**Analyzing SQL Agent Job and Job Step History in SQL Server**

By: [Atif Shehzad](https://www.mssqltips.com/sqlserverauthor/23/atif-shehzad/)   |   Updated: 2009-05-15

FROM: <https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/>

**Problem**

Almost every DBA configures SQL Agent Jobs for automation of repetitive tasks. This valuable facility works in the background in SQL Server and saves a lot of effort and manual rework. You may have a number of jobs scheduled and often there is the need to analyze the data to see which jobs are taking a long time or which job steps are taking a long time.  As you add more jobs and overhead to the server these times become even more critical and analyzing the data is key.  SSMS offers some insight into your jobs, but there is no easy way to slice and dice the data to see what is going on for a particular job or at a particular time of the day.

Some of the questions you may want to ask include:

* How much difference in performance is there for the current scheduled time compared to changing the time to midnight?
* Exactly how much time is consumed now to complete the job?
* Have jobs performed better after changes or there is some negative impact?
* Which step of a specified job is taking longer than it takes normally to complete?

Questions like these require analysis performed over the history of the jobs. So how do you generate the required set of data in a form that is suitable for analysis?  Or a situation where someone does not have access to SSMS or EM to access the job history for verification and analysis?

**Solution**

To generate a comprehensive and descriptive set from the job history the following script can be run in SSMS/QA. It uses the system database msdb and some of the system tables related to jobs and job history.

--Script # 1: To generate steps history of all jobs

USE msdb

GO

SELECT j.name JobName,h.step\_name StepName,

CONVERT(CHAR(10), CAST(STR(h.run\_date,8, 0) AS dateTIME), 111) RunDate,

STUFF(STUFF(RIGHT('000000' + CAST ( h.run\_time AS VARCHAR(6 ) ) ,6),5,0,':'),3,0,':') RunTime,

h.run\_duration StepDuration,

case h.run\_status when 0 then 'failed'

when 1 then 'Succeded'

when 2 then 'Retry'

when 3 then 'Cancelled'

when 4 then 'In Progress'

end as ExecutionStatus,

h.message MessageGenerated

FROM sysjobhistory h inner join sysjobs j

ON j.job\_id = h.job\_id

ORDER BY j.name, h.run\_date, h.run\_time

GO

This script will return execution history of all jobs arranged by job, run date and run time. For each step in a job, related information is displayed to give you additional insight. We can analyze each job step and the execution time for each step. We can then see if there is an unusual duration of time consumed by this step versus other run times.

The Duration is presented in seconds. If the value is less than one second then zero is used for the duration.

For this example no filter has been applied, history of both enabled and disabled jobs is generated.

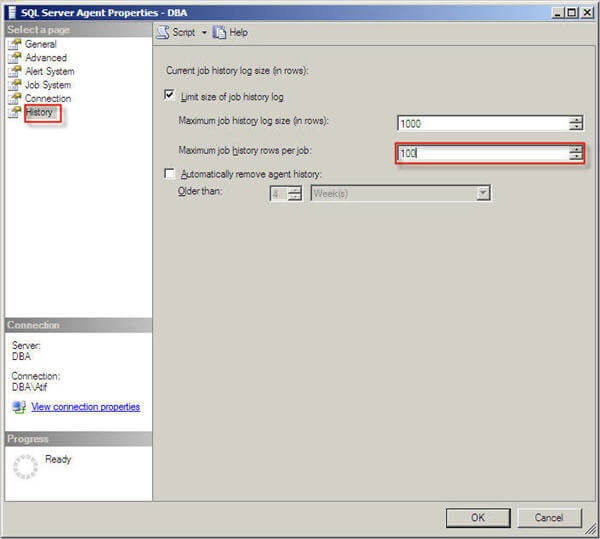


Two sets of job records have been highlighted above. Step *[Job outcome*] is associated with each set of steps in a job and its duration is shown as cumulative duration of all steps. Results of executing script # 1 will be different on your server according to your scheduled jobs.

Columns selected in script # 1 have the following descriptions.

|  |  |
| --- | --- |
| **Column** | **Description** |
| [JobName] | Name of job as specified |
| [StepName] | Name of step as specified |
| [RunDate] | Date when job run |
| [RunTime] | Time when job run |
| [StepDuration] | Duration in seconds that a step took to complete |
| [ExecutionStatus] | Execution status of step |
| [MessageGenerated] | Message generated at end of step |

By default SQL Server stores a maximum of 100 rows for a job. It is important to clarify that 100 rows are for the whole job not for a single step of a job. If you require more than 100 rows for analysis then you can change the default setting in the following way. Go to the Properties tab of SQL Server Agent in SSMS. Then you can configure the number of rows as shown in the screenshot below. After the change you will be able to store more historical rows for analysis as specified.



**Next Steps**

* Next time when you need to implement changes for your jobs do not forget to consider the impacts on your job schedule
* If you are required to maintain long term history data, then use the above mentioned method to increase the storage of history rows or you can create a separate archive table to keep this data to analyze
* To be more specific for a requirement, you can apply filters in the script on the *sysJobHistory* table. This table contains information about the execution of scheduled jobs by SQL Server Agent.
* You can also apply a filter on *run\_date*, *run \_time*, *name* or if you want apply a filter on run\_status of the jobs and use the following mapping
  + 0 - Failed
  + 1 - Succeeded
  + 2 - Retry
  + 3 - Canceled
  + 4 - In progress

Last Updated: 2009-05-15

|  |  |
| --- | --- |
| **Friday, June 28, 2019 - 9:29:21 AM - Nelida** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| I am Peruvian, I find excellent tip, very complete  Thanks for sharing knowledge | |

|  |  |
| --- | --- |
| **Monday, January 08, 2018 - 8:06:02 AM - Greg Robidoux** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Thanks Krishna.  This has been updated. | |

|  |  |
| --- | --- |
| **Monday, January 08, 2018 - 1:34:58 AM - Krishna** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Nice Article.    Note-Instead of GO you typed as GG in second line of the scripts. | |

|  |  |
| --- | --- |
| **Tuesday, March 01, 2016 - 12:43:26 AM - Atif Shehzad** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| @Devi. Thanks for adding more to code. | |

|  |  |
| --- | --- |
| **Friday, February 26, 2016 - 10:57:55 AM - Davi Durki** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Here's an expansion of the original code that lets you find out (historically) what jobs are stepping on each others' toes.  If you have a number of nightly or weekly jobs where the run time isn't always consistent, this is a great way to help you spread them out.  It's not perfect but perhaps enough to start with.  Thanks to Atif for the original code - it was a great starting point for me.  ---      with history\_L1 as (  SELECT top 100 percent j.name JobName,j.job\_id, h.step\_name StepName,  CONVERT(CHAR(10), CAST(STR(h.run\_date,8, 0) AS dateTIME), 111) RunDate,  STUFF(STUFF(RIGHT('000000' + CAST ( h.run\_time AS VARCHAR(6 ) ) ,6),5,0,':'),3,0,':') RunTime,  h.run\_duration StepDuration,  case h.run\_status when 0 then 'failed'  when 1 then 'Succeded'  when 2 then 'Retry'  when 3 then 'Cancelled'  when 4 then 'In Progress'  end as ExecutionStatus  FROM sysjobhistory h  inner join sysjobs j  ON j.job\_id = h.job\_id  where  j.name not in (  -- Throw out some irrelevant jobs like replication tasks or periodic transaction log backups.  'Agent history clean up: distribution', 'ASPstate: ASPState\_Job\_DeleteExpiredSessions', 'Distribution clean up: distribution',  'Expired subscription clean up', 'Replication agents checkup', 'syspolicy\_purge\_history', 'Transaction log backup.Subplan\_1'  )  ORDER BY j.name, h.run\_date, h.run\_time  )  select \*, CONVERT(datetime, rundate + ' ' + runtime) as StartDateTime, DATEADD(ss, StepDuration, CONVERT(datetime, rundate + ' ' + runtime)) as FinishDateTime  into #history  from history\_L1  where StepName not like '(job outcome)'  order by CONVERT(datetime, rundate + ' ' + runtime)      select distinct h1.JobName, h1.job\_id, h1.StartDateTime, h1.FinishDateTime, h2.JobName, h2.StartDateTime, h2.FinishDateTime  from #history h1  inner join #history h2  on h1.JobName <> h2.JobName  and h2.FinishDateTime >= h1.StartDateTime  and h2.StartDateTime <= h1.FinishDateTime    drop table #history | |

|  |  |
| --- | --- |
| **Friday, December 05, 2014 - 2:48:12 AM - Atif Shehzad** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| @Ankita. By default SQL Server stores a maximum of 100 rows for a job. It is important to clarify that 100 rows are for the whole job not for a single step of a job. If you require more than 100 rows for analysis then you can change the default setting (as mentioned in last part of article). Go to the Properties tab of SQL Server Agent in SSMS. Then you can configure the number of rows as shown in the screenshot below. After the change you will be able to store more historical rows for analysis as specified. | |

|  |  |
| --- | --- |
| **Thursday, December 04, 2014 - 10:15:42 AM - Ankita Bikrol** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Can you please tell me for how many day data is stored in these tables? Can I see the history of past 30-60 days? | |

|  |  |
| --- | --- |
| **Monday, September 08, 2014 - 9:02:27 AM - Anusha** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Hi Atif,  Thank you so much for your script.  But when I run the script am getting error like    A fatal scripting error occurred.  Incorrect syntax was encountered while parsing Go.    Request you to please let me know how to resolve this issue.  Awaiting your reply. | |

|  |  |
| --- | --- |
| **Friday, September 05, 2014 - 5:28:42 PM - Henry B. Stinson** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| You should precalcuate the DATEPART( ) function calls into local variables to lessen the RBAR procesing.  Of course there is still the SUBSTRING( ) and CONVERT( ) function calls.  You might even write a function to convert h.run\_date to a standard DATE or DATETIME type. | |

|  |  |
| --- | --- |
| **Monday, September 03, 2012 - 1:51:40 AM - Atif Shehzad** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| @Dinesh. You may add following conditions to get the executions of current day only  and substring(convert(varchar(8),h.run\_date),1,4) = datepart(yy,getdate()) and substring(convert(varchar(8),h.run\_date),5,2) = datepart(mm,getdate()) and substring(convert(varchar(8),h.run\_date),7,2) = datepart(dd,getdate())    Thanks | |

|  |  |
| --- | --- |
| **Friday, August 31, 2012 - 12:20:40 AM - Dinesh** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| If i want only todays date then ? | |

|  |  |
| --- | --- |
| **Thursday, August 23, 2012 - 11:20:31 AM - GregJ** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| THANK YOU!  *Exactly* what I needed. | |

|  |  |
| --- | --- |
| **Tuesday, May 29, 2012 - 4:08:14 AM - sneha** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Thank you it usefull for me | |

|  |  |
| --- | --- |
| **Thursday, December 29, 2011 - 10:04:57 AM - Ankit Shah** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Hi Atif,   thanks for kind help . Let me know if you can work out on my idea.  Thanks for the link. | |

|  |  |
| --- | --- |
| **Thursday, December 29, 2011 - 2:33:15 AM - Atif Shehzad** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Hi Ankit,  I have noticed a tip similar to your requirement. Please have a look at http://www.mssqltips.com/sqlservertip/2347/send-email-in-a-tabular-format-using-sql-server-database-mail  It would not mail the result as attachment but as a table in email. I would work on your idea and would let you know as it is done.  Thanks | |

|  |  |
| --- | --- |
| **Wednesday, December 28, 2011 - 4:11:13 PM - Ankit Shah** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Hi Atif ,  Nice post and pretty smoothly working. well i want to setup kind that the Query Result of the query you scripted i want to get in email with excel sheet. Do you know how to do it?  It woluld be great if you provide me steps | |

|  |  |
| --- | --- |
| **Tuesday, March 22, 2011 - 2:01:25 AM - Atif Shehzad** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| @Erasmus. The "step duration" field is displayed without any conversion/formating. BOL gives its description as "Elapsed time in the execution of the job or step in **HHMMSS** format". And it is of integer data type. So in your case 130 is actually 1 minute 30 seconds. For example if you execute following code in any job step  WAITFOR DELAY '00:01:30' select 'ok'    the resultatnt step duration would be 130 i.e. 1 minute and thirty seconds.  Thanks | |

|  |  |
| --- | --- |
| **Monday, March 21, 2011 - 5:06:06 AM - Elmo Erasmus** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| I really like this tip. Just a question. The "StepDuration" field - it seems in my query it does not actually give the seconds that the step took. For example 130 is actually 1:30 or 1 minute 30 seconds. How can we get query to display 90?   |  |  | | --- | --- | | StepDuration] | Duration in seconds that a step took to complete | | |

|  |  |
| --- | --- |
| **Thursday, October 22, 2009 - 2:35:25 AM - @tif** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| @okiftequar. Are u asking about assoiated failure message that we can see in windows event viewer? | |

|  |  |
| --- | --- |
| **Thursday, October 22, 2009 - 2:19:31 AM - wwphx** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| [quote user="okiftequar"]  Upto normal point of view its very nice. But i am facing a problem. I want to view Warning Jobs Messages also in the same query.  So how i can view the warnings jobs in sql server 2005 by sql query. I am getting the messages of failure and success but how i can get the job warnings by sql query. Pls its urgent.[/quote]  I'm sorry, M.I., I don't quite follow what you're asking for.  Maybe it's just that it's late and my brain is slow, but are you asking for SQL Agent warnings?  Those are the only warnings that I can think of off-hand.  As far as I can recall, SQL is pretty much binary: it succeeds or fails, no success but with warnings. | |

|  |  |
| --- | --- |
| **Wednesday, October 21, 2009 - 10:03:43 AM - okiftequar** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Upto normal point of view its very nice. But i am facing a problem. I want to view Warning Jobs Messages also in the same query.  So how i can view the warnings jobs in sql server 2005 by sql query. I am getting the messages of failure and success but how i can get the job warnings by sql query. Pls its urgent.   Thanks in advance...     M.I. | |

|  |  |
| --- | --- |
| **Friday, May 29, 2009 - 10:49:34 AM - wwphx** | [Back To Top](https://www.mssqltips.com/sqlservertip/1752/analyzing-sql-agent-job-and-job-step-history-in-sql-server/#top) |
| Nice little query and tip, Atif.  I'm quite fond of working with system tables and views to extract information like this, and I ALWAYS change the number of rows stored for job history, typically to 9999.  Obviously for comparison purposes you'd need to archive this information in order to see jobs or steps that are suddenly taking longer to execute. If you did archive this information, you might consider keeping the date/time fields as integers to save a bit of space, I've seen performance-gathering databases get obnoxiously huge. | |