

Quick Scan Report – Optimize for Ad Hoc Workloads

The Optimize for Ad Hoc Workloads server configuration can improve performance, and is extremely unlikely to negatively impact performance.

This was a new feature that was introduced in SQL Server 2008, and as with many new features in SQL Server, it is not turned on by default.

What Optimize for Ad Hoc Workloads Changes.

Without the Optimize for Ad Hoc Workloads feature enabled, when a plan is compiled, it is kept around in memory (cached) until something else pushes it out of the cache.

With the Optimize for Ad Hoc Workloads feature enabled, the first time a plan is compiled, it is not kept around in cache, rather a stub is kept around to indicate that they query has previously been compiled. The second time the plan is compiled it is kept around in memory. Over time you end up with fewer one time use queries in memory, reducing the memory impact of the plan cache. All other multi-use ad hoc plans will need to be compiled twice on the first and second time that they are used.

What it doesn't to

The Optimize for Ad Hoc Workloads server configuration doesn't change the size of the plan cache in SQL Server, it does however prevent the single use ad hoc plans from using up plan cache space, making more space for other cached plans.

TSQL

To view the current setting.

```
sp_configure 'show advanced options';
GO
sp_configure 'optimize for ad hoc workloads';
```

To turn on Optimize for Ad Hoc Workloads.

```
sp_configure 'optimize for ad hoc workloads',1;
GO
reconfigure;
GO
```

Or you can just right click on the warning in the Database Health Monitor Quick Scan Report and turn it on.

Turning it off.

```
sp_configure 'optimize for ad hoc workloads',0;
GO
reconfigure;
GO
```

To examine the plan cache details.

```
--To see all object in the plan cache cache
SELECT [objtype],
       [usecounts],
       [cacheobjtype],
       [size_in_bytes],
       [text]
FROM sys.dm_exec_cached_plans
CROSS APPLY sys.dm_exec_sql_text(plan_handle)
WHERE [usecounts] = 1
      and LOWER(objtype) = 'ad hoc';
```

To see the single use ad hoc objects in the plan cache

```
SELECT [objtype],
       [usecounts],
       [cacheobjtype],
       [size_in_bytes],
       [text]
FROM sys.dm_exec_cached_plans
CROSS APPLY sys.dm_exec_sql_text(plan_handle)
WHERE [usecounts] = 1
      AND LOWER(objtype) = 'ad hoc';
```

To compare the total space used between single use and multi-use ad hoc.

```
SELECT SUM(CASE WHEN [usecounts] = 1 THEN [size_in_bytes]
                ELSE 0 END) / 1024.0 / 1024.0 as singleUseAdHocMB,
       SUM(CASE WHEN [usecounts] = 1 THEN 0
                ELSE [size_in_bytes] END) / 1024.0 / 1024.0 as multiUseAdHocMB
FROM sys.dm_exec_cached_plans
CROSS APPLY sys.dm_exec_sql_text(plan_handle)
WHERE LOWER(objtype) = 'ad hoc';
```

Should I turn this on?

Run the following query and look at the results.

```
WITH adHocCTE AS
(
  SELECT SUM(CASE WHEN [usecounts] = 1 THEN [size_in_bytes]
                ELSE 0
                END) / 1024.0 / 1024.0 as singleUseAdHocMB,
         SUM(CASE WHEN [usecounts] = 1 THEN 0
```

```
ELSE [size_in_bytes]
END) / 1024.0 / 1024.0 as multiUseAdHocMB
FROM sys.dm_exec_cached_plans
CROSS APPLY sys.dm_exec_sql_text(plan_handle)
WHERE LOWER(objtype) = 'ad hoc'
)
SELECT singleUseAdHocMB, multiUseAdHocMB,
       singleUseAdHocMB * 100.00 /
       (singleUseAdHocMB + multiUseAdHocMB) AS adHocPercent
FROM adHocCTE;
```

If the adHocPercentage is greater than 20% or the singleUseAdHocMB is > 100 then it is probably worth turning this option on, however it really depends on your specific environment.

Reasons to not use Optimize for Ad Hoc Workloads.

Tools such as Database Health, or others that query the plan cache to see what queries have been run. If a majority of your ad hoc queries are always run exactly twice, those queries may be slightly slower on their second run due to the recompile of the plan.