

Apply SQL Server Data Compression

Last Modified on 05 August 2021

⚠ Important: If you have the [SQL Sentry Scalability Pack](#) implemented in your SQL Sentry [database](#), most of these tables are already using partitioned columnstore compression. Do not apply the data compression scripts to those tables.

You may, however, apply the data compression scripts for the tables under the "Event and Other Tables" tab (EventSourceHistory, EventSourceHistoryDetail, PerformanceAnalysisTraceData, PerformanceAnalysisPlanOpTotals, and PerformanceAnalysisTraceQueryStats).

See the [Installation Recommendations](#) article for more information about the SQL Sentry Scalability Pack.

SQL Server data compression is used to save space on disk storage, as well as in memory. The CPU overhead is negligible in certain cases, and we recommend applying the following compression scripts as the CPU overhead has been analyzed for the SQL Sentry application. It's also recommended that you do the following when applying data compression:

- Decide if you'll apply compression all at once or over a couple of days.
- Consider running `sp_spaceused` before and after applying compression to see the impact in your database.
- Use `sp_estimate_data_compression_savings`, SSMS reports, or [SQL Sentry Performance Advisor](#) to know the sizes of these indexes in your database.

`Sort_in_tempdb` may be **On** or **Off**, but **On** is the recommended setting. Indexes may be compressed online or offline, while offline locks the table. Set this to the option that best suits your environment.

📌 Note: Consider applying compression to the smaller objects before the larger ones. This is recommended to reduce the need for additional disk space during the index changes.

Remember that the indexes will be decompressed during index rebuilds, so you should plan to have sufficient disk space for logs and tempdb to account for the full table size.

For more information, see the [Applying Data Compression to the SQL Sentry Database Part 2](#) and [Part 5](#) articles.

Compression Scripts

🔗 The scripts below are also available via the [Apply Data Compression to SentryOne Database.sql](#) file in the [sentryone/sentryone-sql-scripts](#) repository on GitHub.

⚠ Warning: Do not apply compression to these tables if you have the SQL Sentry [Scalability Pack](#) implemented in your SQL Sentry [database](#), because they are already using columnstore compression.

```
-- PerformanceAnalysisDataRollup2

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisDataRollup2',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisDataRollup
ON dbo.PerformanceAnalysisDataRollup2
REBUILD PARTITION = ALL
WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);
```

```
-- PerformanceAnalysisDataRollup4

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisDataRollup4',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisDataRollup ON dbo.PerformanceAnalysisDataRollup4
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);
```

```
-- PerformanceAnalysisDataRollup6

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisDataRollup6',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisDataRollup ON dbo.PerformanceAnalysisDataRollup6
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);
```

```
-- PerformanceAnalysisDataRollup8

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisDataRollup8',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisDataRollup ON dbo.PerformanceAnalysisDataRollup8
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);
```

```
-- PerformanceAnalysisDataRollup11

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisDataRollup11',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisDataRollup ON dbo.PerformanceAnalysisDataRollup11
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);
```

```
-- PerformanceAnalysisDataRollup12

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisDataRollup12',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisDataRollup ON dbo.PerformanceAnalysisDataRollup12
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);
```

```
-- PerformanceAnalysisDataRollup13

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisDataRollup13',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisDataRollup ON dbo.PerformanceAnalysisDataRollup13
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);
```

```

-- PerformanceAnalysisDataRollup14

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
  @schema_name = N'dbo',
  @object_name = N'PerformanceAnalysisDataRollup14',
  @index_id = NULL,
  @partition_number = NULL,
  @data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisDataRollup ON dbo.PerformanceAnalysisDataRollup14
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

```

-- PerformanceAnalysisData

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
  @schema_name = N'dbo',
  @object_name = N'PerformanceAnalysisData',
  @index_id = NULL,
  @partition_number = NULL,
  @data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisData_Wide ON dbo.PerformanceAnalysisData
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

```

-- PerformanceAnalysisDataDatabaseCounter

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
  @schema_name = N'dbo',
  @object_name = N'PerformanceAnalysisDataDatabaseCounter',
  @index_id = NULL,
  @partition_number = NULL,
  @data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisData_Wide ON dbo.PerformanceAnalysisDataDatabaseCounter
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

```

-- PerformanceAnalysisDataDiskCounter

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
  @schema_name = N'dbo',
  @object_name = N'PerformanceAnalysisDataDiskCounter',
  @index_id = NULL,
  @partition_number = NULL,
  @data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisData_Wide ON dbo.PerformanceAnalysisDataDiskCounter
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

```

-- PerformanceAnalysisDataSQLDBCounter

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
  @schema_name = N'dbo',
  @object_name = N'PerformanceAnalysisDataSQLDBCounter',
  @index_id = NULL,
  @partition_number = NULL,
  @data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisData_Wide ON dbo.PerformanceAnalysisDataSQLDBCounter
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

```

-- PerformanceAnalysisDataTableAndIndexCounter

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
  @schema_name = N'dbo',
  @object_name = N'PerformanceAnalysisDataTableAndIndexCounter',
  @index_id = NULL,
  @partition_number = NULL,
  @data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisData_Wide ON dbo.PerformanceAnalysisDataTableAndIndexCounter
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

```

-- PerformanceAnalysisDataTintriCounter

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
  @schema_name = N'dbo',
  @object_name = N'PerformanceAnalysisDataTintriCounter',
  @index_id = NULL,
  @partition_number = NULL,
  @data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisData_Wide ON dbo.PerformanceAnalysisDataTintriCounter
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

```

-- PerformanceAnalysisDataVMCounter

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
  @schema_name = N'dbo',
  @object_name = N'PerformanceAnalysisDataVMCounter',
  @index_id = NULL,
  @partition_number = NULL,
  @data_compression = N'PAGE';

-- alter index to use page compression
ALTER INDEX IX_PerformanceAnalysisData_Wide ON dbo.PerformanceAnalysisDataVMCounter
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

Event and Other Tables

Note: You may apply compression to the following tables with or without the [SentryOne Scalability Pack](#).

```
-- EventSourceHistory

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'EventSourceHistory',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'PAGE';

SELECT index_id, name FROM sys.indexes WHERE object_id = OBJECT_ID('dbo.EventSourceHistory');

-- alter indexes to use page compression
ALTER INDEX IX_MaxIDs ON dbo.EventSourceHistory
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_IncompleteRecs ON dbo.EventSourceHistory
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_Unique1 ON dbo.EventSourceHistory
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_Unique2 ON dbo.EventSourceHistory
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_FailedObjectsInRange ON dbo.EventSourceHistory
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_GlobalViews ON dbo.EventSourceHistory
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_DetailInserts ON dbo.EventSourceHistory
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

--Consider using ROW compression on the clustered index if index maintenance duration is a concern.
ALTER INDEX PK_EventHistory ON dbo.EventSourceHistory
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);
```

```

-- EventSourceHistoryDetail

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'EventSourceHistoryDetail',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'ROW';

SELECT index_id, name FROM sys.indexes WHERE object_id = OBJECT_ID('dbo.EventSourceHistoryDetail');

-- alter indexes to use row compression
ALTER INDEX PK_EventHistoryDetail ON dbo.EventSourceHistoryDetail
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = ROW);

ALTER INDEX IX_EventSourceHistoryID ON dbo.EventSourceHistoryDetail
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = ROW);

-- alter indexes to use page compression
ALTER INDEX IX_Unique2 ON dbo.EventSourceHistoryDetail
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_Unique1 ON dbo.EventSourceHistoryDetail
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_MasterDetailCorrelationTrigger ON dbo.EventSourceHistoryDetail
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_IncompleteRecs ON dbo.EventSourceHistoryDetail
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_MaxIDs ON dbo.EventSourceHistoryDetail
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```

```

-- PerformanceAnalysisTraceData

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisTraceData',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'ROW';

SELECT index_id, name FROM sys.indexes WHERE object_id = OBJECT_ID('dbo.PerformanceAnalysisTraceData');

-- alter indexes to use page compression
ALTER INDEX IX_MaxIDs ON dbo.PerformanceAnalysisTraceData
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_PerformanceAnalysisTraceData_Wide ON dbo.PerformanceAnalysisTraceData
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

-- alter index to use row compression
ALTER INDEX PK_PerformanceAnalysisTraceData ON dbo.PerformanceAnalysisTraceData
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = ROW);

ALTER INDEX IX_PurgeProcess ON dbo.PerformanceAnalysisTraceData
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = ROW);

```

```

-- dbo.PerformanceAnalysisPlanOpTotals

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisPlanOpTotals',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'ROW';

SELECT index_id, name FROM sys.indexes WHERE object_id = OBJECT_ID('dbo.PerformanceAnalysisPlanOpTotals');

-- alter indexes to use row compression
ALTER INDEX PK_PerformanceAnalysisTracePlanOpTotals ON dbo.PerformanceAnalysisPlanOpTotals
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = ROW);

ALTER INDEX IX_PerformanceAnalysisTracePlanOpTotals_Unique ON dbo.PerformanceAnalysisPlanOpTotals
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = ROW);

```

```

-- dbo.PerformanceAnalysisTraceQueryStats

-- estimate compression savings
EXEC sys.sp_estimate_data_compression_savings
@schema_name = N'dbo',
@object_name = N'PerformanceAnalysisTraceQueryStats',
@index_id = NULL,
@partition_number = NULL,
@data_compression = N'ROW';

SELECT index_id, name FROM sys.indexes WHERE object_id = OBJECT_ID('dbo.PerformanceAnalysisTraceQueryStats');

-- alter indexes to use row compression
ALTER INDEX PK_PerformanceAnalysisTraceQueryStats ON dbo.PerformanceAnalysisTraceQueryStats
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = ROW);

ALTER INDEX IX_PurgeProcess ON dbo.PerformanceAnalysisTraceQueryStats
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = ROW);

-- alter indexes to use page compression
ALTER INDEX IX_PerformanceAnalysisTraceQueryStats_ObjectLookup ON dbo.PerformanceAnalysisTraceQueryStats
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

ALTER INDEX IX_PerformanceAnalysisTraceQueryStats_Unique ON dbo.PerformanceAnalysisTraceQueryStats
REBUILD PARTITION = ALL WITH (SORT_IN_TEMPDB = ON, DATA_COMPRESSION = PAGE);

```