebook.ipynb
```

RELATED LINKS

Online Version: <https://learn.microsoft.com/powershell/module/sqlserver/invoke-sqlnotebook>

PowerShell extension

SQLServer_Cmdlets

Service Principal

Managed Identity