PowerShell Get-Help on command 'Complete-Transaction'

PS C:\Users\wahid> Get-Help Complete-Transaction

NAME

Complete-Transaction

SYNOPSIS

Commits the active transaction.

SYNTAX

Complete-Transaction [-Confirm] [-WhatIf] [<CommonParameters>]

DESCRIPTION

The `Complete-Transaction` cmdlet commits an active transaction. When you

commit a transaction, the commands in the transaction are finalized and the

data affected by the commands is changed.

If the transaction includes multiple subscribers, to commit the transaction,

you must enter one `Complete-Transaction` command for every

`Start-Transaction` command.

The `Complete-Transaction` cmdlet is one of a set of cmdlets that support the

transactions feature in Windows PowerShell. For more information, see

about_Transactions (../Microsoft.PowerShell.Core/About/about_Transactions.md).

PARAMETERS

  -Confirm <System.Management.Automation.SwitchParameter>

    Prompts you for confirmation before running the cmdlet.

  -WhatIf <System.Management.Automation.SwitchParameter>

    Shows what would happen if the cmdlet runs. The cmdlet is not run.

  <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).

--------------- Example 1: Commit a transaction ---------------

Set-Location hkcu:\software

Start-Transaction

New-Item MyCompany -UseTransaction

Get-ChildItem m*

Hive: HKEY_CURRENT_USER\software

SKC	VC	Name	Property
82	1	Microsoft	{(default)}

Complete-Transaction

Get-ChildItem m*

Hive: HKEY_CURRENT_USER\Software

```
SKC  VC Name                 Property
---  -- ----                 --------
82    1 Microsoft            {(default)}
0     0 MyCompany            {}
```

This example shows what happens when you use the `Complete-Transaction` cmdlet
to commit a transaction.

The `Start-Transaction` command starts the transaction. The New-Item command
uses the UseTransaction parameter to include the command in the transaction.

The first `Get-ChildItem` command shows that the new item has not yet been
added to the registry.

The `Complete-Transaction` command commits the transaction, which makes the
registry change effective. As a result, the second `Get-ChildItem` command
shows that the registry is changed.

Example 2: Commit a transaction that has more than one subscriber

```
Set-Location hkcu:\software
Start-Transaction
New-Item MyCompany -UseTransaction
```

Hive: HKEY_CURRENT_USER\Software

```
SKC  VC Name                 Property
---  -- ----                 --------
0     0 MyCompany            {}
```

```
Start-Transaction
Get-Transaction
```

```
RollbackPreference   SubscriberCount  Status

------------------   ---------------  ------

Error                2               Active
```

```
New-ItemProperty -Path MyCompany -Name MyKey -Value -UseTransaction
```

```
MyKey

-----

123
```

```
Complete-Transaction
```

```
Get-Transaction
```

```
RollbackPreference   SubscriberCount  Status

------------------   ---------------  ------

Error                1               Active
```

```
Get-ChildItem m*
```

```
Hive: HKEY_CURRENT_USER\Software
```

```
SKC  VC Name                   Property

---  -- ----                   --------

82   1 Microsoft               {(default)}
```

```
Complete-Transaction
```

```
Get-ChildItem m*
```

```
Hive: HKEY_CURRENT_USER\Software
```

```
SKC  VC Name                   Property

---  -- ----                   --------
```

```
82   1 Microsoft              {(default)}
0    1 MyCompany             {MyKey}
```

This example shows how to use `Complete-Transaction` to commit a transaction that has more than one subscriber.

To commit a multi-subscriber transaction, you must enter one `Complete-Transaction` command for every `Start-Transaction` command. The data is changed only when the final `Complete-Transaction` command is submitted.

For demonstration purposes, this example shows a series of commands entered at the command line. In practice, transactions are likely to be run in scripts, with the secondary transaction being run by a function or helper script that is called by the main script.

In this example, a `Start-Transaction` command starts the transaction. A `New-Item` command with the UseTransaction parameter adds the MyCompany key to the Software key. Although the `New-Item` cmdlet returns a key object, the data in the registry is not yet changed.

A second `Start-Transaction` command adds a second subscriber to the existing transaction. The `Get-Transaction` cmdlet confirms that the subscriber count is 2. A New-ItemProperty command with the UseTransaction parameter adds a registry entry to the new MyCompany key. Again, the command returns a value, but the registry is not changed.

The first `Complete-Transaction` command reduces the subscriber count by 1. This is confirmed by a `Get-Transaction` command. However, no data is changed, as evidenced by a `Get-ChildItem m*` command.

The second `Complete-Transaction` command commits the entire transaction and changes the data in the registry. This is confirmed by a second `Get-ChildItem m*` command, which shows the changes.

Example 3: Perform a transaction that does not change any data

Set-Location hkcu:\software

Start-Transaction

New-Item MyCompany -UseTransaction

Get-ChildItem m*

Hive: HKEY_CURRENT_USER\Software

SKC VC Name   Property

--- -- ----   --------

82   1 Microsoft   {(default)}

Get-ChildItem m* -UseTransaction

Hive: HKEY_CURRENT_USER\Software

SKC VC Name   Property

--- -- ----   --------

82   1 Microsoft   {(default)}

0   0 MyCompany   {}

Complete-Transaction

This example shows the value of using Get-* commands, and other commands that

do not change data, in a transaction. When a `Get-*` command is used in a

transaction, it gets the objects that are part of the transaction. This allows

you to preview the changes in the transaction before the changes are committed.

In this example, a transaction is started. A New-Item command with the

UseTransaction parameter adds a new key to the registry as part of the

transaction.

Because the new registry key is not added to the registry until the

`Complete-Transaction` command is run, a simple `Get-ChildItem` command shows

the registry without the new key.

However, when you add the UseTransaction parameter to the `Get-ChildItem`

command, the command becomes part of the transaction, and it gets the items in

the transaction even if they are not yet added to the data.

REMARKS

To see the examples, type: "get-help Complete-Transaction -examples".

For more information, type: "get-help Complete-Transaction -detailed".

For technical information, type: "get-help Complete-Transaction -full".

For online help, type: "get-help Complete-Transaction -online"