

Using the PowerShell cmdlets

The installer automatically sets up PowerShell to use the SQL Clone PowerShell module (located at: C:\Program Files (x86)\Red Gate\SQL Clone Preview\Redgate.SqlClone.PowerShell).

Complete PowerShell cmdlet documentation is found at [Cmdlet reference](#).

Use the cmdlets like this. Finish all invocations with **-Verbose** to see debug output.

Set connection string for use by SQL Clone. You should only do this the first time.

```
Add-InstantCloneConnection -Instance ".\sql2012" -Verbose # For Windows authentication
```

```
Add-InstantCloneConnection -Instance ".\sql2012" -Username "johnsmith" -Password "mypassword" -Verbose # For SQL Server authentication
```

Save a snapshot of my WidgetProd database to disk

```
Save-InstantCloneSnapshot -DatabaseName WidgetProd -SnapshotName WidgetSnp1 -Verbose
```

Create a couple of clones from the WidgetSnp1 snapshot

```
New-InstantCloneClone -NewDatabaseName WidgetClone1 -SnapshotName WidgetSnp1 -Verbose
```

```
New-InstantCloneClone -NewDatabaseName WidgetClone2 -SnapshotName WidgetSnp1 -Verbose
```

I'm bored with these clones now. Let's drop them

```
Remove-InstantCloneClone -CloneName WidgetClone1 -Verbose
```

```
Remove-InstantCloneClone -CloneName WidgetClone2 -Verbose
```

Let's drop that snapshot too

```
Remove-InstantCloneSnapshot -SnapshotName WidgetSnp1 -Verbose
```

When you restart your machine, clone databases will be offline until you run the **restore** cmdlet:

```
Restore-InstantClone -Verbose
```

See all the clones and snapshots with the **Show-InstantCloneSnapshots/Clones** cmdlets

```
Show-InstantCloneSnapshots -Verbose
```

```
Show-InstantCloneClones -Verbose
```

See all cmdlets available with the **Get-Command** cmdlet:

```
Get-Command -Module RedGate.InstantClone.PowerShell
```

Setting storage locations

Snapshots and clones are stored by default in C:\Users\YourUser\Documents\Red Gate Instant Clone. If you want to change this, use the **Save-InstantCloneOptions** cmdlet:

```
Save-InstantCloneOptions -SnapshotFolder D:\IC\Snapshots -Verbose
```

```
Save-InstantCloneOptions -CloneFolder D:\IC\Clones -Verbose
```

Working as a team with the PowerShell cmdlets

You may want to create a snapshot on one machine and use it on another. For instance, your colleagues might want to use your snapshots, or you might want to create a snapshot on one SQL Server and clone it on another.

Snapshots are, by default, local. Their metadata is stored locally, and they are only cloneable on your machine. You can create a **Shared Snapshot** which is stored in your **Shared Snapshot Folder**. This snapshot can be used on any machine.

To use shared snapshots, each computer should have its *SharedSnapshotsFolder* set to a common location e.g. \\john-pc\sharedsnapshots. When you take a snapshot, use the *-PutInSharedFolder* parameter. The snapshots will then be created in \\john-pc\sharedsnapshots. Then, on any machine, simply create a clone as usual.

If the shared snapshot folder becomes unavailable (e.g. that machine is powered-off) then the clone will stop working. Once the shared-snapshot folder is restored, a restart of the virtual disk service on the client machine is required.

Creating a shared snapshot using the PowerShell cmdlets

In this example we have an Adventureworks2012 database on the machine PRODSERVER that we want to clone on machine SARAH-PC. We use a SMB share on the machine JOHN-PC to share the snapshot.

On PRODSERVER we run:

```
Save-InstantCloneOptions -SharedSnapshotFolder "\\JOHN-PC\SharedSnapshots" -Verbose
```

```
Save-InstantCloneSnapshot -DatabaseName AdventureWorks2012 -SnapshotName aworks_shared -Verbose -PutInSharedFolder
```

On SARAH-PC we run:

`Save-InstantCloneOptions -SharedSnapshotFolder "\\JOHN-PC\SharedSnapshots" -Verbose`

`New-InstantCloneClone -NewDatabaseName awc1_shared -SnapshotName aworks_shared -Verbose`

Sharing an existing snapshot using the PowerShell cmdlets

You may want to create a snapshot on one machine, but clone it on another. To do this:

- On the source machine Setup your shared folder with the **Save-InstantCloneOptions** cmdlet:

`Save-InstantCloneOptions -SharedSnapshotFolder [\\myServer\SharedSnapshots\myServer\SharedSnapshots] -Verbose`

- Create a snapshot as usual with the **Save-InstantCloneSnapshot** cmdlet:

`Save-InstantCloneSnapshot -DatabaseName AdventureWorks -SnapshotName AdvSnap -Verbose`

- Run the **Save-InstantCloneSharedSnapshot** cmdlet. This will copy your Snapshot into the shared-directory so it can be accessed by others.

`Save-InstantCloneSharedSnapshot -SnapshotName AdvSnap -NewSnapshotName AdvSnapShared`

To delete a shared-snapshot, delete the relevant GUID-named directory.

On the target machine:

- Setup your shared folder with the **Save-InstantCloneOptions** cmdlet:

`Save-InstantCloneOptions -SharedSnapshotFolder [\\myServer\SharedSnapshots\myServer\SharedSnapshots] -Verbose`

- (Optional) Run **Show-InstantCloneSnapshot** cmdlet. *AdvSnapShared* is listed.

`Show-InstantCloneSnapshots -Verbose`

- Run the **New-InstantCloneClone** cmdlet to create a new clone from the shared-snapshot.

`New-InstantCloneClone -SnapshotName AdvSnapShared -NewDatabaseName AdvClone -Verbose`