

msdb.dbo.backupset Slow (msdb..backupset)

Database Health monitor will report when the performance is slow accessing msdb.dbo.backupset from the Database Overview page. You may be seeing a message like the following:

Warnings and Errors	
Warning	MoreInfo
Log file size of 1.1MB appears unusually large compared to the data file size of 4.1MB	http://databasehealth.com/unusually-large-log
Log or data files on the C Drive could cause the server to run out of disk space and crash.	http://databasehealth.com/Data-Or-Log-On-CDrive
DB Collation(Latin1_General_CI_AS_KS_WS) differs from TempDB Collation (SQL_Latin1_...	http://databasehealth.com/tempDB-Collation-Mismatch
No full backup in last 24 hours	http://databasehealth.com/no-full-backups-in-24-hours
msdb.dbo.backupset slow	http://databasehealth.com/msdb.dbo.backupset-slow
Warning Cache Hit Ratio is low.	http://databasehealth.com/Low-Cache-Hit-Ratio

The table msdb.dbo.backupset grows over time, whenever a full, incremental or log backup is run a row is inserted into the backupset table. This will continue to grow overtime if you do not clean it up.

To check the size of your msdb.dbo.backupset table, just run the following query.

```
SELECT count(*)
FROM msdb.dbo.backupset;
```

The screenshot shows the SQL Server Enterprise Manager interface. On the left, the Object Explorer displays the 'server2012 (SQL Server 11.0.2218 - sa)' instance. Under 'Databases', 'msdb' is expanded, and 'dbo.backupset' is selected. On the right, the SQL Query window shows the query: `SELECT count(*) FROM msdb.dbo.backupset;`. The Results pane shows a single row with the value 264482. The status bar at the bottom indicates 'server2012 (11.0 RTM) | sa (53) | msdb | 00:00:05 | 1 rows'.

msdb.dbo.backupset or msdb..backupset

Whats the difference between msdb.dbo.backupset and msdb..backupset? Well they both reference the same table. The double dot syntax implies the default schema which is dbo on the msdb database. So either

way msdb.dbo.backupset and msdb..backupset are the same table.

How many rows is too many?

The question of too many really depends on the performance of your SQL Server. If you are able to query this table, and get speedy results (less than a second), then you may not have too many rows, however if just counting the number of rows takes more than a couple seconds as shown in the screen shot, then you may have too many rows.

Also, you may want to ask yourself how many backups do you need to know about. You can use msdb.dbo.backupset to see the history of your backups, do you care about how often a log backup was run 3 years ago? If you don't care then you may want to purge old data, if you do care, then you may want to copy the backup history off to another table.

Purging History

If you just try to delete from the msdb..backupset table you will get errors about foreign key violations. Instead there is a stored procedure available in the msdb database called sp_delete_backup history that will do the cleanup for you. You give it a date and it will delete everything prior to that date.

```
EXEC msdb..sp_delete_backuphistory '2015-01-01';
```

If you have years of data in this table, it may take a very long time to do the cleanup, so you might want to start by deleting a smaller batch at a time. You could do something like this:

```
EXEC msdb..sp_delete_backuphistory '2010-01-01';
EXEC msdb..sp_delete_backuphistory '2011-01-01';
EXEC msdb..sp_delete_backuphistory '2012-01-01';
EXEC msdb..sp_delete_backuphistory '2013-01-01';
EXEC msdb..sp_delete_backuphistory '2014-01-01';
EXEC msdb..sp_delete_backuphistory '2015-01-01';
```

Or instead of hard coding a date to pass into sp_delete_backup history you could use a parameter, for instance the following deletes everything older than 30 days.

```
DECLARE @date AS NVARCHAR(20);
SET @date = CAST(CAST(GETDATE() - 30 AS DATE) AS VARCHAR);
EXEC msdb..sp_delete_backuphistory @date;
```

Recommendations

It is a good idea to have a regular job that will purge the msdb.dbo.backupset after a period of time.

If you want to delete backup history older than 180 days you can schedule a task that runs this TSQL script:

```
DECLARE @date AS NVARCHAR(20);  
SET @date = CAST(CAST(GETDATE() - 180 AS DATE) AS VARCHAR);  
EXEC msdb..sp_delete_backuphistory @date;
```

Need Help?

If you need help with this or with anything else related to SQL Server performance and tuning, you can contact [Stedman Solutions](#) to help with your SQL Server needs.

Database Health Monitor

If you are not using [Database Health Monitor](#) take a look at the [free download](#) to help analyze your SQL Server performance.