

Database Scoped Configuration Overview of functionality

Uwe Ricken

Uwe Ricken

db Berater GmbH

db Berater GmbH
Planung – Installation – Optimierung

<http://www.db-berater.de>
info@db-berater.de

I am working with IT-systems since early 1990's and with the main focus on **Microsoft SQL Server** since version 6.0. I started with development of database applications in 1998 with a professional CRM-System based on Microsoft products (Microsoft Office and **Microsoft SQL Server**).

Since 2008 I'm focused exclusively on **Microsoft SQL Server** and since 2008 I'm working in 3rd level support teams for banks, insurances and global industries.

Since May 2013 I'm a **Microsoft Certified Master: SQL Server 2008** which was an amazing way into the depth of **Microsoft SQL Server**.

In July 2013 I have been awarded with the MVP Award for **Microsoft SQL Server**.

www: <http://www.db-berater.de>
email: uwe.ricken@db-berater.de
blog: <http://www.sqlmaster.de>
twitter: [@dbberater](https://twitter.com/dbberater)
xing: http://www.xing.com/profile/Uwe_Ricken



Database Scoped Configuration

- What is Database Scoped Configuration?
- What Configuration Items are available?
 - Server Settings
 - Columnstore configuration items
 - InMemory configuration items
 - Intelligent Query Processing

What is Database Scoped Configuration?

- Before Microsoft SQL Server 2016 configuration options could only applied on Server Level!
- One setting for all databases made it – partially – impossible to host databases of multiple applications on one host
 - e.g. Sharepoint requires a MAXDOP = 1
- With the possibility of the configuration of “Database Scoped” configuration items we can implement more granular database environments
- Database Scoped Configuration can be configured different for an **Always On Availability Group**

Database Scoped Configuration



- **Max Degree of Parallelism**
- **Parameter Sniffing**
- **Optimize for AdHoc Workloads**
- **Identity Cache**
- **Cardinal Estimation**



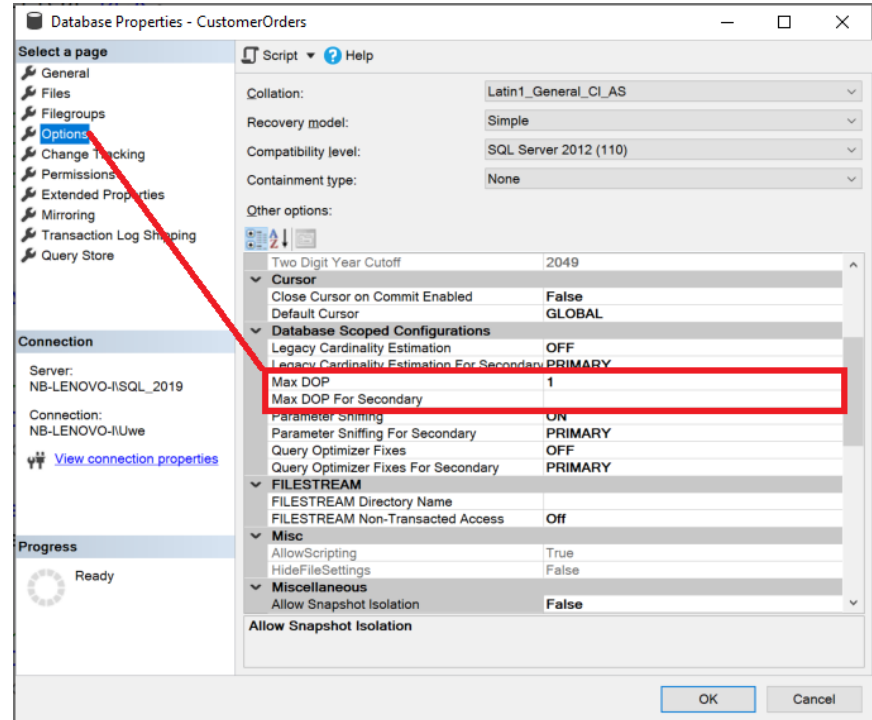
General configuration items

- Options which had to be configured on server level only
 - [Max Degree of Parallelism](#)¹
 - [Parameter Sniffing](#)²
 - [Optimize for AdHoc Workloads](#)
 - [Identity Cache](#)
 - [Cardinal Estimation](#)

1. <https://www.db-berater.de/2020/03/database-scoped-configuration-max-degree-of-parallelism/>
2. <https://www.db-berater.de/2020/03/database-scoped-configuration-parameter-sniffing/>

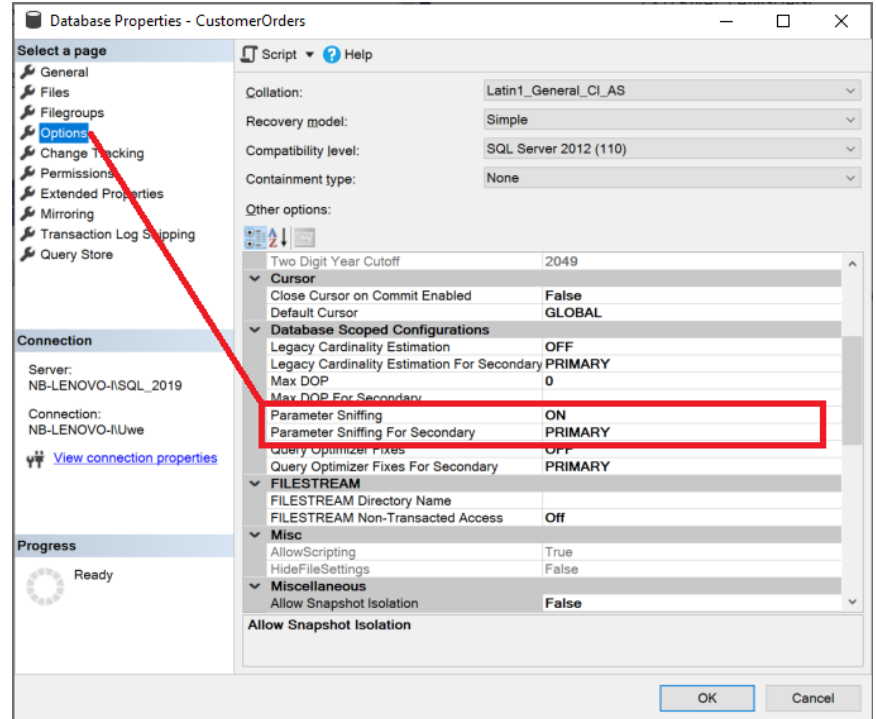
MAXDOP

- Defines the (maximum) number of cores for a query when it runs in parallel
- Command
`ALTER DATABASE SCOPED CONFIGURATION SET MAXDOP = 1;`
- Will be overwritten by query hint
- **Will NOT be overwritten of more restrictive server configuration setting!**
- **Forces a new execution plan!**



PARAMETER_SNIFFING

- Common Problem when table has skewed data
- Can be solved by
 - Enable TF 4136
 - Server
 - Session
 - Query Hints
 - OPTIMIZE FOR UNKNOWN
 - QUERYTRACEON 4136
 - Using @Parameters inside the proc



OPTIMIZE_FOR_AD_HOC_WORKLOADS

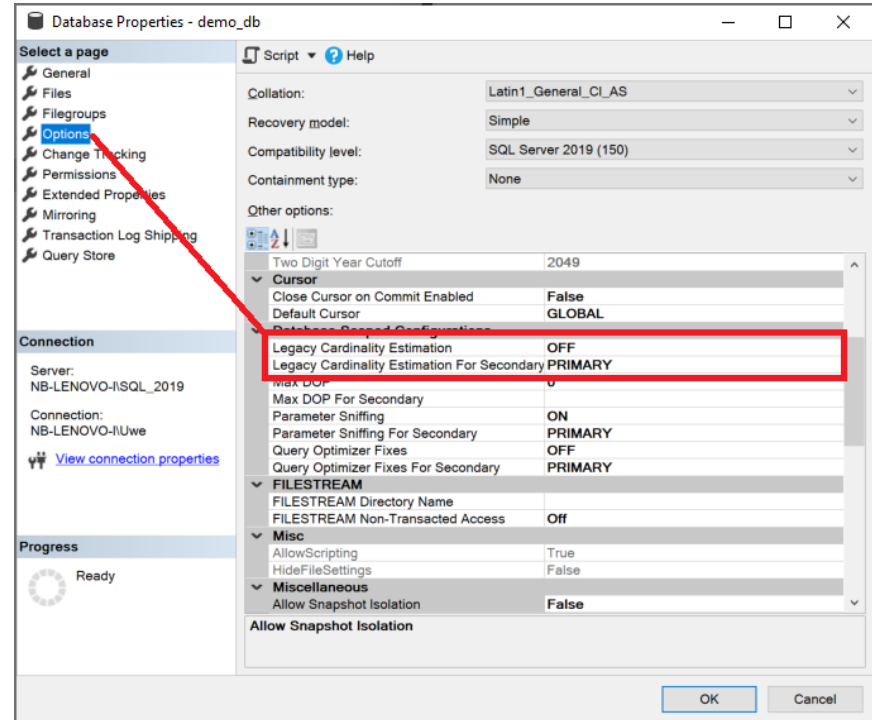
- Enables or disables a compiled plan stub to be stored in cache when a batch is compiled for the first time.
- The default is OFF.
- Once the database scoped configuration OPTIMIZE_FOR_AD_HOC_WORKLOADS is enabled for a database, a compiled plan stub will be stored in cache when a batch is compiled for the first time.
- Plan stubs have a smaller memory footprint compared to the size of the full compiled plan.
- **If a batch is compiled or executed again, the compiled plan stub will be removed and replaced with a full compiled plan.**
- **Changing this option will flush the plan cache for the database!**

IDENTITY_CACHE (>= SQL Server 2017)

- Enables or disables identity cache at the database level.
- The default is ON.
- Identity caching is used to improve INSERT performance on tables with identity columns.
- To avoid gaps in the values of an identity column in cases where the server restarts unexpectedly or fails over to a secondary server, disable the IDENTITY_CACHE option.
- This option **is similar to** the existing **Trace Flag 272**, except that it can be set at the database level rather than only at the server level.

LEGACY_CARDINALITY_ESTIMATION

- Set the query optimizer cardinality estimation model to the SQL Server 2012 and earlier version independent of the compatibility level of the database.
- The default is OFF, which sets the query optimizer cardinality estimation model based on the compatibility level of the database.
- Setting LEGACY_CARDINALITY_ESTIMATION to ON is equivalent to **Trace Flag 9481**.



Cardinal Estimation

- Queries with predicates that use comparison operators between different columns of the same table.
- Queries with predicates that use operators, and any one of the following are true:
 - There are no statistics on the columns involved on either side of the operators.
 - The distribution of values in the statistics is not uniform, but the query seeks a highly selective value set. This situation can be especially true if the operator is anything other than the equality (=) operator.
- The predicate uses the not equal to (!=) comparison operator or the NOT logical operator.
- Queries that use any of the SQL Server built-in functions or a scalar-valued, user-defined function whose argument is not a constant value.
- Queries that involve joining columns through arithmetic or string concatenation operators.
- Queries that compare variables whose values are not known when the query is compiled and optimized.

Any Questions?

Thank you