**[PowerShell Script To Get SQL Agent Job Statuses](http://sqlservertimes2.com/?p=389)**

by [Tim and Lori Edwards](http://sqlservertimes2.com)

Several weeks ago on Twitter, Colin Stasiuk ([Blog](http://www.BenchmarkITConsulting.com) – [Twitter](http://www.twitter.com/BenchmarkIT)), ) asked if anyone had a script to pull back job statuses from the SQL Server Agent. I had been doing some work on a SQL Server DBA Console and had written some PowerShell scripts to give me various pieces of information and put together a PowerShell script that could be run as a SQL Agent job to periodically report the statuses of SQL Agent jobs. Eventually, I think Colin went with a different solution, but I figured I would go ahead and post the PowerShell script that I came up with. This solution has been tested against SQL Server 2000/2005/2008.

This first script is just a SQL script to create the SQL Server table that the PowerShell script writes the status information to and can be downloaded here, [LastJobStatus\_Table.sql](http://sqlservertimes2.com/wp-content/uploads/2010/01/LastJobStatus_Table.sql).

[view source](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#viewSource)



[print](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#printSource)[?](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#about)

1.CREATE TABLE [dbo].[LastJobStatus](

2. [ServerName] [nvarchar](128) NULL,

3. [Job\_Name] [nvarchar](128) NULL,

4. [Run\_Date] [datetime] NULL,

5. [Job\_Duration] [time](7) NULL,

6. [Run\_Status] [varchar](50) NULL,

7. [Sample\_Date] [datetime] NULL

8.) ON [PRIMARY]

The next is the PowerShell script that does all of the work bringing back the SQL Agent job statuses. It takes a parameter of Server, which is the name of the SQL Server that you want job statuses from. Make sure that you change the name “MgtServer” to whatever the name is of the SQL Server where you intend to store the results from this script. You’ll also need to change the root directory for where your scripts are loaded to match your environment. This script can be downloaded here, [Get-LastJobStatusServer.ps1](http://sqlservertimes2.com/wp-content/uploads/2010/01/Get-LastJobStatusServer.ps1).

[view source](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#viewSource)



[print](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#printSource)[?](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#about)

01.param([string]$Server=$(Throw "Parameter missing: -Server ServerName"))

02.[void][reflection.assembly]::LoadWithPartialName("Microsoft.SqlServer.Smo")

03.

04.##Point to library files

05.$scriptRoot = "D:\DBAScripts"

06.

07.--change script root directory to match your environment

08.#$scriptRoot = Split-Path (Resolve-Path $myInvocation.MyCommand.Path)

09.. $scriptRoot\LibrarySmo.ps1 #Part of Chad Miller's SQLPSX project on CodePlex

10.. $scriptRoot\DataTable.ps1 #Included with the scripts for this blog, also from a CodePlex project (http://www.codeplex.com/PSObject)

11.

12.Set-Alias -Name Test-SqlConn -Value D:\DBAScripts\Test-SqlConn.ps1

13.

14.##Define variables

15.

16.## open database connection

17.$conn = New-Object System.Data.SqlClient.SqlConnection("Data Source=$server;

18.Initial Catalog=master; Integrated Security=SSPI")

19.$conn.Open()

20.$cmd = $conn.CreateCommand()

21.$cmd.CommandText= " CREATE TABLE #JobsRun ( ServerName nvarchar(128),

22. Job\_Name nvarchar(128),

23. Run\_Date datetime,

24. Job\_Duration time(7),

25. Run\_Status varchar(50),

26. Sample\_Date datetime

27. );

28. insert into #JobsRun

29. select @@SERVERNAME AS ServerName

30. ,j.name Job\_Name

31. ,(msdb.dbo.agent\_datetime(jh.run\_date,jh.run\_time)) As Run\_Date

32. ,substring(cast(run\_duration + 1000000 as varchar(7)),2,2) + ':' +

33. substring(cast(run\_duration + 1000000 as varchar(7)),4,2) + ':' +

34. substring(cast(run\_duration + 1000000 as varchar(7)),6,2) Job\_Duration

35. ,case when run\_status = 0

36. then 'Failed'

37. when run\_status = 1

38. then 'Succeed'

39. when run\_status = 2

40. then 'Retry'

41. when run\_status = 3

42. then 'Cancel'

43. when run\_status = 4

44. then 'In Progress'

45. end as Run\_Status

46. ,GETDATE() As Sample\_Date

47. FROM msdb.dbo.sysjobhistory jh

48. join msdb.dbo.sysjobs j

49. on jh.job\_id = j.job\_id

50. where step\_id = 0

51. and enabled = 1

52. order by cast(cast(run\_date as char) + ' ' +

53. substring(cast(run\_time + 1000000 as varchar(7)),2,2) + ':' +

54. substring(cast(run\_time + 1000000 as varchar(7)),4,2) + ':' +

55. substring(cast(run\_time + 1000000 as varchar(7)),6,2) as datetime) desc

56.

57. delete from MgtServer.DBA\_Console.dbo.LastJobStatus where ServerName = '$server' -- Change 'MgtServer' to the name of whatever the SQL Server is in

58. -- your env that will house the LastJobStatus table which stores the

59. -- results of this script

60. insert into MgtServer.DBA\_Console.dbo.LastJobStatus (ServerName, Job\_Name, Run\_Date, Job\_Duration, Run\_Status, Sample\_Date)

61. select jr.ServerName,

62. jr.Job\_Name,

63. jr.Run\_Date,

64. jr.Job\_Duration,

65. jr.Run\_Status,

66. jr.Sample\_Date

67. from #JobsRun jr

68. where Run\_Date = ( select max(jr1.Run\_Date)

69. from #JobsRun jr1

70. where jr1.Job\_Name = jr.Job\_Name)

71. drop table #JobsRun; "

72.$cmd.ExecuteNonQuery()

73.$conn.Close()

There are references in the above script to a LibrarySMO.ps1 script that can be obtained from [CodePlex](http://sqlpsx.codeplex.com/) (see the comments in the script for URL) and a [DataTable.ps1](http://sqlservertimes2.com/wp-content/uploads/2010/01/DataTable.ps1) script (also from CodePlex, but included in the download file for this blog, for your convenience.)

[view source](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#viewSource)



[print](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#printSource)[?](http://sqlserverpedia.com/blog/uncategorized/powershell-script-to-get-sql-agent-job-statuses/#about)

01.# Taken from out-dataTable script from the PowerShell Scripts Project

02.# http://www.codeplex.com/PsObject/WorkItem/View.aspx?WorkItemId=7915

03.

04.Function out-DataTable {

05.

06. $dt = new-object Data.datatable

07. $First = $true

08.

09. foreach ($item in $input){

10. $DR = $DT.NewRow()

11. $Item.PsObject.get\_properties() | foreach {

12. if ($first) {

13. $Col = new-object Data.DataColumn

14. $Col.ColumnName = $\_.Name.ToString()

15. $DT.Columns.Add($Col) }

16. if ($\_.value -eq $null) {

17. $DR.Item($\_.Name) = "[empty]"

18. }

19. elseif ($\_.IsArray) {

20. $DR.Item($\_.Name) =[string]::Join($\_.value ,";")

21. }

22. else {

23. $DR.Item($\_.Name) = $\_.value

24. }

25. }

26. $DT.Rows.Add($DR)

27. $First = $false

28. }

29.

30. return @(,($dt))

31.

32.}