**SQL Server Database Migration Checklist**

By: [Jugal Shah](https://www.mssqltips.com/sqlserverauthor/47/jugal-shah/" \o "author profile for Jugal Shah)   |   Updated: 2010-02-05   |   [Comments (11)](https://www.mssqltips.com/sqlservertip/1936/sql-server-database-migration-checklist/" \l "comments" \o "read and post comments)   |   Related: [More](https://www.mssqltips.com/sql-server-dba-resources/) > [Upgrades and Migrations](https://www.mssqltips.com/sql-server-tip-category/24/upgrades-and-migrations/)

**Problem**

Many times we get requests to migrate a single user database or multiple user databases to different server. At the time of migrating databases, there are chances that we can miss some important steps, for example trustworthy property, dependent jobs, linked server, logins etc. There are also chances we are missing important DBA items which can hurt SQL Server performance.  In this tip, I cover tasks that should be followed when moving databases from one server to another server.

**Solution**

As a solution to decrease making mistakes during the migration process I have developed the below database migration checklist. In addition, there are steps which should be taken which can enhance the database performance after the migration.

**Pre-Migration Checklist**

1. Analyze the disk space of the target server for the new database, if the disk space is not enough add more space on the target server
2. Confirm the data and log file location for the target server
3. Collect the information about the Database properties (Auto Stats, DB Owner, Recovery Model, Compatibility level, Trustworthy option etc)
4. Collect the information of dependent applications, make sure application services will be stopped during the database migration
5. Collect the information of database logins, users and their permissions. (Optional)
6. Check the database for the Orphan users if any
7. Check the SQL Server for any dependent objects (SQL Agent Jobs and Linked Servers)
8. Check, if the database is part of any maintenance plan

Below are various scripts you can run to collect data.

**Script to Check the Disk and Database Size**

-- Procedure to check disc space

exec master..xp\_fixeddrives

-- To Check database size

exec sp\_helpdb [dbName]

or

use [dbName]

select str(sum(convert(dec(17,2),size)) / 128,10,2) + 'MB'

from dbo.sysfiles

GO

**Script to Check Database Properties**

select

sysDB.database\_id,

sysDB.Name as 'Database Name',

syslogin.Name as 'DB Owner',

sysDB.state\_desc,

sysDB.recovery\_model\_desc,

sysDB.collation\_name,

sysDB.user\_access\_desc,

sysDB.compatibility\_level,

sysDB.is\_read\_only,

sysDB.is\_auto\_close\_on,

sysDB.is\_auto\_shrink\_on,

sysDB.is\_auto\_create\_stats\_on,

sysDB.is\_auto\_update\_stats\_on,

sysDB.is\_fulltext\_enabled,

sysDB.is\_trustworthy\_on

from sys.databases sysDB

INNER JOIN sys.syslogins syslogin ON sysDB.owner\_sid = syslogin.sid

**Another Script to Check Database Properties**

declare @dbdesc varchar(max)

declare @name varchar(10)

set @name='Master'

SELECT @dbdesc = 'Status=' + convert(sysname,DatabasePropertyEx(@name,'Status'))

SELECT @dbdesc = @dbdesc + ', Updateability=' + convert(sysname,DatabasePropertyEx(@name,'Updateability'))

SELECT @dbdesc = @dbdesc + ', UserAccess=' + convert(sysname,DatabasePropertyEx(@name,'UserAccess'))

SELECT @dbdesc = @dbdesc + ', Recovery=' + convert(sysname,DatabasePropertyEx(@name,'Recovery'))

SELECT @dbdesc = @dbdesc + ', Version=' + convert(sysname,DatabasePropertyEx(@name,'Version'))

-- These props only available if db not shutdown

IF DatabaseProperty(@name, 'IsShutdown') = 0

BEGIN

SELECT @dbdesc = @dbdesc + ', Collation=' + convert(sysname,DatabasePropertyEx(@name,'Collation'))

SELECT @dbdesc = @dbdesc + ', SQLSortOrder=' + convert(sysname,DatabasePropertyEx(@name,'SQLSortOrder'))

END

-- These are the boolean properties

IF DatabasePropertyEx(@name,'IsAutoClose') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsAutoClose'

IF DatabasePropertyEx(@name,'IsAutoShrink') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsAutoShrink'

IF DatabasePropertyEx(@name,'IsInStandby') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsInStandby'

IF DatabasePropertyEx(@name,'IsTornPageDetectionEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsTornPageDetectionEnabled'

IF DatabasePropertyEx(@name,'IsAnsiNullDefault') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsAnsiNullDefault'

IF DatabasePropertyEx(@name,'IsAnsiNullsEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsAnsiNullsEnabled'

IF DatabasePropertyEx(@name,'IsAnsiPaddingEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsAnsiPaddingEnabled'

IF DatabasePropertyEx(@name,'IsAnsiWarningsEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsAnsiWarningsEnabled'

IF DatabasePropertyEx(@name,'IsArithmeticAbortEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsArithmeticAbortEnabled'

IF DatabasePropertyEx(@name,'IsAutoCreateStatistics') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsAutoCreateStatistics'

IF DatabasePropertyEx(@name,'IsAutoUpdateStatistics') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsAutoUpdateStatistics'

IF DatabasePropertyEx(@name,'IsCloseCursorsOnCommitEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsCloseCursorsOnCommitEnabled'

IF DatabasePropertyEx(@name,'IsFullTextEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsFullTextEnabled'

IF DatabasePropertyEx(@name,'IsLocalCursorsDefault') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsLocalCursorsDefault'

IF DatabasePropertyEx(@name,'IsNullConcat') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsNullConcat'

IF DatabasePropertyEx(@name,'IsNumericRoundAbortEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsNumericRoundAbortEnabled'

IF DatabasePropertyEx(@name,'IsQuotedIdentifiersEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsQuotedIdentifiersEnabled'

IF DatabasePropertyEx(@name,'IsRecursiveTriggersEnabled') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsRecursiveTriggersEnabled'

IF DatabasePropertyEx(@name,'IsMergePublished') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsMergePublished'

IF DatabasePropertyEx(@name,'IsPublished') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsPublished'

IF DatabasePropertyEx(@name,'IsSubscribed') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsSubscribed'

IF DatabasePropertyEx(@name,'IsSyncWithBackup') = 1

SELECT @dbdesc = @dbdesc + ', ' + 'IsSyncWithBackup'

SELECT @dbdesc

**Script to List Orphan Users**

sp\_change\_users\_login 'report'

GO

**Script to List Linked Servers**

select \*

from sys.sysservers

**Script to List Database Dependent Jobs**

select

distinct

name,

database\_name

from sysjobs sj

INNER JOIN sysjobsteps sjt on sj.job\_id = sjt.job\_id

**Database Migration Checklist**

These are the steps you would go through to make the change.

1. Stop the application services

2. Change the database to read-only mode (Optional)

-- Script to make the database readonly

USE [master]

GO

ALTER DATABASE [DBName] SET READ\_ONLY WITH NO\_WAIT

GO

ALTER DATABASE [DBName] SET READ\_ONLY

GO

3. Take the latest backup of all the databases involved in migration

4. Restore the databases on the target server on the appropriate drives

5. Cross check the database properties as per the database property script output, change the database properties as per the pre migration- checklist

**Script to Change DB Owner**

This will change the database owner to "sa".  This can be used to change to any owner you would like.

USE databaseName

EXEC sp\_changedbowner 'sa'

**Script to Turn on Trustworthy Option**

If trustworthy option was set, this will turn it on for the database.

ALTER DATABASE database\_name SET TRUSTWORTHY ON

**Script to Change the Database Compatibility Level**

When you upgrade to a new version, the old compatibility level will remain.  This script shows [how to change the compatibility level](https://www.mssqltips.com/sqlservertip/1436/upgrading-sql-server-databases-and-changing-compatibility-levels/) to SQL Server 2005 compatibility .

ALTER DATABASE DatabaseName

SET SINGLE\_USER

GO

EXEC sp\_dbcmptlevel DatabaseName, 90;

GO

ALTER DATABASE DatabaseName

SET MULTI\_USER

GO

6. Execute the output of Login transfer script on the target server, to create logins on the target server you can get the code from this technet article: <http://support.microsoft.com/kb/246133>.  
  
7. Check for [Orphan Users](https://www.mssqltips.com/sqlservertip/1590/understanding-and-dealing-with-orphaned-users-in-a-sql-server-database/) and Fix Orphan Users

**Script to Check and Fix Orphan Users**

-- Script to check the orphan user

EXEC sp\_change\_users\_login 'Report'

--Use below code to fix the Orphan User issue

DECLARE @username varchar(25)

DECLARE fixusers CURSOR

FOR

SELECT UserName = name FROM sysusers

WHERE issqluser = 1 and (sid is not null and sid <> 0x0)

and suser\_sname(sid) is null

ORDER BY name

OPEN fixusers

FETCH NEXT FROM fixusers

INTO @username

WHILE @@FETCH\_STATUS = 0

BEGIN

EXEC sp\_change\_users\_login 'update\_one', @username, @username

FETCH NEXT FROM fixusers

INTO @username

END

CLOSE fixusers

DEALLOCATE fixusers

8. Execute DBCC UPDATEUSAGE on the restored database.  
  
Run the DBCC UPDATEUSAGE command against the migrated database when upgrading to a newer version of SQL Server.

DBCC UPDATEUSAGE('database\_name') WITH COUNT\_ROWS

DBCC CHECKDB

OR

DBCC CHECKDB('database\_name') WITH ALL\_ERRORMSGS

9. Rebuild Indexes (Optional)As per the requirement and time window you can execute this option.

Take a look at this tip to [rebuild all indexes](https://www.mssqltips.com/sqlservertip/1367/sql-server-script-to-rebuild-all-indexes-for-all-tables-and-all-databases/).

This will rebuild or reorganize all indexes for a particular table.  
  
**Index Rebuild :-**This process drops the existing Index and Recreates the index.  
**Index Reorganize :-**This process physically reorganizes the leaf nodes of the index.

-- Script for Index Rebuild

USE [DBName];

GO

ALTER INDEX ALL ON [ObjectName] REBUILD

GO

-- Script for Index Reorganize

USE AdventureWorks;

GO

ALTER INDEX ALL ON [ObjectName] REORGANIZE

GO

10. [Update index statistics](https://www.mssqltips.com/sqlservertip/1606/execute-update-statistics-for-all-sql-server-databases/)

sp\_updatestats

11. Recompile procedures

Take a look at this tip to [recompile all objects](https://www.mssqltips.com/sqlservertip/1260/script-to-recompile-all-sql-server-database-objects/).

This will recompile a particular stored procedure.

sp\_recompile 'procedureName'

12. Start the application services, check the application functionality and check the Windows event logs.  
  
13. Check the SQL Server Error Log for login failures and other errors

Take a look at this tip on how to[read SQL Server error logs](https://www.mssqltips.com/sqlservertip/1476/reading-the-sql-server-log-files-using-tsql/).

EXEC xp\_readerrorlog 0,1,"Error",Null

14. Once the application team confirms that application is running fine take the databases offline on the source server or make them read only

-- Script to make the database readonly

USE [master]

GO

ALTER DATABASE [DBName] SET READ\_ONLY WITH NO\_WAIT

GO

ALTER DATABASE [DBName] SET READ\_ONLY

GO

-- Script to take the database offline

EXEC sp\_dboption N'DBName', N'offline', N'true'

OR

ALTER DATABASE [DBName] SET OFFLINE WITH

ROLLBACK IMMEDIATE

**Next Steps**

* Test the process to determine how much time and disk space would be needed by using the backup and recovery process.
* Meet with your technical and business teams to find out how much time is available for the migration and plan the activity
* Design the rollback plan, if the application is not working fine
* Add more migration cases in your checklist, for example check if the database requires any server level change (For example CLR, XP\_Cmdshell etc)
* Some of these scripts give you the base command to update a portion of the data, enhance the process to hit each object in your database.