PowerShell Get-Help on command 'Get-Transaction'

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

NAME

Get-Transaction

SYNOPSIS

Gets the current (active) transaction.

SYNTAX

Get-Transaction [<CommonParameters>]

DESCRIPTION

The `Get-Transaction` cmdlet gets an object that represents the current transaction in the session.

This cmdlet never returns more than one object, because only one transaction is active at a time. If you start one or more independent transactions (by using the Independent parameter of Start-Transaction), the most recently started transaction is active, and that is the transaction that `Get-Transaction` returns.

When all active transactions have either been rolled back or committed, this cmdlet shows the transaction that was most recently active in the session.

This 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

<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: Get the current transaction ------------

```
Start-Transaction
Get-Transaction

RollbackPreference   SubscriberCount   Status
------------------   ---------------   ------
Error                1                 Active
```

This command uses the `Get-Transaction` cmdlet to get the current transaction.

Example 2: Show the properties and methods of the transaction object

```
Get-Transaction | Get-Member

Name               MemberType Definition
----               ---------- ----------
Dispose            Method     System.Void Dispose(), System.Void
Dispose(Boolean disposing)
```

```
Equals             Method     System.Boolean Equals(Object obj)
GetHashCode        Method     System.Int32 GetHashCode()
GetType            Method     System.Type GetType()
ToString           Method     System.String ToString()
IsCommitted        Property   System.Boolean IsCommitted {get;}
IsRolledBack       Property   System.Boolean IsRolledBack {get;}
RollbackPreference Property   System.Management.Automation.RollbackSeverity
RollbackPreference {get;}
SubscriberCount    Property   System.Int32 SubscriberCount {get;set;}
```

This command uses the `Get-Member` cmdlet to show the properties and methods of the transaction object.

Example 3: Show the property values of a rolled back transaction

```
Set-Location hklm:\software
Start-Transaction
New-Item MyCompany -UseTransaction
Undo-Transaction
Get-Transaction

RollbackPreference   SubscriberCount   Status
------------------   ---------------   ----------
Error                0                 RolledBack
```

This command shows the property values of a transaction object for a transaction that has been rolled back.

Example 4: Show the property values of a committed transaction

```
Set-Location hklm:\software
Start-Transaction
New-Item MyCompany -UseTransaction
Complete-Transaction
Get-Transaction
```

```
RollbackPreference   SubscriberCount   Status
------------------   ---------------   ---------
Error                1                 Committed
```

This command shows the property values of a transaction object for a transaction that has been committed.

- Example 5: Start a transaction while another is in progress -

```
Set-Location hklm:\software
Start-Transaction
New-Item MyCompany -UseTransaction
Start-Transaction
New-Item MyCompany2 -UseTransaction
Get-Transaction

RollbackPreference   SubscriberCount   Status
------------------   ---------------   ------
Error                2                 Active

Complete-Transaction
Get-Transaction

RollbackPreference   SubscriberCount   Status
------------------   ---------------   ------
Error                1                 Active

Complete-Transaction
Get-Transaction

RollbackPreference   SubscriberCount   Status
------------------   ---------------   ---------
Error                1                 Committed
```

This example shows the effect on the transaction object of starting a transaction while another transaction is in progress. Typically, this happens when a script that runs a transaction includes a function or calls a script that contains another complete transaction.

Unless the second `Start-Transaction` command includes the Independent parameter, `Start-Transaction` does not create a new transaction. Instead, it adds a second subscriber to the original transaction.

The first `Start-Transaction` command starts the transaction. A New-Item command with the UseTransaction parameter is part of the transaction.

A second `Start-Transaction` command adds a subscriber to the transaction. The next `New-Item` command is also part of the transaction.

The first `Get-Transaction` command shows the multi-subscriber transaction. Notice that the subscriber count is 2.

The first Complete-Transaction command does not commit the transaction, but it reduces the subscriber count to 1.

The second `Complete-Transaction` command commits the transaction.

Example 6: Start an independent transaction while another is in progress

```
Set-Location hklm:\software
Start-Transaction
Get-Transaction
```

```
RollbackPreference   SubscriberCount   IsRolledBack   IsCommitted
------------------   ---------------   ------------   -----------
Error                1                 False          False
```

```
Start-Transaction -Independent
Get-Transaction

RollbackPreference SubscriberCount IsRolledBack IsCommitted
------------------ --------------- ------------ -----------
Error              1               False        False

Complete-Transaction
Get-Transaction
Complete-Transaction
Get-Transaction
```

This example shows the effect on the transaction object of starting an independent transaction while another transaction is in progress.

The first `Start-Transaction` command starts the transaction. A `New-Item` command with the UseTransaction parameter is part of the transaction.

A second `Start-Transaction` command adds a subscriber to the transaction. The next `New-Item` command is also part of the transaction.

The first `Get-Transaction` command shows the multi-subscriber transaction. Note that the subscriber count is 2.

The `Complete-Transaction` command reduces the subscriber count to 1, but it does not commit the transaction.

The second `Complete-Transaction` command commits the transaction.

REMARKS

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

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

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

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