**Support for 15,000 Partitions in SQL Server 2008 SP2 and SQL Server 2008 R2 SP1**

|  |  |
| --- | --- |
|  | logo-sql08.gif |

SQL Server Technical Article

**Writers:** Pooja Harjani, Susan Price, Jos de Bruijn - Microsoft Corporation

**Technical Reviewers:** Hardik Bati, Eric Hanson, Hermann Daeubler, Juergen Thomas, Cameron Gardiner - Microsoft Corporation

**Published:** September 2010 **Updated:** March 2011

**Applies to:** SQL Server 2008 with Service Pack 2 and later
 SQL Server 2008 R2 with Service Pack 1 and later

**Summary:** In SQL Server 2008 and SQL Server 2008 R2, the number of partitions on tables and indexes is limited to 1,000. This paper discusses how SQL Server 2008 SP2 and SQL Server 2008 R2 SP1 address this limitation by providing an option to increase the limit to 15,000 partitions. It describes how support for 15,000 partitions can be enabled and disabled on a database. It also talks about performance characteristics, certain limitations associated with this support, known issues, and their workarounds. This support is targeted to enterprise customers and ISVs with large-scale decision support or data warehouse requirements.

Copyright

The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This white paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED, OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in, or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

© 2010 Microsoft Corporation. All rights reserved.

Microsoft and SQL Server are trademarks of the Microsoft group of companies.

Contents

[Introduction 4](#_Toc288495742)

[Problem 4](#_Toc288495743)

[Solution 4](#_Toc288495744)

[sp\_db\_increased\_partitions Stored Procedure 4](#_Toc288495745)

[Arguments 5](#_Toc288495746)

[Permission 5](#_Toc288495747)

[Supported Versions 5](#_Toc288495748)

[Supported Editions 5](#_Toc288495749)

[Enabling Support for 15,000 Partitions 5](#_Toc288495750)

[Disabling Support for 15,000 Partitions 6](#_Toc288495751)

[Impact on Database Migration/Upgrade 7](#_Toc288495752)

[Migrating a Database Across Versions of SQL Server 7](#_Toc288495753)

[Migrating from SQL Server 2008 SP2 to SQL Server 2008 R2 RTM 7](#_Toc288495754)

[Migrating from SQL Server 2008 SP2 to SQL Server 2008 R2 SP1 8](#_Toc288495755)

[Migrating a Database Across Service Packs for the Same Version of SQL Server 9](#_Toc288495756)

[Impact on Log Shipping 9](#_Toc288495757)

[Impact on Database Mirroring 10](#_Toc288495758)

[Enabling Support for 15,000 Partitions on a Database in a Mirrored Environment 10](#_Toc288495759)

[Disabling Support for 15,000 Partitions on a Database in a Mirrored Environment 10](#_Toc288495760)

[Impact on Replication 10](#_Toc288495761)

[Impact on Manageability 10](#_Toc288495762)

[Impact on Performance 10](#_Toc288495763)

[Memory 11](#_Toc288495764)

[Partitioned Indexes 11](#_Toc288495765)

[Workaround 11](#_Toc288495766)

[DBCC 11](#_Toc288495767)

[Queries 11](#_Toc288495768)

[Conclusion 11](#_Toc288495769)

# Introduction

Support for a maximum of 15,000 partitions in tables and indexes has been introduced in Microsoft SQL Server 2008 Service Pack 2 (SP2) in the Enterprise, Developer, and Evaluation editions. This support can be enabled at a database-level granularity. When enabled, a partition function can be created or altered to generate a maximum of 15,000 partitions. This article provides details on how support for 15,000 partitions can be enabled and disabled, its limitations, and some known issues and workarounds.

This support is targeted to enterprise customers and ISVs with large-scale decision support or data warehouse requirements. Before using this support, you must understand the impact of its limitations and known issues.

# Problem

SQL Server 2005 introduced table and index partitioning. Partitioning can make large tables and indexes more manageable and scalable. For more information about partitioning, see [Partitioned Tables and Indexes](http://msdn.microsoft.com/en-us/library/ms188706%28v%3DSQL.100%29.aspx) (http://msdn.microsoft.com/en-us/library/ms188706(v=SQL.100).aspx). In SQL Server 2005, SQL Server 2008, and SQL Server 2008 R2, the number of partitions is limited to 1,000.

Customers primarily use partitioning to facilitate the management of large fact tables in data warehouses. Data warehouse customers commonly load data as a batch. Daily loads are the most common pattern, but increasingly customers want to load data more than once a day. With the limit of 1,000 partitions, if customers load daily, they can store less than three years of data in a partitioned table, whereas business requirements often mandate that data be retained for longer periods of time, such as seven years. The 1,000 partitions maximum becomes a limitation for customers in this scenario.

If merging of partitions is too complex and time-consuming, customers prefer to have the flexibility to create a large number of partitions and use them as and when required. The 1,000 partitions maximum also becomes a limitation in this scenario.

# Solution

In SQL Server 2008 SP2 and SQL Server 2008 R2 SP1, you can choose to enable support for 15,000 partitions at a database-level granularity by using the new **sp\_db\_increased\_partitions** stored procedure. You can also disable support on a database (after it has been enabled) and set the limit on the number of partitions back to 1,000.

# sp\_db\_increased\_partitions Stored Procedure

sp\_db\_increased\_partitions [@dbname = ] ‘dbname’

[, [@increased\_partitions = ] increased\_partitions ]

## Arguments

[@dbname=] ‘*dbname*’ is the name of the database for which support for 15,000 partitions is to be checked, enabled, or disabled. *dbname* is **sysname**, with a default of NULL. If *dbname* is NULL, it indicates the current database.

 [@increased\_partitions=] ‘*increased\_partitions*’ is the argument used to enable or disable support for 15,000 partitions on the specified database. *increased\_partitions* is **varchar**(6), with a default of NULL. If *increased\_partitions* is NULL, it outputs the status of support for 15,000 partitions on the specified database.

*increased\_partitions* can be one of the following values.

|  |  |
| --- | --- |
| Value | Description |
| ‘ON’ or ‘TRUE’ | Enables support for 15,000 partitions on the specified database |
| ‘OFF’ or ‘FALSE’ | Disables support for 15,000 partitions on the specified database |

## Permission

ALTER DATABASE permission on the database is required to run the stored procedure **sp\_db\_increased\_partitions**.

## Supported Versions

The stored procedure **sp\_db\_increased\_partitions** is available only in SQL Server 2008 SP2 and SQL Server 2008 R2 SP1. It is not available in SQL Server 2008 R2 RTM.

## Supported Editions

Like partitioned tables and indexes, the stored procedure **sp\_db\_increased\_partitions** is available only in the Datacenter, Enterprise, Developer, and Evaluation editions of SQL Server.

# Enabling Support for 15,000 Partitions

To enable support for 15,000 partitions on a database, use the stored procedure **sp\_db\_increased\_partitions**. The command is as follows:

exec sp\_db\_increased\_partitions '<database-name>', 'ON'

The output is:

|  |
| --- |
| increased\_partitions |
| 1 |

When the command is executed, SQL Server internally increases the database version number. The database version must be increased to indicate that the data in this database can have more than 1,000 partitions.

Upgrading to a newer database version prevents the migration of a database enabled for 15,000 partitions to earlier versions of SQL Server 2008 and SQL Server 2008 R2, because versions released prior to SQL Server 2008 SP2 and SQL Server 2008 R2 SP1 are unable to interpret the increased number of partitions. You can enable support for 15,000 partitions only on user databases, not on system databases such as **master** and **tempdb.** The distribution database that is used for replication cannot be enabled for 15,000 partitions.

To find out whether a database has support for 15,000 partitions enabled, use the following command:

exec sp\_db\_increased\_partitions '<database-name>'

The output is:

|  |
| --- |
| increased\_partitions |
| 1 |

Changing support for 15,000 partitions cannot be performed inside a user transaction. Changing the state to enable or disable support for 15,000 partitions requires an exclusive lock on the database during the entire process. This command will fail if there are active users in the database because this prevents the exclusive lock on the database.

# Disabling Support for 15,000 Partitions

To disable support for 15,000 partitions on a database, use the stored procedure **sp\_db\_increased\_partitions**. The command is as follows:

exec sp\_db\_increased\_partitions '<database-name>', 'OFF'

The output is:

|  |
| --- |
| increased\_partitions |
| 0 |

Before disabling support for 15,000 partitions, you must reduce the number of partitions on tables and indexes in the database to 1,000 or less by merging partitions or dropping the partition function. When support for 15,000 partitions is disabled, the database version is decreased to a number supported by SQL Server 2008, SQL Server 2008 SP1, and SQL Server 2008 R2 RTM.

You must also set the database to the simple recovery model before you disable support for 15,000 partitions on the database. The reason for this is to break the log chain. Consider the following scenario:

1. Do a full physical backup of a database (DB1) on SQL Server 2008 SP1. This backup will have the database version that is supported by SQL Server 2008 SP1.
2. Upgrade the instance to SQL Server 2008 SP2.
3. Enable support for 15,000 partitions on the database. Enabling support increases the database version number.
4. Create a table with 15,000 partitions and insert some data into those partitions.
5. Mark down time T.
6. Reduce the number of partitions on the database to 1,000 or fewer by merging partitions or dropping the partition function.
7. Disable support for 15,000 partitions on the database. Disabling support decrements the database version.
8. Take a log backup. This log backup will have the database version that is supported by SQL Server 2008 SP2 and SQL Server 2008 SP1.
9. Restore DB1 on SQL Server 2008 SP1, starting from full and then continuing with log backup with a STOPAT clause for time T.

This sequence creates a database with more than 1,000 partitions on a system that does not support it. To prevent a sequence of actions that could result in more than 1,000 partitions in a database with a version number that does not indicate support for more than 1,000 partitions, you are required to set the database to simple recovery model to break the log chain before you disable support for 15,000 partitions on the database.

# Impact on Database Migration/Upgrade

This section describes the impact on database migration in the context of support for 15,000 partitions. Specifically, it explores the following scenarios:

* Migrating a database across versions of SQL Server
* Migrating a database across service packs for the same version of SQL Server

## Migrating a Database Across Versions of SQL Server

You can migrate a database that was created in an earlier version of SQL Server to a later version. This is a typical database upgrade and is supported. However, you cannot migrate a database that was created in a later version of SQL Server to an earlier version.

For example, suppose that you want to migrate a SQL Server 2005 database to a SQL Server 2008 or a SQL Server 2008 R2 instance. These are typical database upgrade scenarios and are supported. Like SQL Server 2008 SP1, SQL Server 2008 SP2 and SQL Server R2 SP1 support rolling upgrades. For information about rolling upgrades, see SQL Server 2008 and SQL Server 2008 R2 Books Online. By default, a database does not have support for 15,000 partitions enabled when you upgrade it to a SQL Server 2008 SP2 or SQL Server 2008 R2 SP1 instance.

### Migrating from SQL Server 2008 SP2 to SQL Server 2008 R2 RTM

Suppose that you want to migrate a SQL Server 2008 SP2 database to a SQL Server 2008 R2 RTM instance. If support for 15,000 partitions has not been enabled on the database in the SQL Server 2008 SP2 instance, there are no restrictions on migrating the database to SQL Server 2008 R2 RTM. However, if support for 15,000 partitions is enabled, you cannot migrate the database to an instance of SQL Server 2008 R2 RTM. Because SQL Server 2008 R2 RTM does not support more than 1,000 partitions, the database with support for 15,000 partitions enabled has a later version than the SQL Server 2008 R2 RTM instance, which prevents migration. SQL Server 2008 SP2 databases with support for 15,000 partitions have version number 662, while SQL Server 2008 R2 RTM supports databases with version numbers up to and including 661.

If you have an instance of SQL Server 2008 SP2 where support for 15,000 partitions has been enabled on a database and the instance is upgraded to SQL Server 2008 R2 RTM, the database cannot be opened and you will not be able to use the database.

#### Workaround 1

If you have not upgraded the instance to SQL Server 2008 R2 RTM, follow these steps:

1. Reduce the number of partitions on tables and indexes in the database to 1,000 or fewer by merging partitions or dropping the partition function.
2. Disable support for 15,000 partitions on the database by running the stored procedure that will downgrade the database version.
3. Upgrade the instance to SQL Server 2008 R2 RTM.

#### Workaround 2

If you have already upgraded the instance to SQL Server 2008 R2 RTM, follow these steps:

1. Take the database offline.
2. Detach the database from the SQL Server 2008 R2 RTM instance.
3. Attach the database to a separate SQL Server 2008 SP2 instance.
4. Reduce the number of partitions on tables and indexes in the database to 1,000 or fewer by merging partitions or dropping the partition function.
5. Disable support for 15,000 partitions on the database by running the stored procedure that will downgrade the database version.
6. Detach the database from the SQL Server 2008 SP2 instance.
7. Attach the database to the SQL Server 2008 R2 RTM instance.

### Migrating from SQL Server 2008 SP2 to SQL Server 2008 R2 SP1

There are no restrictions on migrating the database to SQL Server 2008 R2 SP1.

If you have an instance of SQL Server 2008 SP2, there are two ways to upgrade the instance to SQL Server 2008 R2 SP1.

#### Option 1

Upgrade the SQL Server 2008 SP2 instance to SQL Server 2008 R2 RTM and then install SQL Server 2008 R2 SP1.

If you have an instance of SQL Server 2008 SP2 where support for 15,000 partitions has been enabled on a database and the instance is upgraded to SQL Server 2008 R2 RTM, the database cannot be opened and you will not be able to use the database. After installing SQL Server 2008 R2 SP1 over SQL Server 2008 R2 RTM, you will be able to use the database.

#### Option 2

Upgrade the SQL Server 2008 SP2 instance using the slipstream method to install SQL Server 2008 R2 RTM and SQL Server 2008 R2 SP1 together. For more information about slipstreaming, refer to the [SQL Server Setup](http://blogs.msdn.com/b/petersad/) (<http://blogs.msdn.com/b/petersad>) blog.

## Migrating a Database Across Service Packs for the Same Version of SQL Server

An example of this type of migration is migrating a database from SQL Server 2008 SP1 to SQL Server 2008. SQL Server usually allows you to migrate a database created on a service pack instance to an instance without the service pack or with a different service pack because the database version number typically does not change between different service packs of the same SQL Server version. However, SQL Server 2008 SP2 and SQL Server 2008 R2 SP1 are exceptions. You cannot migrate a SQL Server 2008 SP2 database that has support for 15,000 partitions enabled to an instance of SQL Server 2008 or SQL Server 2008 SP1. Similarly, you cannot migrate a SQL Server 2008 R2 SP1 database that has support for 15,000 partitions enabled to an instance of SQL Server 2008 R2 RTM. To migrate a SQL Server 2008 SP2 or a SQL Server 2008 R2 SP1 database to their respective previous service packs or RTM releases, you must disable support for 15,000 partitions on the database. However, if support for 15,000 partitions has not been enabled on the database in the instance of SQL Server 2008 SP2 or SQL Server 2008 R2 SP1, there are no restrictions on migrating them to their respective previous service packs or RTM releases.

# Impact on Log Shipping

Log shipping is based on log backup/restore, so it behaves similarly to backup/recovery. If support for 15,000 partitions is not enabled, there are no restrictions. However, if support for 15,000 partitions is enabled, consider the following scenarios.

|  |  |  |
| --- | --- | --- |
| **Instance on Primary** | **Instance on Secondary** | **Impact** |
| SQL Server 2008 SP2 | SQL Server 2008 SP2 | These scenarios are supported. You can enable support for 15,000 partitions on the primary. This is a logged operation. When the transaction log backup is restored on the secondary, support for 15,000 partitions is automatically enabled. |
| SQL Server 2008 R2 SP1 | SQL Server 2008 R2 SP1 |
| SQL Server 2008 SP2 | SQL Server 2008 RTM | These scenarios are only supported if the database on the primary does not have support for 15,000 partitions enabled. Restoring log records or physical backup on the secondary will fail if support for 15,000 partitions is enabled on the primary.  |
| SQL Server 2008 SP2 | SQL Server 2008 SP1 |
| SQL Server 2008 R2 SP1 | SQL Server 2008 R2 RTM |

# Impact on Database Mirroring

If support for 15,000 partitions is not required, no changes are required in the mirroring configuration. However, to enable or disable support for 15,000 partitions on a database in the mirroring environment, you must follow the steps described here.

## Enabling Support for 15,000 Partitions on a Database in a Mirrored Environment

1. Upgrade both the primary and mirrored instance to SQL Server 2008 SP2 or upgrade both the primary and mirrored instance to SQL Server 2008 R2 SP1.
2. Remove database mirroring.
3. Enable support for 15,000 partitions on the primary database.
4. Re-establish database mirroring.

## Disabling Support for 15,000 Partitions on a Database in a Mirrored Environment

1. Remove database mirroring.
2. Follow the steps described in the “Disabling Support for 15,000 Partitions” section.
3. Re-establish database mirroring.

For more information about database mirroring and things to consider in the mirroring configuration, see SQL Server 2008 and SQL Server 2008 R2 Books Online.

# Impact on Replication

If support for 15,000 partitions has not been enabled on a database, there are no changes in replication behavior. If support for 15,000 partitions has been enabled on a database, snapshot replication, transactional replication, merge replication, change data capture, and change tracking are not supported.

# Impact on Manageability

It is recommended that you manually generate scripts for creating and managing partition functions, tables, and indexes with more than 1,000 partitions. You can run the scripts in Query Editor in SQL Server Management Studio. However, SQL Server Management Studio itself does not support the creation and management of partition functions, tables, and indexes with more than 1,000 partitions.

# Impact on Performance

The new, higher limit on the number of partitions supported also affects memory, partitioned indexes, DBCC commands, and queries. This section describes the changes and provides workarounds as needed.

## Memory

We recommend that you use at least 16 GB of RAM if a large number of partitions are in use. If the system does not have enough memory, DML and index creation and rebuild statements can fail due to insufficient memory.

With many memory-intensive processes running on the system with 16 GB of RAM, it is possible that the DML and index creation and rebuild statements that run on a large number of partitions may still run out of memory. Therefore, the larger the amount of memory, the less likely you are to encounter performance and memory issues.

## Partitioned Indexes

As described in [Special Guidelines for Partitioned Indexes](http://msdn.microsoft.com/en-us/library/ms187526.aspx) (http://msdn.microsoft.com/en-us/library/ms187526.aspx), memory limitations can affect the performance or ability of SQL Server to build a partitioned index. This is especially the case with nonaligned indexes.

Creating and rebuilding nonaligned indexes on a table with more than 1,000 partitions is not supported. The functionality is not specifically disabled but you may run into performance and memory issues when you create and rebuild nonaligned indexes on a table with more than 1,000 partitions.

Creating and rebuilding aligned indexes could take longer to execute as the number of partitions increases. We also recommend that you not run multiple create and rebuild index commands at the same time as you may run into performance and memory issues. If you encounter performance and memory issues, you can try the following workaround.

### Workaround

Rebuild a single partition of the partitioned index at a time. It is important to note that rebuilding of a partitioned index cannot be performed online. For more information about rebuilding indexes, see SQL Server 2008 and SQL Server 2008 R2 Books Online.

## DBCC

With a larger number of partitions, DBCC commands could take longer to execute as the number of partitions increases.

## Queries

With a larger number of partitions, queries could take longer to execute as the number of partitions increases.

# Conclusion

Support for 15,000 partitions is targeted to enterprise customers and ISVs with large-scale decision support or data warehouse requirements. As explained in this white paper, there are implications on certain features, including performance, when this support is enabled or disabled. We recommend that you evaluate the implications and test the impact of enabling support for 15,000 partitions in your test lab before you deploy the larger number of partitions to a production environment.

**For more information:**

[SQL Server Website](http://www.microsoft.com/sqlserver) (http://www.microsoft.com/sqlserver)

[SQL Server TechCenter](http://technet.microsoft.com/en-us/sqlserver) (http://technet.microsoft.com/en-us/sqlserver)

[SQL Server DevCenter](http://msdn.microsoft.com/en-us/sqlserver) (http://msdn.microsoft.com/en-us/sqlserver)

Did this paper help you? Please give us your feedback. Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

* Are you rating it high due to having good examples, excellent screen shots, clear writing, or another reason?
* Are you rating it low due to poor examples, fuzzy screen shots, or unclear writing?

This feedback will help us improve the quality of white papers we release.

Send feedback.