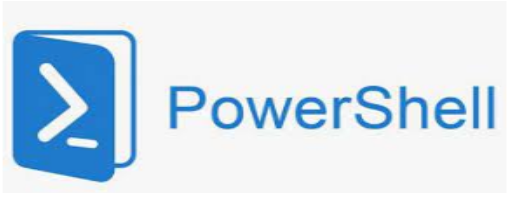

Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Start-Transaction'

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

NAME

Start-Transaction

SYNOPSIS

Starts a transaction.

SYNTAX

Start-Transaction [-Independent] [-RollbackPreference {Error |

TerminatingError | Never}] [-Timeout <System.Int32>] [-Confirm] [-WhatIf]

[<CommonParameters>]

DESCRIPTION

The `Start-Transaction` cmdlet starts a transaction, which is a series of commands that are managed as a unit. A transaction can be completed, or committed. Alternatively, it can be completely undone, or rolled back, so that any data changed by the transaction is restored to its original state. Because the commands in a transaction are managed as a unit, either all commands are committed or all commands are rolled back.

By default, if any command in the transaction generates an error, transactions are rolled back automatically. You can use the RollbackPreference parameter to change this behavior.

The cmdlets used in a transaction must be designed to support transactions. Cmdlets that support transactions have a UseTransaction parameter. To perform transactions in a provider, the provider must support transactions. The Windows PowerShell Registry provider in Windows Vista and later versions of the Windows operating system supports transactions. You can also use the Microsoft.PowerShell.Commands.Management.TransactedString class to include expressions in transactions on any version of the Windows system that supports Windows PowerShell. Other Windows PowerShell providers can also support transactions.

Only one transaction can be active at a time. If you start a new, independent transaction while a transaction is in progress, the new transaction becomes the active transaction, and you must commit or roll back the new transaction before you make any changes to the original transaction.

`Start-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

-Independent <System.Management.Automation.SwitchParameter>

Indicates that this cmdlet starts a transaction that is independent of any transactions in progress. By default, if you use `Start-Transaction` while another transaction is in progress, a new subscriber is added to the transaction in progress. This parameter has an effect only when a transaction is already in progress in the session.

By default, if you use `Start-Transaction` while a transaction is in

progress, the existing transaction object is reused and the subscriber count is incremented. The effect is much like joining the original transaction. An `Undo-Transaction` command rolls back the whole the transaction. To complete the transaction, you must enter a `Complete-Transaction` command for each subscriber. Because most transactions that are in progress at the same time are related, the default is sufficient for most uses.

If you specify the Independent parameter, this cmdlet creates a new transaction that can be completed or undone without affecting the original transaction. However, because only one transaction can be active at a time, you must complete or roll back the new transaction before resuming work on the original transaction.

-RollbackPreference <System.Management.Automation.RollbackSeverity>
Specifies the conditions under which a transaction is automatically rolled back. The acceptable values for this parameter are:

- `Error` The transaction is rolled back automatically if a terminating or non-terminating error occurs. - `TerminatingError` The transaction is rolled back automatically if a terminating error occurs. - ``Never`` The transaction is never rolled back automatically.

The default value is `Error`.

-Timeout <System.Int32>
Specifies the maximum time, in minutes, that the transaction is active. When the time-out expires, the transaction is automatically rolled back.

By default, there is no time-out for transactions that are started at the command line. When transactions are started by a script, the default time-out is 30 minutes.

-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: Start and roll back a transaction ---------

```
Set-Location hkcu:\software
Start-Transaction
New-Item "ContosoCompany" -UseTransaction
New-ItemProperty "ContosoCompany" -Name "MyKey" -Value 123 -UseTransaction
Undo-Transaction
```

These commands start and then roll back a transaction. Because the transaction

is rolled back, no changes are made to the registry.

--------- Example 2: Start and complete a transaction ---------

```
Set-Location hkcu:\software
Start-Transaction
New-Item "ContosoCompany" -UseTransaction
New-ItemProperty "ContosoCompany" -Name "MyKey" -Value 123 -UseTransaction
Complete-Transaction
```

These commands start and then complete a transaction. No changes are made to

the registry until the `Complete-Transaction` command is used.

-------- Example 3: Use different rollback preferences --------

```
Set-Location HKCU:\software
Start-Transaction
New-Item -Path "NoPath" -Name "ContosoCompany" -UseTransaction
New-Item -Path . -Name "ContosoCompany" -UseTransaction
Start-Transaction -RollbackPreference never
New-Item -Path "NoPath" -Name "ContosoCompany" -UseTransaction
New-Item -Path . -Name "ContosoCompany" -UseTransaction

# Start-Transaction (-rollbackpreference error)

Start-Transaction
New-Item -Path "NoPath" -Name "ContosoCompany" -UseTransaction
New-Item : The registry key at the specified path does not exist.

At line:1 char:9
+ new-item <<<<  -Path NoPath -Name ContosoCompany -UseTransaction

New-Item -Path . -Name "Contoso" -UseTransaction

New-Item : Cannot use transaction. The transaction has been rolled back or has
timed out.
At line:1 char:9
+ New-Item <<<<  -Path . -Name ContosoCompany -UseTransaction

# Start-Transaction (-rollbackpreference never)

Start-Transaction -RollbackPreference never
New-Item -Path "NoPath" -Name "ContosoCompany" -UseTransaction

New-Item : The registry key at the specified path does not exist.
At line:1 char:9
+ New-Item <<<<  -Path NoPath -Name "ContosoCompany" -UseTransaction
```

```
New-Item -Path . -Name "ContosoCompany" -UseTransaction

Hive: HKEY_CURRENT_USER\Software
SKC  VC Name                           Property
---  -- ----                           --------
0    0 ContosoCompany                  {}
Complete-Transaction

# Succeeds
```

This example demonstrates the effect of changing the RollbackPreference parameter value.

In the first set of commands, `Start-Transaction` does not use RollbackPreference . As a result, the default value (Error) is used. When an error occurs in a transaction command, that is, the specified path does not exist, the transaction is automatically rolled back.

In the second set of commands, `Start-Transaction` uses RollbackPreference with a value of Never. As a result, when an error occurs in a transaction command, the transaction is still active and can be completed successfully.

Because most transactions must be performed without error, the default value of RollbackPreference is typically preferred.

Example 4: Use this cmdlet while a transaction is in progress

```
Set-Location HKCU:\software
Start-Transaction
New-Item "ContosoCompany" -UseTransaction
Start-Transaction
Get-Transaction
New-Item "ContosoCompany2" -UseTransaction
```

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

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

This example shows the effect of using `Start-Transaction` while a transaction is in progress. The effect is much like joining the transaction in progress.

Although this is a simplified command, this scenario frequently occurs when the transaction involves running a script that includes a complete transaction.

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

The second `Start-Transaction` command adds a new subscriber to the transaction. The `Get-Transaction` command now returns a transaction with a subscriber count of 2. The second `New-Item` command is part of the same transaction.

No changes are made to the registry until the whole transaction is completed. To complete the transaction, you must enter two `Complete-Transaction` commands, one for each subscriber. If you were to roll back the transaction at any point, all the transaction would be rolled back for both subscribers.
Example 5: Start an independent transaction while one is in progress

```
Set-Location HKCU:\software
Start-Transaction
New-Item "ContosoCompany" -UseTransaction
Start-Transaction -Independent
Get-Transaction
```

```
Undo-Transaction
New-ItemProperty -Path "ContosoCompany" -Name "MyKey" -Value 123
-UseTransaction
Complete-Transaction
Get-ChildItem contoso*
Get-Transaction

RollbackPreference   SubscriberCount   Status
------------------   ---------------   ------
Error                1                 Active
Undo-Transaction
New-ItemProperty -Path "ContosoCompany" -Name "MyKey" -Value 123
-UseTransaction
MyKey
-----
123
Complete-Transaction
Get-ChildItem contoso*
Hive: HKEY_CURRENT_USER\Software
SKC  VC Name                          Property
---  -- ----                         --------
0    1 MyCompany                     {MyKey}
```

This example shows the effect of using the Independent parameter of `Start-Transaction` to start a transaction while another transaction is in progress. In this case, the new transaction is rolled back without affecting the original transaction.

Although the transactions are logically independent, because only one transaction can be active at a time, you must roll back or commit the newest transaction before resuming work on the original transaction.

The first set of commands starts a transaction. The `New-Item` command is part

of the first transaction.

In the second set of commands, the `Start-Transaction` command uses the Independent parameter. The `Get-Transaction` command that follows shows the transaction object for the active transaction, which is the newest one. The subscriber count is equal to 1, that shows that the transactions are unrelated.

When the active transaction is rolled back by using an `Undo-Transaction` command, the original transaction becomes active again.

The `New-ItemProperty` command, which is part of the original transaction, finishes without error, and the original transaction can be completed by using the `Complete-Transaction` command. As a result, the registry is changed.
-- Example 6: Run commands that are not part of a transaction --

```
Set-Location hkcu:\software
Start-Transaction
New-Item "ContosoCompany1" -UseTransaction
New-Item "ContosoCompany2"
New-Item "ContosoCompany3" -UseTransaction
Get-ChildItem contoso*

Hive: HKEY_CURRENT_USER\Software
SKC  VC Name                           Property
---  -- ----                           --------
0   0 ContosoCompany2                  {}

Complete-Transaction
Get-ChildItem contoso*

Hive: HKEY_CURRENT_USER\Software
SKC  VC Name                           Property
---  -- ----                           --------
```

```
0   0 ContosoCompany1               {}
0   0 ContosoCompany2               {}
0   0 ContosoCompany3               {}
```

This example demonstrates that commands that are submitted while a transaction is in progress can be included in the transaction or not included. Only commands that use the UseTransaction parameter are part of the transaction.

The first and third `New-Item` commands use the UseTransaction parameter. These commands are part of the transaction. Because the second `New-Item` command does not use the UseTransaction parameter, it is not part of the transaction.

The first Get-ChildItem command shows the effect. The second `New-Item` command is completed immediately, but the first and third `New-Item` commands are not effective until the transaction is committed.

The `Complete-Transaction` command commits the transaction. As a result, the second Get-ChildItem command shows that all of the new items are added to the registry.

Example 7: Roll back a transaction that does not finish in a specified time

```
Start-Transaction -Timeout 2

# Wait two minutes...

Get-Transaction
New-Item HKCU:\Software\ContosoCompany -UseTransaction
Start-Transaction -Timeout 2

# Wait two minutes...

Get-Transaction
```

```
RollbackPreference   SubscriberCount   Status
------------------   ---------------   -----------
Error                1                 RolledBack

New-Item HKCU:\Software\ContosoCompany -UseTransaction

New-Item : Cannot use transaction. The transaction has been rolled back or has
timed out.
At line:1 char:9
+ new-item <<<<  MyCompany -UseTransaction
```

This command uses the Timeout parameter of `Start-Transaction` to start a transaction that must be completed within two minutes. If the transaction is not finished when the time-out expires, it is rolled back automatically.

When the time-out expires, you are not notified, but the Status property of the transaction object is set to RolledBack and commands that use the UseTransaction parameter fail.

REMARKS

To see the examples, type: "get-help Start-Transaction -examples".
For more information, type: "get-help Start-Transaction -detailed".
For technical information, type: "get-help Start-Transaction -full".
For online help, type: "get-help Start-Transaction -online"