# BCS SQL Server installation standards

Location:

N:\Technical Environment\SQLServer\BCS\_SQL\_Installation\_2024

1. **Prerequisite**

Client to provide service accounts for the SQL Server Services using their naming standard (preferably individual account for individual SQL Server services, however, this is not necessary and is dependent on the level of security of the client).

While this is not essential (the SQL Server services can be set to run as their default local service or local user account if needed), for best practice security configuration the Database Engine and SQL Server Agent should run under a non-expiring Active Directory (AD) service accounts and use delegation.

The AD SQL Engine service account must have the:

* Account is trusted for delegation,
* be registered with a Service Principal Name (SPN) on Active Directory to have Kerberos support enabled,
* Instant file initialisation (IFI) and how to enable it to speed up growth for your SQL Server database files,
* Lock pages in memory.

and the server running the SQL Server instance must have “Computer is trusted for delegation” on Active Directory. The client and server computers must be part of the same Windows domain or in a trusted domain. Note that the “Account is sensitive and cannot be delegated” check box must not be selected for the user requesting delegation.

Correct SQL Server installation media (with all additional SPs and CUs) is loaded and made available by the client on the server.

Client has provided the necessary permissions on the server for the SQL Server installation (Local Administrator is required on the server).

The system administrator will verify/confirm that the server cluster sizes disks/LUNs are configured to 64 KB.

If more than 10 instances should be installed, please notify the client to change the limit:

<https://learn.microsoft.com/en-us/troubleshoot/windows-server/identity/default-workstation-numbers-join-domain>

The vendor and/or developer will provide details regarding the optimal auto-growth setting and initial database settings.

**Useful links:**

* **Enable computer and user accounts to be trusted for delegation**

<https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/enable-computer-and-user-accounts-to-be-trusted-for-delegation>

* **Register a Service Principal Name for Kerberos connections**

<https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/register-a-service-principal-name-for-kerberos-connections?view=sql-server-ver16>

* **Lock pages in memory**

<https://learn.microsoft.com/en-us/windows/security/threat-protection/security-policy-settings/lock-pages-in-memory>

* **Database instant file initialization**

<https://learn.microsoft.com/en-us/sql/relational-databases/databases/database-instant-file-initialization?view=sql-server-ver16>

* **SQL Server memory configuration options**

[Server memory configuration options - SQL Server | Microsoft Learn](https://learn.microsoft.com/en-us/sql/database-engine/configure-windows/server-memory-server-configuration-options?view=sql-server-ver16)

1. **SQL Server Services Installation**

As per Microsoft recommendations, the bare minimum services will be installed; Database Engine Services; Client Tools Connectivity, Management Tools – Basic/Complete. This is unless otherwise specified by the client and/or application vendor and developer.

Corresponding users/application hosts should be checked and planned to install a compatible version of SQL server drivers and Client Tools Connectivity.

Firewall rules should be checked and corrected if required on both sides – SQL Server and client/application.

1. **Folders Location**

The following are the folder locations for a SQL Server installation where the instance name will be dependent on the named SQL Server instance (the default SQL Server instance will be called MSSQLSERVER).

BCS recommends separating the drive onto the different spindles and LUN to provide the optimal disk layout for the SQL Server instance; preferably, RAID 5 or 10 are used for the Database data file, RAID 10 is used for the Database transaction log file, RAID 10 is for TempDB, and RAID 1 for Database Backup.

The customer may opt for a lesser number of drives, but in general, the preferred configuration is as follows:

**D:\Program Files\MSSQL<versioncode>.<instancename>** contains the SQL Server Binaries

**E:\MSSQL.<instancename>\UserDB\<dbname>** contains User database data file

**E:\MSSQL<versioncode>.<instancename>\MSSQL\Data** contains System databases

**F:\MSSQL.<instancename>\UserLog\<dbname>** contains User database transaction log file

**G:\MSSQL.<instancename>\TempDB** contains tempdb database

**H:\MSSQL.<instancename>\Backup\<dbname>** contains SQL Server backup

1. **Authentication Mode**

SQL Server allows for Windows Only Authentication or Mixed Mode Authentication, which allows both Windows and SQL Server Authentication. BCS preference is to configure both Windows and SQL Server Authentication mode (unless specified otherwise by the client). This will allow a secure SQL Server Authenticated account, sa, to be created adding as a security backup option for DBA support and also allow applications the option of connecting to the Database Engine using a Windows or SQL Server account.

1. **SQL Server Service Account**

In most cases, a service account for the SQL Server Database Engine and one for the SQL Server Agent service will be sufficient (Note: it is possible to use just one service account for both services, if needed). The account that the SQL Server Database Engine service is running under will be configured for “Perform volume maintenance tasks” and “Lock pages in memory”.

Enabling Instant File Initialization and “Perform volume maintenance tasks” will keep SQL Server from “zeroing out” new data file space that has been created or expanded (this doesn’t apply to transaction log file) so it can significantly improve these execution time. However, note that this introduces a security risk. When a database is deleted where Instant File Initialization is enabled, the data isn’t zeroed out, which means the existing data may still exist on the drive and potentially still accessible. Note: it’s up to the client to determine whether this feature should be enabled, particularly on a secure SQL Server installation.

Opinion is varied regarding enabling of LPIM (Lock Pages In memory) on the latest Windows Server versions, given memory management has been improved. Some people believe this is no longer required as less unnecessary paging will occurred on the Database Engine (in particular on 64 bit environments). However, the BCS recommendation is to enable LPIM even on latest Windows Server editions.

1. **SQL Server Instance Static Port**

For a secure SQL Server installation with default SQL Server instance, BCS recommends to use a static port number (besides the common default port 1433 or 1434).

By default, named SQL Server instances use dynamic ports. For named instances, BCS also recommend configuring static ports (this means the SQL Server Browser service can be disabled).

However, if you set up a static port for a default SQL Server instance, the SQL Server instance will not be able to resolve connection string without the static port number. This means any connection string will need to include the static port number, or a TCP alias or a global default port needs to be created on the application server end.

BCS prefer to install a named SQL Server instance wherever possible for a secure SQL Server installation.

Note: if you set up the SQL Server instance to use a static port and that port number is already used by the server for other purposes, the SQL Server instance won’t start.

1. **Multiple TempDB Files**

The number of TempDB data file should be split depending on the number of logical processors (i.e. a server with 4 logical processors should have 4 tempdb data file).

The rule of thumb is that a maximum of 8 tempdb data file should be created for server with more than 8 logical processors and only add additional tempdb data files in multiple of 4 (and up to the number of logical processors) until the contention is reduced, as indicated in the following Microsoft KB article <https://support.microsoft.com/en-us/kb/2154845>.

The initial size for TempDB should be big enough to avoid TempDB growth that leads to file fragmentation. Initial auto-growth settings should be set to 1024MB and adjusted accordingly if this growth causes high VLF (Virtual Log File) counts.

1. **Auto Growth Setting**

The rule of thumb is the data and transaction log file auto-growth setting for the System databases are set to the following:

* Master: 16MB
* Msdb: 32MB
* TempDB: 1024MB
* Model: 256MB

For the User databases, this will depend on the application vendor or developer but if no preference exists, they will be set to 256MB.

Note: these will be monitored and altered if necessary based on the actual activity on the SQL Server instance and changes in the baseline size.

1. **Job History**

The job history should be configured to retain the history jobs for a period of time (rather than for a number of entries). BCS will configure a 4 week retention period for the job history unless otherwise specified by the application vendor, user and/or developer.

1. **Special treatment for SSISDB**

If the SQL server instance intended to be a host SSIS catalogue, additional (re)configurations should be considered:

* SSISDB database by default is located at the same location as system databases. But in difference from system databases, it can be reasonably big. So consider to move this database to the same subfolder where all user databases are hosted,
* For better control of SSISDB review the Retention period, Server-wide default logging level and maximum number of versions per project parameters.

1. **AAG/BAG configuration**

If the SQL server instance is intended to have AAG/BAG, additional (re)configurations should be considered:

* Windows Failover Cluster Feature should be installed on every AAG/BAG node
* Enable Windows Failover Clustering Configuration for SQL Server
* Enable the SQL Server 2016 Always On Availability Groups Feature in SQL Server Configuration Manager

1. **Query store configuration.**

If the SQL server instance is 2016 or higher, the Query Store can be enabled for every user database. For more details please have a look the <https://learn.microsoft.com/en-us/sql/relational-databases/performance/manage-the-query-store?view=sql-server-ver16&tabs=ssms>

1. **SQL Server Error Log**

The SQL Server log is scheduled to recycle daily at 12am by the “Cycle ERRORLOG” SQL Server Agent job. A retention limit setting of 40 will allow for at least one month’s worth of error logs to be retained on the SQL Server instance (taking into account any possible SQL Server instance restarts during the month, which will also recycle the error log).

1. **SQL Server Agent Jobs Owner**

The owner of all SQL Server Agent jobs will be set to sa or the service account that the SQL Server Agent service is running as, unless otherwise specified by the application vendor, user and/or developer for specific SQL Server Agent jobs.

This includes the SQL Server Agent job created by the standard SQL Server maintenance plan.

1. **SQL Server Min/Max Memory**

The minimum memory for the SQL Server instance on a dedicated server should be set to at least 128MB to allow the SQL Server instance to at least start up.

The maximum memory for the SQL Server instance on a dedicated server will be based on the following calculation provided by Microsoft article – less or equal of 75% RAM. ~~The best practice recommendation is to reserve 1GB of RAM for the Operating System plus 1GB for every 4GB up to 16GB, and a further 1GB for every additional 8GB installed above 16GB, with the remaining memory allocated to the SQL Server instance.~~

Note: if the server has other applications running on it besides SQL Server, then this allocation may need to be adjusted. In this case, BCS requires the assistance of the application vendor and/or developer to help determine the recommended maximum memory setting.

The maximum memory configuration will also depend on the number of SQL Server instances on this server and/or if there are any other SQL Server service components installed beyond the bare minimal SQL Server service installation.

Further monitoring (preferably during heavy application usage) of the physical memory usage will be required after the server has been running for a certain period of time to see whether the configuration needs to be adjusted due to memory pressures.

Note: the total maximum memory configuration for all the SQL Server Failover Cluster instances can’t exceed the total physical memory of one node (i.e. there need to be significant physical memory for all the SQL Server Failover Cluster instances to failover to the same node and run without memory pressure).

1. **SQL Server Maintenance Plans**

BCS uses the Ola Hallengren maintenance plan to provide the following maintenance tasks. The installation scripts can be downloaded from <https://ola.hallengren.com>

* Integrity Check – Done daily at 12am, Include Index
* Database Backup – Done daily at 1am (to the backup location specified in section 3 above) with compression (if available) for the full database backup, verify backup file, backup maintenance clean-up task to delete old backup is done after the backup task, minimal retention period of 4 days to take into account long weekends. This is to minimise the time required to complete database restores, i.e. local backup is easier and quicker to restore than organising a retrieval from tape or other external devices.
* Transaction Log Backup – Only for User Database, done daily every 1 hour to backup location specified in (3) with compression (if available), verify backup file, backup maintenance clean-up task to delete old backup is done after the backup task, retention period of 2 days.
* Rebuild Index and Update statistics – Done Saturday at 2am, Tables and Views, Sort results in tempdb

|  |  |
| --- | --- |
| DatabaseBackup - SYSTEM\_DATABASES - FULL | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d master -Q "EXECUTE [dbo].[DatabaseBackup] @Databases = 'SYSTEM\_DATABASES', @Directory = 'H:\MSSQL.MSSQLSERVER\Backup', @BackupType = 'FULL', @Verify = 'Y', @CleanupTime = '96', @CheckSum = 'Y', @LogToTable = 'Y', @Compress = 'Y'" -b |
| DatabaseBackup - USER\_DATABASES - FULL | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d master -Q "EXECUTE [dbo].[DatabaseBackup] @Databases = 'USER\_DATABASES', @Directory = 'H:\MSSQL.MSSQLSERVER\Backup', @BackupType = 'FULL', @Verify = 'Y', @CleanupTime = '96', @CheckSum = 'Y', @LogToTable = 'Y', @Compress = 'Y'" -b |
| DatabaseIntegrityCheck - SYSTEM\_DATABASES | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d master -Q "EXECUTE [dbo].[DatabaseIntegrityCheck] @Databases = 'SYSTEM\_DATABASES', @LogToTable = 'Y'" -b |
| DatabaseIntegrityCheck - USER\_DATABASES | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d master -Q "EXECUTE [dbo].[DatabaseIntegrityCheck] @Databases = 'USER\_DATABASES', @LogToTable = 'Y'" -b |
| IndexOptimize - SYSTEM\_DATABASES | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d master -Q "EXECUTE [dbo].[IndexOptimize] @Databases = 'SYSTEM\_DATABASES', @LogToTable = 'Y', @SortInTempdb = 'Y', @FragmentationMedium = 'INDEX\_REORGANIZE', @FragmentationHigh = 'INDEX\_REBUILD\_OFFLINE', @FragmentationLevel1 = 5, @FragmentationLevel2 = 30, @UpdateStatistics = 'ALL', @OnlyModifiedStatistics = 'Y', @LockTimeout = 1800" -b |
| IndexOptimize - USER\_DATABASES | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d master -Q "EXECUTE [dbo].[IndexOptimize] @Databases = 'USER\_DATABASES', @LogToTable = 'Y', @SortInTempdb = 'Y', @FragmentationMedium = 'INDEX\_REORGANIZE', @FragmentationHigh = 'INDEX\_REBUILD\_OFFLINE', @FragmentationLevel1 = 5, @FragmentationLevel2 = 30, @UpdateStatistics = 'ALL', @OnlyModifiedStatistics = 'Y', @LockTimeout = 1800" -b |
| CommandLog Cleanup | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d master -Q "DELETE FROM [dbo].[CommandLog] WHERE StartTime < DATEADD(dd,-30,GETDATE())" -b |
| Output File Cleanup | cmd /q /c "For /F "tokens=1 delims=" %v In ('ForFiles /P "E:\MSSQL11.MSSQLSERVER\MSSQL\LOG" /m \*\_\*\_\*\_\*.txt /d -30 2^>^&1') do if EXIST "E:\MSSQL11.MSSQLSERVER\MSSQL\LOG"\%v echo del "E:\MSSQL11.MSSQLSERVER\MSSQL\LOG"\%v& del "E:\MSSQL11.MSSQLSERVER\MSSQL\LOG"\%v" |
| sp\_delete\_backuphistory | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d msdb -Q "DECLARE @CleanupDate datetime SET @CleanupDate = DATEADD(dd,-30,GETDATE()) EXECUTE dbo.sp\_delete\_backuphistory @oldest\_date = @CleanupDate" -b |
| sp\_purge\_jobhistory | sqlcmd -E -S $(ESCAPE\_SQUOTE(SRVR)) -d msdb -Q "DECLARE @CleanupDate datetime SET @CleanupDate = DATEADD(dd,-30,GETDATE()) EXECUTE dbo.sp\_purge\_jobhistory @oldest\_date = @CleanupDate" -b |