# Configuring SQL Server Backup Based on Weekly Cycle

By GZ

# Weekly SQL Server Backup Process Description

Weekly backup process is based on Full database backup and following set of differential backups. In parallel to these 2 backup types, T-Log backup must be run to be able to truncate the database log and be able to restore database up to point of failure (or specified time). T-Log backup will not be discussed in this document. Its configuration remains the same no matter how often you perform Full database backup.

Weekly process usually recommended for VLDB with size above 1TB where time for the full backup takes several hours. Full and differential backups are very resource intensive processes and running backup for long-time every day will significantly impact database performance during this time.

Usually, full backup is scheduled to run on Saturday (or Sunday), and differential backup is running daily at other days.

# Implementation of Customized Full and Differential Backups based on Weekly Cycle

Solution is implemented as single stored procedure [dbo].[DBA\_DBBackup\_WithExceptions\_Incl] deployed in Admin database[[1]](#footnote-1). Stored procedure has the following parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 1 | @path | VARCHAR(max) | NULL |

**Purpose.** It stores path for backup location (for on premise server) OR name of Storage Account for Azure Servers

**Examples.** @path = 'H:\SQL1\_Backup\DBBkup\Full\' ( for on-premises server)

@path = 'gxxbackns1sto01' ( name of storage account)

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 2 | @includeDB | VARCHAR(max) | NULL |

**Purpose.** It stores comma-separated list of DBs to perform Full or Differential backup. If parameter is NULL, all databases are backed up.

If Admin database has table dbo.tbl\_BackupExceptions,[[2]](#footnote-2)

Text, application

Description automatically generated

additional tables can be excluded from the list if there are rows with ExcludeFromFull= 1

**Examples.** @includeDB = 'TestDB1,TestDB2,TestDB3' - backup performed for only 3 listed databases

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 3 | @excludeDB | VARCHAR(max) | NULL |

**Purpose.** It stores comma-separated list of DBs to be excluded from Full or Differential backup

**Examples.** @excludeDB = 'TestDB1,TestDB2,TestDB3' - backup performed for all databases on the server, excluding 3 listed databases

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 4 | @BackupType | char(1) | 'F' |

**Purpose.** It stores values for backup types. Only values F (Full) or D (Differential) are allowed

**Examples.** @BackupType = 'D' - perform differential backup

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 5 | @CopyOnly | char(1) | 'Y' |

**Purpose.** It indicates whether copy-only or normal backup is requested. Only values Y (Yes) or N (No) are allowed

**Examples.** @CopyOnly = 'N' - perform normal backup

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 6 | @RunOnPrimary | char(1) | 'N' |

**Purpose.** It indicates whether backup will be run on Primary or Secondary replica. Only values Y (Yes) or N (No) are allowed

**Examples.** @RunOnPrimary = 'Y' - run backup on Primary replica.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 7 | @PrintOnly | char(1) | 'N' |

**Purpose.** It indicates whether backup will be run, or backup statement only will be generated. Only values Y (Yes) or N (No) are allowed

**Examples.** @PrintOnly = 'Y' - create backup statement without executing an actual command. It is helpful when you need to know how many backup files will be used when backup process will run next time.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 8 | @AlternativePath | varchar(max) | NULL |

**Purpose.** It indicates where backup will be stored if backup to URL cannot be performed. It assumes local folder on the server. If Backup to URL cannot be performed due to size of database, and parameter is not specified, procedure will fail.

**Examples.** @AlternativePath = 'P:\Backup\Full\' - create backup in specified folder if compressed database size is greater than backup to URL can handle.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 9 | @ChangeBackupType | char(1) | 'N' |

**Purpose.** It indicates whether it is allowed to change the backup type if a differential backup cannot be performed. Only values Y (Yes), Q (quit), N (No) are allowed. Parameter is applicable only if Differential backup is chosen and only for AG-based databases.

**Examples.** @ChangeBackupType = 'Y' - differential backup is requested, but if it is not possible to run it, full backup will be performed.

@ChangeBackupType = 'Q' - differential backup is requested, and if it is not possible to run it, quit the procedure with an error.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 10 | @MissingBkpDays | Int | 8 |

**Purpose.** It indicates min number of days since last Differential or Full backups when switching backup type is allowed. It cannot be less than 2.

**Examples.** @MissingBkpDays = 9 - switch to Full backup is allowed if differential backup is requested, but it was > 9 days since last Full or Differential backup on current replica.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 11 | @BufferCount | int | NULL |

**Purpose.** It specifies the total number of I/O buffers to be used for the backup operation. You can specify any positive integer; however, large numbers of buffers might cause "out of memory" errors because of inadequate virtual address space in the Sqlservr.exe process. We allow this parameter to set only for server with > 48 cores.

In each case, testing must be performed to set not default value, used by SQL Server. Default value used by SQL Server depends on number of cores, data drives and other external configurations.

**Examples.** @BufferCount = 400 - in M128 servers where BXLDXK is running, we saw optimal performance with value 400.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 12 | @NumberOfFiles | int | NULL |

**Purpose.** It indicates number of files (stripes) in backup. It can be between 1 and 64.

If parameter is not specified, number of files is set internally by SQL Server or based on value of parameter @CompressionPct below.

**Examples.** @NumberOfFiles = 64 - run backup with 64 stripes.

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Datatype** | **Default Value** |
| 13 | @CompressionPct | decimal(6,2) | NULL |

**Purpose.** It indicates backup compression ratio. If parameter is not specified, compression ratio is set to 0.2 (20%).

Compression ratio is calculated as total size of backup files divided by total used (not allocated) space across all data files and filegroups.

**Examples.** @CompressionPct = 0.26 - run backup with 64 stripes.

# Configuring SQL Server Backup based on Weekly Cycle (Full and Differential Backups) at CONTOSO

To switch from existing daily Full backup process to Weekly Full + Differential backups cycle, the following steps needs to be performed:

1. Deploy code with stored procedure [dbo].[DBA\_DBBackup\_WithExceptions\_Incl] in Admin database in every replica.
2. Disable current job in every replica in AG that performs daily backups for all databases. At CONTOSO, this default job is configured through SQL QC and called “DBA DBBackup” (see attached file 02\_Disable\_And\_Rename\_Old\_FullBackup\_Job.sql)
3. In each replica configure daily job using this procedure that runs daily backup for system and small databases that require daily cycle. Exclude VLDB that requires weekly backup cycle using @excludeDB parameter. Use proper name for storage account (see attached file 03\_Create\_New\_Regular\_Full\_Backup\_Daily\_Job.sql).

For example, modify 3 “exposed” parameters to configure proper storage account, exclusion list of databases and backup file storage terms.

Text

Description automatically generated with medium confidence



1. In each replica configure weekly job that perform Full backup for VLDB. Note, if differential backup is assumed, Full backup job must run on Primary replica only and not as Copy-Only. Use @includeDB parameter to list all databases. Use proper name for storage account (see attached file 04\_Create\_Full\_Backup\_Weekly\_Job\_For\_VLDB.sql).

For example, modify 2 “exposed” parameters to configure proper storage account and inclusion list for databases that require Full backup

Text

Description automatically generated



1. In each replica configure weekly job that runs Differential backups for VLDB 6 days a week (except day when Full backup is running). Use @includeDB parameter to list all databases. Use proper name for storage account (see attached file 05\_Create\_Diff\_Backup\_Daily\_Job\_For\_VLDB.sql)

For example, modify 2 “exposed” parameters to configure proper storage account and inclusion list for databases that require Differential backup.

Text

Description automatically generated with medium confidence



1. If replicas in AG exist in 2 regions, repeat steps 3-5 using local storage account in that region.

For example,

Graphical user interface, application

Description automatically generated



# Customizing Settings for Backup Job

## Using @CompressionPct option

After required jobs are configured with the default settings, you can customize job execution to “fine tine” backup execution and reflect specific of each database. The most important option is @CompressionPct. Based on actual compression ratio that SQL Server is using during the backup, our procedure will select optimal number of files for the backup. That would make backup job run faster. Also, incorrect value of backup stripes may cause backup to fail.

For example, current setting for job that run Full backup of Bixxx database below:

EXEC [Admin].[dbo].[DBA\_DBBackup\_WithExceptions\_Incl] @path = 'gxebixxxpd1sxxxsto', @includeDB = 'Bxxx', @CopyOnly = 'N', @RunOnPrimary = 'Y' ,@BufferCount = 400, @CompressionPct = 0.25

## Using @NumberOfFiles option

Another parameter you may use in case of emergency is @NumberOfFiles. If backup job fails in the middle, it means often that insufficient backup stripes were used by the backup process.

For one time, set @NumberOfFiles = 64. It is max possible value. Re-Run backup job For example,

EXEC [Admin].[dbo].[DBA\_DBBackup\_WithExceptions\_Incl] @path = 'gxxxxqbxxxto', @includeDB = 'Bixxx', @CopyOnly = 'N', @RunOnPrimary = 'Y' ,@BufferCount = 400, @NumberOfFiles = 64

Then re-calculate current compression ratio based on methodology described below and put proper value for @CompressionPct . Remove @NumberOfFiles option to let program choose optimal number of backup stripes based on current compression ratio.

## Using @BufferCount option

For servers with 128 cores we discovered that 400 is the optimal number of buffers. Add this parameter if job runs on servers with such number of cores. Use default values for servers with smaller number of cores.

## How to Estimate Current Compression Ratio

1. Get current usable size of database by running standard Disk Usage report in SSMS

A screenshot of a computer

Description automatically generated



In this example, database is using 42,845 GB.

1. Get Number of backup files from the last successful Full backup in storage account and their average size. You can do that by looking in Azure portal

For example, last backup of database generated 61 files (it starts with \_0) with average size of 164GB

1. Calculate total space for all backup files.

For example, in this case it will be 61\* 164 = 10,004

1. To get current compression ratio, divide Total Backup Space by Total Usable Space.

For example, in our case compression ratio is 10,004 / 42,845 = 0.23

1. You may add 0,02-0.04 points to the result to cover possible changes (For example, if you may expect more BLOB data coming because client added new table with BLOB fields)

Table

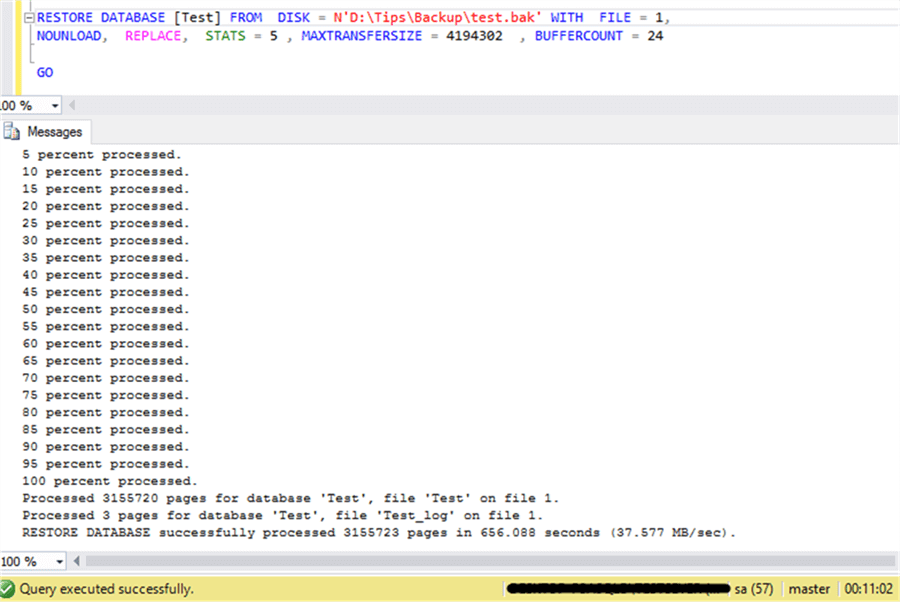
Description automatically generated

Speeding up restore

Restore database [test] from disk = N’D:\Tips\Backup\test.bak’ with file = 1,

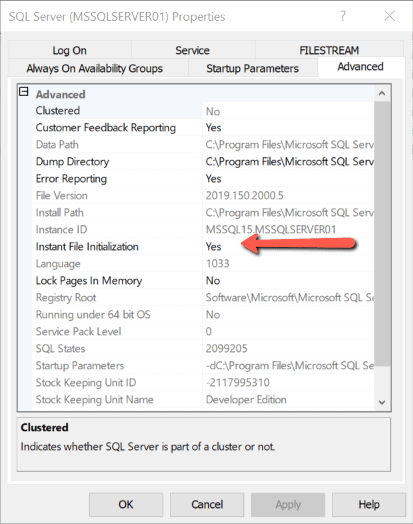
Nounload, replace, stats = 5, maxtransfersize = 4194302, buffercount = 24

[Optimize SQL Server Database Restore Performance (mssqltips.com)](https://www.mssqltips.com/sqlservertip/4935/optimize-sql-server-database-restore-performance/)



To even get faster backups and restores, you could also look at these things:

* Parallel backup operations
* Backup device performance
* Instant file initialization
* Data compression
* Backup compression



[Optimizing SQL Server Backup and Restore – SQLServerCentral](https://www.sqlservercentral.com/blogs/optimizing-sql-server-backup-and-restore)

1. GKO standard database created in each SQL Server to store objects associated with DBA activity [↑](#footnote-ref-1)
2. POLDUK servers have this table to configure database backup [↑](#footnote-ref-2)