

This forum has migrated to [Microsoft Q&A](#). Visit [Microsoft Q&A](#) to post new questions.

[Learn More](#)

United States (English) ▼

[Sign in](#)[Home](#)[Windows Server 2012](#)[Windows Server 2008 R2](#)[Windows Server 2003](#)[Library](#)[Fo](#)[Ask a question](#)[Search related threads](#)[Quick access](#)

Answered by:



3,846

Points  
Top 5%[cmille19](#)Joined Aug  
20085 [cmille19's threads](#)[Show activity](#)

14

# Running SQL Query with Powershell on remote server

[Archived Forums](#) > [Windows PowerShell](#)

Question

0

[Sign in  
to vote](#)

Hi all,

I want to run a SQL query with Powershell on a remote server (named VCOM) in our domain. The server has Win2008 R2 SP1 and MS SQL Server 2008 R2.

My aim is to count the columns of the table "dbo.Comm\_Configstore" in the database "VCOM\_Comm" and get the number in the Powershell window.

The query is called :

```
SELECT COUNT(*) FROM [VCOM_Comm].[dbo].  
[COMM_ConfigStore]
```

This query actually works , if i run it directly in the SQL Server Management Studio.

So I googled a lot and found a script, which might be useful and adjusted it for my demands :

```
$cn = New-Object System.Data.SqlClient.SqlConnection  
$cn.ConnectionString = "Server= VCOM  
;Database=VCOM_Comm ;User ID= User Name  
;Password= Password of user ;Trusted_Connection=True"  
$cn.Open()  
$sql = "SELECT COUNT(*) FROM [VCOM_Comm].[dbo].  
[COMM_ConfigStore]"  
$cmd = new-object System.Data.SqlClient.SqlCommand  
$cmd.Connection = $cn  
$cmd.CommandText = $sql  
$cmd.CommandTimeout = 600  
$cmd.ExecuteNonQuery()  
$cn.Close()
```

The result on the display : -1 (

I can connect to the server via Remote Desktop Session and run the sql database. The right number of columns must be 148. The problem is that i dont get the number shown on the Powershell screen. Can someone tell me the mistake in my script?

Thanks for answering.

Thursday, September 29, 2011 8:49 AM



ATAlex

155 Points

---

## Answers

The problem is you are using an ExecuteNonQuery which by definition does not return a result. It is used for non-queries (insert, updated, delete) instead of select.

Here's some code to return a result set:

6

[Sign in to vote](#)

```
$ServerInstance = "VCOM "
$Database = "VCOM "
$ConnectionTimeout = 30
$Query = "SELECT COUNT(*) FROM [VCOM_Comm].[dbo].[
$QueryTimeout = 120

$conn=new-object System.Data.SqlClient.SqlConnection
$ConnectionString = "Server={0};Database={1};Integrated !
$conn.ConnectionString=$ConnectionString
$conn.Open()
$cmd=new-object system.Data.SqlClient.SqlCommand($Q
$cmd.CommandTimeout=$QueryTimeout
$ds=New-Object system.Data.DataSet
$da=New-Object system.Data.SqlClient.SqlDataAdapter($c
[void]$da.fill($ds)
$conn.Close()
$ds.Tables
```

Marked as answer by [ATAlex](#)

Thursday, September 29, 2011 12:18 PM

Thursday, September 29, 2011 11:49 AM



[cmille19](#)

3,846 Points

---

All replies

The problem is you are using an ExecuteNonQuery which by definition does not return a result. It is used for non-queries (insert, updated, delete) instead of select.

Here's some code to return a result set:

6

Sign in  
to vote

```
$ServerInstance = "VCOM "
$Database = "VCOM "
$ConnectionTimeout = 30
$Query = "SELECT COUNT(*) FROM [VCOM_Comm].[dbo].[
$QueryTimeout = 120

$conn=new-object System.Data.SqlClient.SqlConnection
$ConnectionString = "Server={0};Database={1};Integrated !
$conn.ConnectionString=$ConnectionString
$conn.Open()
$cmd=new-object system.Data.SqlClient.SqlCommand($Q
$cmd.CommandTimeout=$QueryTimeout
$ds=New-Object system.Data.DataSet
$da=New-Object system.Data.SqlClient.SqlDataAdapter($c
[void]$da.fill($ds)
$conn.Close()
$ds.Tables
```

Marked as answer by [ATAlex](#)

Thursday, September 29, 2011 12:18 PM

Thursday, September 29, 2011 11:49 AM



[cmille19](#) 3,846 Points

Thanks, your answer really helped me.

There is so much to learn.....

0

Sign in  
to vote

Thursday, September 29, 2011 12:18 PM



[ATAlex](#) 155 Points

0

[Sign in  
to vote](#)

hi cmille19,

your script worked perfect, can you tell me how to use this script if i have to run a query against multiple SQL instnaces (may be by providing the list of instnaces through a csv file/txt file)

Thanks

Kvelpuri

Wednesday, June 12, 2013 2:25 PM



Kvelpuri

195 Points

0

[Sign in  
to vote](#)

Kvelpuri,

You may use it like this:

where, \$dbserverpath is the path of your text file with sql servers, also I have added to neglect system databases

```
$dbservers = (Get-Content $dbserverspath -Force) |
ForEach-Object{
$server = $_
```

```
[System.Reflection.Assembly]::LoadWithPartialName('Microsoft.SqlServer.SMO') | out-null
```

```
$s = New-Object
```

```
('Microsoft.SqlServer.Management.Smo.Server') $server
```

```
$dbs = $s.Databases
```

```
$dbs | SELECT Name -ExpandProperty Name | ForEach-Object{
```

```
[String]$database = $_
```

```
if (($database -ne "master") -and ($database -ne "tempdb") -and ($database -ne "msdb") -and ($database -ne "model")) {
```

```
$ServerName = $server
```

```
$DatabaseName = $database
```

```
$Query = "SELECT COUNT(1) FROM cs_users"
```

```
#Timeout parameters
```

```
$QueryTimeout = 120
```

```
$ConnectionTimeout = 30
```

#Action of connecting to the Database and executing the query and returning results if there were any.

```
$conn = New-Object
```

```
System.Data.SqlClient.SqlConnection
```

```
$ConnectionString = "Server={0};Database={1};Integrated Security=True;Connect Timeout={2}" -f $ServerName,
```

```
$DatabaseName, $ConnectionTimeout
```

```
$conn.ConnectionString = $ConnectionString
```

```
$conn.Open()
```

```
$cmd = New-Object
```

```
system.Data.SqlClient.SqlCommand($Query, $conn)
```

```
$cmd.CommandTimeout = $QueryTimeout
```

```
$ds = New-Object system.Data.DataSet
```

```
$da = New-Object
```

```
system.Data.SqlClient.SqlDataAdapter($cmd)  
[void]$da.fill($ds)  
$conn.Close()  
$ds.Tables  
}  
}  
}
```

Edited by [Rajat Bhardwaj](#)

Wednesday, November 5, 2014 5:35 PM

Proposed as answer by [Kvelpuri](#)

Thursday, January 21, 2016 10:19 PM

Wednesday, November 5, 2014 5:33 PM



[Rajat Bhardwaj](#)

0 Points

0

[Sign in  
to vote](#)

The problem is you are using an ExecuteNonQuery which by definition does not return a result. It is used for non-queries (insert, updated, delete) instead of select.

Here's some code to return a result set:

```
$ServerInstance = "VCOM "  
$Database = "VCOM "  
$ConnectionTimeout = 30  
$Query = "SELECT COUNT(*) FROM [VCOM_Comm].[dbc  
$QueryTimeout = 120  
  
$conn=new-object System.Data.SqlClient.SqlConnection  
$ConnectionString = "Server={0};Database={1};Integrate  
$conn.ConnectionString=$ConnectionString  
$conn.Open()  
$cmd=new-object system.Data.SqlClient.SqlCommand($  
$cmd.CommandTimeout=$QueryTimeout  
$ds=New-Object system.Data.DataSet  
$da=New-Object system.Data.SqlClient.SqlDataAdapter(  
[void]$da.fill($ds)  
$conn.Close()  
$ds.Tables
```

Hi! I need to query a database that is on another organization, but it's IP connected.

So I can't use integrated security. How can I pass user and password on string?

Thankyou in advance.

FM

Monday, August 7, 2017 7:08 PM



FabioMartinsRB Rede Bahia 5 Points

Please post questions about how to use SQLServer in the SQLServer forum. They will work with you to discover the nature of your issue.

0

[Sign in to vote](#)

You can also contact the DBA for the foreign corp to find out how to remotely access the database. I suspect it is not available remotely.

\\_(ツ)\_/

Monday, August 7, 2017 9:19 PM



jrv Consultant 239,063 Points

I need to query using powershell.  
\*that\* DBA doesn't know powershell.

0

[Sign in to vote](#)

Tuesday, August 8, 2017 11:20 AM



FabioMartinsRB Rede Bahia 5 Points

The DBA can tell you how to connect. Connecting is the same in all languages and all tools. You need to know what the connection string is. You need to know if the security and firewall; allow the connection. Only the remote DBA can tell you those things.

0

[Sign in to vote](#)

You cannot use a username and password with "integrated security". That only works in a domain.

Look up how to use SQLServer connection strings. Get a username, password and an instance name from the remote DBA.

\\_(ツ)\_/

Tuesday, August 8, 2017 11:24 AM



jrv Consultant 239,063 Points



Here is a connect string for SQLServer that uses a username and password:

**"User ID=jsmith;PWD=mypasswordInitial  
Catalog=master;Data Source=OMEGA\SQLEXPRESS"**

1

[Sign in  
to vote](#)

\\_(ツ)\_/

Tuesday, August 8, 2017 11:28 AM



[jrv](#) Consultant 239,063 Points

Hi Fabio,  
you can use dbatools for that:

**Install-Module** dbatools

0

[Sign in  
to vote](#)

Then it is a simple matter of:

```
$Query = "SELECT foo FROM bar"  
$database = "exampledb"  
$wincred = Get-Credential ad\sqladmin  
$server = Connect-DbSqlServer -SqlInstance sql2014 -  
$server.Query($Query, $database, $true)
```

Cheers,  
Fred

PS: Whether you use windows credentials or sql  
credentials doesn't really matter, just use what you have  
available

There's no place like 127.0.0.1

Edited by [FWN](#) Tuesday, August 8, 2017 11:30 AM

Tuesday, August 8, 2017 11:29 AM



[FWN](#) die netzwerker Computern... 30,482 Points

0

[Sign in  
to vote](#)

The DBA can tell you how to connect. Connecting is the same in all languages and all tools. You need to know what the connection string is. You need to know if the security and firewall; allow the connection. Only the remote DBA can tell you those things.

You cannot use a username and password with "integrated security". That only works in a domain.

Look up how to use SQLServer connection strings. Get a username, password and an instance name from the remote DBA.

---

\('ㄿ')\_/

I already know the username, password, instance, etc... firewall is opened... Thanks.

Tuesday, August 8, 2017 11:31 AM



FabioMartinsRB Rede Bahia 5 Points

Here is a connect string for SQLServer that uses a username and password:

**"User ID=jsmith;PWD=mypasswordInitial  
Catalog=master;Data  
Source=OMEGA\SQLEXPRESS"**

0

[Sign in  
to vote](#)

---

\('ㄿ')\_/

Great! I will try that.

Tuesday, August 8, 2017 11:32 AM



FabioMartinsRB Rede Bahia 5 Points

0

[Sign in  
to vote](#)

Hi Fabio,  
you can use dbatools for that:

```
Install-Module dbatools
```

Then it is a simple matter of:

```
$Query = "SELECT foo FROM bar"  
$database = "exampledb"  
$wincred = Get-Credential ad\sqladmin  
$server = Connect-DbaSqlServer -SqlInstance sql2008  
$server.Query($Query, $database, $true)
```

Cheers,  
Fred

There's no place like 127.0.0.1

In my case, I will use that on a script on each 5 minutes, I  
cant get get-gredential.

Anyway, I will try that too... Thanks

Tuesday, August 8, 2017 11:33 AM



**FabioMartinsRB** Rede Bahia 5 Points

0

[Sign in  
to vote](#)

Heya,  
was just an example on how to get that credential object.  
There are many ways to automate creating a credential  
object, for example [Encrypt-Credential](#) (handy when you  
need to empower a task running under another account  
or as system).

Cheers,  
Fred

There's no place like 127.0.0.1

Tuesday, August 8, 2017 12:10 PM



**FWN** die netzwerker Computern... 30,482 Points

The credentials for dbatools does not perform a "SQLServer Authentication" logon. It merely allows alternate "Windows" credentials to be used.

---

0

[Sign in  
to vote](#)

\(\ツ)/

Tuesday, August 8, 2017 12:15 PM



jrv Consultant 239,063 Points

I would also note that any DBA that allows SQLServer authentication from external (non-domain) clients is taking a chance as the passwords are sent unencrypted unless you are requiring an encrypted connection.

---

0

[Sign in  
to vote](#)

\(\ツ)/

Tuesday, August 8, 2017 12:18 PM



jrv Consultant 239,063 Points

0

[Sign in  
to vote](#)

Here is the connection string for an encrypted connection.

**"User ID=jsmith;Initial Catalog=master;Data Source=OMEGA\SQLEXPRESS;Use Encryption for Data=True"**

Create a file on your desktop called "test.udl". Double click the file and it will open the connection string wizard. You will be able to see and set all options for all installed providers and copy the string.

The DBATools module is nice for many things but completely unnecessary for simple queries.

\(ツ)/

Tuesday, August 8, 2017 12:23 PM



jrv Consultant 239,063 Points

0

[Sign in  
to vote](#)

The credentials for dbatools does not perform a "SQLServer Authentication" logon. It merely allows alternate "Windows" credentials to be used.

\(ツ)/

Actually ... you are wrong about that (or I darned well hope you are, or Chrissy is going to go ballistic!).

We use PSCredential for all credentials, windows or sql. The variable name was simply a copy from the example.

There's no place like 127.0.0.1

Tuesday, August 8, 2017 12:29 PM



FWN die netzwerker Computern... 30,482 Points

0

[Sign in  
to vote](#)

Here is the connection string for an encrypted connection.

**"User ID=jsmith;Initial Catalog=master;Data Source=OMEGA\SQLEXPRESS;Use Encryption for Data=True"**

Create a file on your desktop called "test.udl". Double click the file and it will open the connection string wizard. You will be able to see and set all options for all installed providers and copy the string. The DBATools module is nice for many things but completely unnecessary for simple queries.

---

\(˘)₊

It *may* be a slight case of overkill. Overall, I think it's handy to have around when having to deal with SQL even only on an infrequent basis.

Admittedly, me being a contributor and team member, I may be slightly biased ;)

---

There's no place like 127.0.0.1

Tuesday, August 8, 2017 12:32 PM



**FWN** die netzwerker Computern... 30,482 Points

---

0

[Sign in](#)  
[to vote](#)

he documentation explicitly states "Windows Credentials" and not "SQLServer authentication".

Look at the code. If the cred is not being used to extract the creds and set SQLServer login then it won't work.

See this:

```
PS D:\scripts> $srv = new-object Microsoft.SqlServer.M
PS D:\scripts> $srv.Credentials=Get-Credential jsmith
'Credentials' is a ReadOnly property.
```

To use this you would need to create a proper connect string that uses a username and password.

\(ツ)/

Tuesday, August 8, 2017 12:44 PM



jrv Consultant 239,063 Points

0

[Sign in](#)  
[to vote](#)

This is how we would have to use a SQLServer authenticated account with SMO.

```
$srv = [Microsoft.SqlServer.Management.Smo.Server]::
$srv.ConnectionContext.LoginSecure = $false
$srv.ConnectionContext.Login = 'jsmith'
$srv.ConnectionContext.Password = 'MyPassword'
$srv.ConnectionContext.ServerInstance = 'omega\sqllex
$srv
```

\(ツ)/

Tuesday, August 8, 2017 12:58 PM



jrv Consultant 239,063 Points

Just sayin', but ... <function code>

```
.PARAMETER Credential  
Credential object used to connect to the SQL Server
```

0

[Sign in  
to vote](#)

And the code does it pretty much like in your example ...

There's no place like 127.0.0.1

Tuesday, August 8, 2017 1:03 PM



[FWN](#) die netzwerker Computern... 30,482 Points

We can also add encryption to the SMO connection:

```
$srv.ConnectionContext.EncryptConnection = $true
```

\('▽')/

0

[Sign in  
to vote](#)

Tuesday, August 8, 2017 1:03 PM



[jrv](#) Consultant 239,063 Points



0

[Sign in  
to vote](#)

Just sayin', but ... <[function code](#)>

```
.PARAMETER Credential  
Credential object used to connect to the SQL S
```

And the code does it pretty much like in your example ...

There's no place like 127.0.0.1

I missed that. Haven't tested it. I generally just use SMO directly as it has all elements easily exposed. DBATools is useful and will get better I am sure.

\(˘ᗜ˘)/

Tuesday, August 8, 2017 1:10 PM



[jrv](#) Consultant 239,063 Points

0

[Sign in  
to vote](#)

Just sayin', but ... <[function code](#)>

```
.PARAMETER Credential  
Credential object used to connect to the SQL S
```

And the code does it pretty much like in your example ...

There's no place like 127.0.0.1

How does the CmdLet determine which credential you are trying to use? Does it try both? It might be better to provide a switch for SQLAuthentication.

\(˘ᗜ˘)/

Tuesday, August 8, 2017 1:13 PM



[jrv](#) Consultant 239,063 Points

0

[Sign in  
to vote](#)

Just sayin', but ... <[function code](#)>

```
.PARAMETER Credential  
Credential object used to connect to the SQL S
```

And the code does it pretty much like in your example ...

There's no place like 127.0.0.1

How does the CmdLet determine which credential you are trying to use? Does it try both? It might be better to provide a switch for SQLAuthentication.

\(ツ)/

It would be better in a technical sense.

Given our target audience ... it must be automatic. Currently we do it solely by checking for "\" within the name (ugh!), but a revision is pending ... it's just so much yet to do, it's taking a bit of time. And it will include a parameter for forcing the type, with native autodetection if you don't use it when we're done.

There's no place like 127.0.0.1

Tuesday, August 8, 2017 1:21 PM



**FWN** die netzwerker Computern... 30,482 Points

0

[Sign in  
to vote](#)

Given our target audience ... it must be automatic. Currently we do it solely by checking for "\" within the name (ugh!), but a revision is pending ... it's just so much yet to do, it's taking a bit of time. And it will include a parameter for forcing the type, with native autodetection if you don't use it when we're done.

---

There's no place like 127.0.0.1

Sounds good. I'll keep an eye on the revisions to see how you guys make out. It would definitely be useful for simplifying the complexity of SMO for managing instances and data centers.

---

\(ツ)/

Tuesday, August 8, 2017 1:24 PM



[jrv](#) Consultant

239,063 Points

0

[Sign in  
to vote](#)

Sounds good. I'll keep an eye on the revisions to see how you guys make out. It would definitely be useful for simplifying the complexity of SMO for managing instances and data centers.

:)

In case you are curious enough to give it a look: We have a funny thing going on:

The [DbalInstanceparameter] type, a flexible input adapter. If you manually connect using Connect-DbasqlServer, you can give any of our function the object as input and it will get reused. Give it a string and it will take it (you can also give it ADComputer objects, if the default instance is alright with you). Was rather helpful when implementing the other functions, as we can't possibly provide all the connection options on every single function ...

---

There's no place like 127.0.0.1

Tuesday, August 8, 2017 1:50 PM



[FWN](#) die netzwerker Computern...

30,482 Points

0

[Sign in  
to vote](#)

Sounds good. I'll keep an eye on the revisions to see how you guys make out. It would definitely be useful for simplifying the complexity of SMO for managing instances and data centers.

;)

In case you are curious enough to give it a look: We have a funny thing going on:

The [DbalInstanceparameter] type, a flexible input adapter.

If you manually connect using Connect-DbasqlServer, you can give any of our function the object as input and it will get reused. Give it a string and it will take it (you can also give it ADComputer objects, if the default instance is alright with you). Was rather helpful when implementing the other functions, as we can't possibly provide all the connection options on every single function ...

---

There's no place like 127.0.0.1

How much to overload a parameter before it is a liability; this is a common quandary. It is useful and can make things easier but can also make maintenance and documentation a headache. I always get caught in a loop when I consider this and similar issues of design. In this case it makes the CmdLets friendly. It can make the CmdLet more useful in pipelines.

---

\('ㄟ')/

Tuesday, August 8, 2017 4:53 PM



[jrv](#) Consultant

239,063 Points

0

[Sign in  
to vote](#)

How much to overload a parameter before it is a liability; this is a common quandary. It is useful and can make things easier but can also make maintenance and documentation a headache.

True dat.

On the other hand, it also makes maintenance *easier*. You have to maintain the c# code (we have a full library in a VS2017 library for that), on the other hand, any updates and fixes automatically propagate to all functions. Doing this in script could require manual maintenance of all 300+ functions.

There's no place like 127.0.0.1

Wednesday, August 9, 2017 7:33 AM



**FWN** die netzwerker Computern... 30,482 Points