**MS SQL SERVER**

**Standard Operating Procedures**

**(SOP)**

**Creation/Deletion of Database on SQL server 2008**

**Submitted to**

****

**BY**

****

**CIS, Wipro Limited**

Document Details

|  |  |
| --- | --- |
| Project Name | Innogy SE |
| Account | CIS |
| IT Component/Application Title | Creation/Deletion of Database on SQL server 2008 |
| Current Version | 1.5 |
| List of Contributors | Pradipkumar Kansara, Kiran Bhalekar, Ayushi Gautam, Sivakumar Goud, Nanda Kishore Reddy |
| Customer Contact Information |  |

Version History

(All revisions made to this document must be listed in chronological order. All revisions must be approved. Review and Approval can be done by an internal source or by the customer)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Version | Date of Revision | Description | Author | Reviewed By | Approved By |
| 1.0 | 29-08-2015 | Intial Draft | Kiran Bhalekar | Pradipkumar Kansara | Mahesh Keshatwar |
| 1.1 | **04-09-2015** | **Document Completed** | **Kiran Bhalekar** | **Pradipkumar Kansara** | **Mahesh Keshatwar** |
| 1.2 | **21-08-2016** | **Updated document to standard format** | **Kiran Bhalekar** | **Pradipkumar Kansara** | **Mahesh Keshatwar** |
| 1.3 | **17-05-2017** | **Wipro and Innogy Logo changed** | **Ayushi Gautam** | **Pradipkumar Kansara** | **Mahesh Keshatwar** |
| 1.4 | **02-09-2017** | **Updated document to Wipro Standard format with header and footer** | **Sivakumar Goud** | **Pradipkumar Kansara** | **Santosh Badiger** |
| 1.5 | **07-10-2018** | **Updated the content** | **Nanda Kishore Reddy** | **Pradipkumar Kansara** | **Santosh Badiger** |

**Table of Contents**

[1. Purpose 3](#_Toc270346443)

[2. Scope 3](#_Toc270346444)

[3. Overview 3](#_Toc270346445)

[4. Detailed source 3](#_Toc270346446)

[5. Different topology on create and delete](#_Toc270346447)  5

6. Diverse Classification on Create preference 7

# 1. Purpose

The purpose of this document is on creation/deletion of Database on SQL Server 2008.

# 2. Scope

Understanding the rationale usage and impact of create and delete or drop to specific to database in sql2008.

# 3. Overview

SQL 2008 is a rich platform for building database solutions that is found in many businesses from the very small to the extremely large. SQL 2008 comes in a number of versions with differing levels of functionality and scalability depending on usage needs.

# 4. Detailed Source

**Creating a Database**

* [SQL Server 2008](http://msdn.microsoft.com/en-us/library/ms175198(v=SQL.100).aspx)
* [SQL Server 2005](http://msdn.microsoft.com/en-us/library/ms175198(v=SQL.90).aspx)

To create a database, you have to determine the name of the database, its owner, its size, and the files and file groups used to store it.

Before creating a database, you should consider the following:

* To create a database, you must, at a minimum, have CREATE DATABASE, CREATE ANY DATABASE, or ALTER ANY DATABASE permission.
* In SQL Server, certain permissions are set on the data and log files of each database. The permissions prevent the files from being accidentally tampered with if they reside in a directory that has open permissions.
* The user who creates the database becomes the owner of the database.
* A maximum of 32,767 databases can be created on an instance of SQL Server.
* The name of the database must follow the rules specified
* All user-defined objects in the **model** database are copied to all newly created databases. You can add any objects, such as tables, views, stored procedures, and data types, to the **model** database to be included in all newly created databases.

**Database Files and File groups**

Three types of files are used to store a database. These include primary files, secondary files, and transaction logs. Your database must have a primary data file and at least one transaction log file. You can optionally create one or more secondary data files and additional transaction log files.

Primary files

These files contain the startup information for the database. The primary files are also used to store data. Every database has one primary file.

Secondary files

These files hold all the data that does not fit in the primary data file. Databases do not need secondary data files if the primary file is large enough to hold all the data in the database. Some databases may be large enough to require multiple secondary data files, or they may use secondary files on separate disk drives to spread the data across multiple disks.

Transaction logs

These files hold the log information used to recover the database. There must be at least one transaction log file for each database, although there may be more than one. The minimum size for a log file is 512 KB.

When you create a database, make the data files as large as possible, based on the maximum amount of data you expect in the database.

**File Initialization**

Data and log files are initialized to overwrite any existing data left on the disk from previously deleted files. These files are also initialized and filled with zeros when you perform one of the following operations:

* Create a database
* Add files to an existing database
* Increase the size of an existing file
* Restore a database or file group

In SQL Server, data files can be initialized instantaneously. This allows for fast execution of the previously mentioned file operations.

**Database Snapshots**

You can use the CREATE DATABASE statement to create a read-only static view, called a database snapshot, of an existing database, the source database. A database snapshot is consistent transaction-wise with the source database as it existed at the time of the snapshot's creation. A source database can have multiple snapshots.

**DROP DATABASE**

Removes one or more databases or database snapshots from an instance of SQL Server.

DROP DATABASE { database\_name | database\_snapshot\_name } [ ,...n ]

http://i.msdn.microsoft.com/Global/Images/clear.gif Arguments

*database\_name*

Specifies the name of the database to be removed. To display a list of databases, use the [sys.databases](http://technet.microsoft.com/en-us/library/ms178534.aspx) catalog view.

*database\_snapshot\_name*

Specifies the name of a database snapshot to be removed.

http://i.msdn.microsoft.com/Global/Images/clear.gif Remarks

To use DROP DATABASE, the database context of the connection cannot be the same as the database or database snapshot to be dropped.

The DROP DATABASE statement must run in autocommit mode and is not allowed in an explicit or implicit transaction. Autocommit mode is the default transaction management mode.

**Dropping a Database**

[System databases](http://technet.microsoft.com/en-us/library/ms178028.aspx) cannot be dropped.

Dropping a database deletes the database from an instance of SQL Server and deletes the physical disk files used by the database. If the database or any one of its files is offline when it is dropped, the disk files are not deleted. These files can be deleted manually by using Windows Explorer. To remove a database from the current server without deleting the files from the file system, use [sp\_detach\_db](http://technet.microsoft.com/en-us/library/ms188031.aspx).

You cannot drop a database currently being used. This means open for reading or writing by any user. To remove users from the database, use ALTER DATABASE to set the database to SINGLE\_USER.

Any database snapshots on a database must be dropped before the database can be dropped.

If the database is involved in log shipping, remove log shipping before dropping the database.

A database can be dropped regardless of its state: offline, read-only, suspect, and so on. To display the current state of a database, use the **sys.databases** catalog view.

A dropped database can be re-created only by restoring a backup. Database snapshots cannot be backed up and, therefore, cannot be restored.

When a database is dropped, the [master database](http://technet.microsoft.com/en-us/library/ms187837.aspx) should be backed up.

**Dropping a Database Snapshot**

Dropping a database snapshot deletes the database snapshot from an instance of SQL Server and deletes the physical NTFS File System sparse files used by the snapshot. For information about using sparse files by database snapshots, see [How Database Snapshots Work](http://technet.microsoft.com/en-us/library/ms187054.aspx).

Dropping a database snapshot clears the plan cache for the instance of SQL Server. Clearing the plan cache causes a recompilation of all subsequent execution plans and can cause a sudden, temporary decrease in query performance. For each cleared cachestore in the plan cache, the SQL Server error log contains the following informational message: "SQL Server has encountered %d occurrence(s) of cachestore flush for the '%s' cachestore (part of plan cache) due to some database maintenance or reconfigure operations". This message is logged every five minutes as long as the cache is flushed within that time interval.

**Dropping a Database Used in Replication**

To drop a database published for transactional replication, or published or subscribed to merge replication, you must first remove replication from the database. For more information about how to remove replication from a database, see [Removing Replication](http://technet.microsoft.com/en-us/library/ms152757.aspx). If a database is damaged or replication cannot first be removed or both, in most cases you still can drop the database by using ALTER DATABASE to set the database offline and then dropping it.

http://i.msdn.microsoft.com/Global/Images/clear.gif Permissions

To execute DROP DATABASE, at a minimum, a user must have CONTROL permission on the database.

http://i.msdn.microsoft.com/Global/Images/clear.gif Examples

**A. Dropping a single database**

The following example removes the Sales database.

DROP DATABASE Sales;

**B. Dropping multiple databases**

The following example removes each of the listed databases.

DROP DATABASE Sales, NewSales;

**C. Dropping a database snapshot**

The following example drops a database snapshot, named sales\_snapshot0600, without affecting the source database.

DROP DATABASE sales\_snapshot0600;

**Diverse Classification on Create preference**

Creates a new database and the files used to store the database, creates a database snapshot, or attaches a database from the detached files of a previously created database.

CREATE DATABASE database\_name

    [ ON

        { [ PRIMARY ] [ <filespec> [ ,...n ]

        [ , <filegroup> [ ,...n ] ]

    [ LOG ON { <filespec> [ ,...n ] } ] }

    ]

    [ COLLATE collation\_name ]

    [ WITH <external\_access\_option> ]

]

[;]

To attach a database

CREATE DATABASE database\_name

    ON <filespec> [ ,...n ]

FOR { ATTACH [ WITH <service\_broker\_option> ]

| ATTACH\_REBUILD\_LOG }

[;]

<filespec> ::=

{

(

    NAME =logical\_file\_name,

    FILENAME = { 'os\_file\_name' | 'filestream\_path' }

        [ , SIZE =size [ KB | MB | GB | TB ] ]

        [ , MAXSIZE = { max\_size [ KB | MB | GB | TB ] | UNLIMITED } ]

        [ , FILEGROWTH =growth\_increment [ KB | MB | GB | TB | % ] ]

) [ ,...n ]

}

<filegroup> ::=

{

FILEGROUP filegroup\_name [ CONTAINS FILESTREAM ] [ DEFAULT ]

    <filespec> [ ,...n ]

}

<external\_access\_option> ::=

{

  [ DB\_CHAINING { ON | OFF } ]

  [ , TRUSTWORTHY { ON | OFF } ]

}

<service\_broker\_option> ::=

{

    ENABLE\_BROKER

  | NEW\_BROKER

  | ERROR\_BROKER\_CONVERSATIONS

}

Create a database snapshot

CREATE DATABASE database\_snapshot\_name

    ON

    (

        NAME = logical\_file\_name,

        FILENAME ='os\_file\_name'

    ) [ ,...n ]

    AS SNAPSHOT OF source\_database\_name

[;]

**Arguments**

database\_name

Is the name of the new database. Database names must be unique within an instance of SQL Server and comply with the rules for [identifiers](http://msdn.microsoft.com/en-us/library/ms175874.aspx).

database\_name can be a maximum of 128 characters, unless a logical name is not specified for the log file. If a logical log file name is not specified, SQL Server generates the logical\_file\_name and the os\_file\_name for the log by appending a suffix to database\_name. This limits database\_name to 123 characters so that the generated logical file name is no more than 128 characters.

If data file name is not specified, SQL Server uses database\_name as both the logical\_file\_name and as the os\_file\_name. The default path is obtained from the registry. The default path can be change by using the Server Properties (Database Settings Page) in Management Studio. Changing the default path requires restarting SQL Server.

ON

Specifies that the disk files used to store the data sections of the database, data files, are explicitly defined. ON is required when followed by a comma-separated list of <filespec> items that define the data files for the primary filegroup. The list of files in the primary filegroup can be followed by an optional, comma-separated list of <filegroup> items that define user filegroups and their files.

PRIMARY

Specifies that the associated <filespec> list defines the primary file. The first file specified in the <filespec> entry in the primary filegroup becomes the primary file. A database can have only one primary file. For more information, see [Files and Filegroups Architecture](http://msdn.microsoft.com/en-us/library/ms179316.aspx).

If PRIMARY is not specified, the first file listed in the CREATE DATABASE statement becomes the primary file.

LOG ON

Specifies that the disk files used to store the database log, log files, are explicitly defined. LOG ON is followed by a comma-separated list of <filespec> items that define the log files. If LOG ON is not specified, one log file is automatically created that has a size that is 25 percent of the sum of the sizes of all the data files for the database or 512 KB, whichever is larger. This file is placed in the default log-

LOG ON cannot be specified on a database snapshot.

COLLATE collation\_name

Specifies the default collation for the database. Collation name can be either a Windows collation name or a SQL collation name. If not specified, the database is assigned the default collation of the instance of SQL Server. A collation name cannot be specified on a database snapshot.

A collation name cannot be specified with the FOR ATTACH or FOR ATTACH\_REBUILD\_LOG clauses. For information about how to change the collation of an attached database,

FOR ATTACH [ WITH <service\_broker\_option> ]

Specifies that the database is created by [attaching](http://msdn.microsoft.com/en-us/library/ms190794.aspx) an existing set of operating system files. There must be a <filespec> entry that specifies the primary file. The only other <filespec> entries required are those for any files that have a different path from when the database was first created or last attached. A <filespec> entry must be specified for these files.

FOR ATTACH requires the following:

* All data files (MDF and NDF) must be available.
* If multiple log files exist, they must all be available.

If a read/write database has a single log file that is currently unavailable, and if the database was shut down with no users or open transactions before the attach operation, FOR ATTACH automatically rebuilds the log file and updates the primary file. In contrast, for a read-only database, the log cannot be rebuilt because the primary file cannot be updated. Therefore, when you attach a read-only database whose log is unavailable, you must provide the log files or files in the FOR ATTACH clause.

In SQL Server, any full-text files that are part of the database that is being attached will be attached with the database. To specify a new path of the full-text catalog, specify the new location without the full-text operating system file name. For more information, see the Examples section.

FOR ATTACH cannot be specified on a database snapshot.

If the database uses Service Broker, use the WITH <service\_broker\_option> in your FOR ATTACH clause:

<service\_broker\_option>

Controls Service Broker message delivery and the Service Broker identifier for the database. Service Broker options can only be specified when the FOR ATTACH clause is used.

ENABLE\_BROKER

Specifies that Service Broker is enabled for the specified database. That is, message delivery is started and is\_broker\_enabled is set to true in the sys.databases catalog view. The database retains the existing Service Broker identifier.

NEW\_BROKER

Creates a new service\_broker\_guid value in both sys.databases and the restored database and ends all conversation endpoints with clean up. The broker is enabled, but no message is sent to the remote conversation endpoints. Any route that references the old Service Broker identifier must be re-created with the new identifier.

ERROR\_BROKER\_CONVERSATIONS

Ends all conversations with an error stating that the database is attached or restored. The broker is disabled until this operation is completed and then enabled. The database retains the existing Service Broker identifier.

When you attach a replicated database that was copied instead of being detached, consider the following:

* If you attach the database to the same server instance and version as the original database, no additional steps are required.
* If you attach the database to the same server instance but with an upgraded version, you must execute [sp\_vupgrade\_replication](http://msdn.microsoft.com/en-us/library/ms188741.aspx) to upgrade replication after the attach operation is complete.
* If you attach the database to a different server instance, regardless of version, you must execute [sp\_removedbreplication](http://msdn.microsoft.com/en-us/library/ms188734.aspx) to remove replication after the attach operation is complete.

FOR ATTACH\_REBUILD\_LOG

Specifies that the database is created by attaching an existing set of operating system files. This option is limited to read/write databases. There must be a <filespec> entry specifying the primary file. If one or more transaction log files are missing, the log file is rebuilt. The ATTACH\_REBUILD\_LOG automatically creates a new, 1-MB log file. This file is placed in the default log-file location. FOR ATTACH\_REBUILD\_LOG requires the following:

* A clean shutdown of the database.
* All data files (MDF and NDF) must be available.

Typically, FOR ATTACH\_REBUILD\_LOG is used when you copy a read/write database with a large log to another server where the copy will be used mostly, or only, for read operations, and will therefore require less log space than the original database.

FOR ATTACH\_REBUILD\_LOG cannot be specified on a database snapshot.

NAME logical\_file\_name

Specifies the logical name for the file. NAME is required when FILENAME is specified, except when specifying one of the FOR ATTACH clauses. A FILESTREAM filegroup cannot be named PRIMARY.

logical\_file\_name

Is the logical name used in SQL Server when referencing the file. Logical\_file\_name must be unique in the database and comply with the rules for [identifiers](http://msdn.microsoft.com/en-us/library/ms175874.aspx). The name can be a character or Unicode constant, or a regular or delimited identifier.

FILENAME { 'os\_file\_name' | 'filestream\_path' }

Specifies the operating system (physical) file name.

'os\_file\_name'

Is the path and file name used by the operating system when you create the file. The file must reside on one of the following devices: the local server on which SQL Server is installed, a Storage Area Network [SAN], or an iSCSI-based network. The specified path must exist before executing the CREATE DATABASE statement. For more information, see "Database Files and Filegroups" in the Remarks section.

SIZE, MAXSIZE, and FILEGROWTH parameters cannot be set when a UNC path is specified for the file.

If the file is on a raw partition, os\_file\_name must specify only the drive letter of an existing raw partition. Only one data file can be created on each raw partition.

Data files should not be put on compressed file systems unless the files are read-only secondary files, or the database is read-only. Log files should never be put on compressed file systems

'filestream\_path'

For a FILESTREAM filegroup, FILENAME refers to a path where FILESTREAM data will be stored. The path up to the last folder must exist, and the last folder must not exist. For example, if you specify the path C:\MyFiles\MyFilestreamData, C:\MyFiles must exist before you run ALTER DATABASE, but the MyFilestreamData folder must not exist.

The filegroup and file (<filespec>) must be created in the same statement. There can be only one file, <filespec>, for a FILESTREAM filegroup.

The SIZE, MAXSIZE, and FILEGROWTH properties do not apply to a FILESTREAM filegroup.

SIZE size

Specifies the size of the file.

SIZE cannot be specified when the os\_file\_name is specified as a UNC path. SIZE does not apply to a FILESTREAM filegroup.

size

Is the initial size of the file.

When size is not supplied for the primary file, the Database Engine uses the size of the primary file in the model database. When a secondary data file or log file is specified but size is not specified for the file, the Database Engine makes the file 1 MB. The size specified for the primary file must be at least as large as the primary file of the model database.

The kilobyte (KB), megabyte (MB), gigabyte (GB), or terabyte (TB) suffixes can be used. The default is MB. Specify a whole number; do not include a decimal. Size is an integer value. For values greater than 2147483647, use larger units.

MAXSIZE max\_size

Specifies the maximum size to which the file can grow. MAXSIZE cannot be specified when the os\_file\_name is specified as a UNC path. MAXSIZE does not apply to a FILESTREAM filegroup.

max\_size

Is the maximum file size. The KB, MB, GB, and TB suffixes can be used. The default is MB. Specify a whole number; do not include a decimal. If max\_size is not specified, the file grows until the disk is full. Max\_size is an integer value. For values greater than 2147483647, use larger units.

UNLIMITED

Specifies that the file grows until the disk is full. In SQL Server, a log file specified with unlimited growth has a maximum size of 2 TB, and a data file has a maximum size of 16 TB.

FILEGROWTH growth\_increment

Specifies the automatic growth increment of the file. The FILEGROWTH setting for a file cannot exceed the MAXSIZE setting. FILEGROWTH cannot be specified when the os\_file\_name is specified as a UNC path. FILEGROWTH does not apply to a FILESTREAM filegroup.

growth\_increment

Is the amount of space added to the file every time new space is required.

The value can be specified in MB, KB, GB, TB, or percent (%). If a number is specified without an MB, KB, or % suffix, the default is MB. When % is specified, the growth increment size is the specified percentage of the size of the file at the time the increment occurs. The size specified is rounded to the nearest 64 KB.

A value of 0 indicates that automatic growth is off and no additional space is allowed.

If FILEGROWTH is not specified, the default value is 1 MB for data files and 10% for log files, and the minimum value is 64 KB.

<filegroup>

Controls the filegroup properties. Filegroup cannot be specified on a database snapshot.

FILEGROUP filegroup\_name

Is the logical name of the filegroup.

filegroup\_name

filegroup\_name must be unique in the database and cannot be the system-provided names PRIMARY and PRIMARY\_LOG. The name can be a character or Unicode constant, or a regular or delimited identifier. The name must comply with the rules for [identifiers](http://msdn.microsoft.com/en-us/library/ms175874.aspx).

CONTAINS FILESTREAM

Specifies that the filegroup stores FILESTREAM binary large objects (BLOBs) in the file system.

DEFAULT

Specifies the named filegroup is the default filegroup in the database.

<external\_access\_option>

Controls external access to and from the database.

DB\_CHAINING { ON | OFF }

When ON is specified, the database can be the source or target of a cross-database ownership chain.

When OFF, the database cannot participate in cross-database ownership chaining. The default is OFF.

To set this option, requires membership in the sysadmin fixed server role. The DB\_CHAINING option cannot be set on these system databases: master, model, tempdb.

TRUSTWORTHY { ON | OFF }

When ON is specified, database modules (for example, views, user-defined functions, or stored procedures) that use an impersonation context can access resources outside the database.

When OFF, database modules in an impersonation context cannot access resources outside the database. The default is OFF.

TRUSTWORTHY is set to OFF whenever the database is attached.

By default, all system databases except the msdb database have TRUSTWORTHY set to OFF. The value cannot be changed for the model and tempdb databases. We recommend that you never set the TRUSTWORTHY option to ON for the master database.

To set this option, requires membership in the sysadmin fixed server role.

database\_snapshot\_name

Is the name of the new database snapshot. Database snapshot names must be unique within an instance of SQL Server and comply with the rules for identifiers. database\_snapshot\_name can be a maximum of 128 characters.

ON ( NAME =logical\_file\_name, FILENAME = 'os\_file\_name') [ ,... n ]

For creating a database snapshot, specifies a list of files in the source database. For the snapshot to work, all the data files must be specified individually. However, log files are not allowed for database snapshots. FILESTREAM filegroups are not supported by database snapshots. If a FILESTREAM data file is included in a CREATE DATABASE ON clause, the statement will fail and an error will be raised.

For descriptions of NAME and FILENAME and their values see the descriptions of the equivalent <filespec> values.

AS SNAPSHOT OF source\_database\_name

Specifies that the database being created is a database snapshot of the source database specified by source\_database\_name. The snapshot and source database must be on the same instance.