



## **Guidelines for using Double-Take<sup>®</sup> to Upgrade from SQL Server 2000 to SQL Server 2005**



Guidelines for using Double-Take to Upgrade from SQL Server 2000 to SQL Server 2005

Revision 1.0.0 published April 2006

Double-Take, GeoCluster, and NSI are registered trademarks of NSI Software, Inc. Balance and Double-Take for Virtual Systems are trademarks of NSI Software, Inc. Microsoft, Windows, and the Windows logo are trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. All other trademarks are the property of their respective companies.

© 2006 NSI Software, Inc. d.b.a. Double-Take Software. All rights reserved.

## *Double-Take<sup>®</sup> Support for Application Failover*

The Double-Take<sup>®</sup> file system replication process is application independent and replicates any file system changes (including permissions and attributes) written to NTFS, FAT or FAT32 file systems by any application or process, subject to specific exceptions called out in the *User's Guide* or readme file. Maintaining point-in-time consistent file system replicas and providing server monitoring and automatic or manual failover of the server name and IP address are the primary functions of the Double-Take software and we offer support to qualified customers should these functions fail to operate in accordance with our published documentation, regardless of what application or process is manipulating the data.

NSI Software, Inc. d.b.a. Double-Take Software may provide application notes and other documents that provide implementation guidelines on how to use Double-Take functions and replicas to manually or automatically failover or recover many popular third-party applications and a general process to accomplish failover or recovery of many other third-party applications. While these steps are believed to be accurate for the specific configuration, Double-Take version, and application versions originally tested, due to the number of possible configurations and variables, Double-Take Software can only test selected combinations and may provide only limited support for the operation and configuration of third-party applications or the behavior of those applications before, during, or after failover, in its discretion. In cases where Double-Take Software has no direct access to or experience with a particular application or configuration, Double-Take Software support may also be limited to only the actual replication of the file system data and failover (name and IP address) of the server.

For assistance in validating, implementing or troubleshooting these or other possible configurations with third-party applications, Double-Take Software and its partners may offer professional services on a fee basis to apply best practices for assisting with third-party applications to recover automatically or manually using replicated data. This, and any other, application note is provided solely for the convenience of our customers and is not intended to bind Double-Take Software to any obligation. Although we try to provide quality information, Double-Take Software makes no claims, promises or guarantees about the accuracy, completeness, or adequacy of the information contained in this document.

# Table of Contents

---

<b>I Document Overview .....</b>	<b>I</b>
1.1 Audience .....	I
1.2 Expectations .....	I
1.3 Related documentation .....	I
1.4 Getting help .....	I
<b>2 Solution Overview .....</b>	<b>2</b>
2.1 Advantages of using Double-Take® in a SQL migration .....	2
2.2 How the migration solution works .....	3
2.3 How the testing solution works .....	4
2.4 Modifying the sample script files .....	5
<b>3 Sample Implementation .....</b>	<b>6</b>
3.1 Requirements .....	6
3.2 Install software on the source .....	6
3.3 Install software on the target .....	6
3.4 Configure mirroring and replication .....	7
3.5 Disconnect clients and verify that queues are empty .....	7
3.6 Attach the target databases .....	8
3.7 Detach the source databases .....	9
3.8 Post-upgrade recommendations .....	9
3.8.1 Update statistics .....	9
3.8.2 Update counts .....	9
3.8.3 Change compatibility mode .....	10
3.9 Move login permissions .....	10
3.10 Configure client connectivity .....	10

---

# I. Document Overview

This document is a Double-Take<sup>®</sup> application note. An application note provides guidelines on the use of Double-Take in a specific environment.

This document contains:

- ◆ **Document Overview**—Explains what an application note contains, how it should be used, what you need to know before trying to use the application note, and where you can go for more information.
- ◆ **Solution Overview**—Explains how the solution works with Double-Take. Review this section to make sure that you understand the theory involved with using Double-Take for this solution.
- ◆ **Sample Implementation**—Describes specific examples of how to use Double-Take for this solution. This includes information about the system setup used in the sample implementation. Use these procedures as a guideline for creating your own implementation. Because no two environments or configurations are exactly the same, you will probably need to implement additional or different steps than what is documented here in order to make the solution work in your environment.

## I.1. Audience

This document is written for network and application administrators who have a working understanding of the applications and environments where the Double-Take solution is to be deployed. You may need to expand on the documented information in order to customize the solution to fit your environment.

Before you use this application note, you should have an understanding of:

- ◆ Double-Take
- ◆ SQL Server 2000
- ◆ SQL Server 2005

## I.2. Expectations

Application notes are intended to provide a framework for configuring a Double-Take solution in a specific environment and to draw attention to decisions you will need to make when configuring your solution.

Because there are an infinite number of possible configuration, network, and environment scenarios, application notes contain general configuration guidelines as well as an example configuration procedure that has been tested for a specific environment.

This document assumes that you are comfortable working with your operating system, Double-Take, and SQL Server.

## I.3. Related documentation

Before you begin to configure your solution, make sure that you have complete documentation for your operating system, application, and Double-Take. This application note does not provide step-by-step instructions for using standard operating system, application, and Double-Take functionality.

The following documents contain additional information that you may need while setting up this solution:

- ◆ Double-Take *User's Guide* or online documentation
- ◆ Reference guides or documentation for SQL Server

## I.4. Getting help

NSI Software, Inc., d.b.a. Double-Take Software has application notes that describe how to configure Double-Take with a variety of popular third-party applications. These application notes are available on the Application Notes page of the Double-Take Software support web site (<http://support.doubletake.com>).

For help using Double-Take, refer to the Double-Take online manual or online help.

Double-Take Software offers professional services on a fee basis to assist you in identifying the best practices for implementing a solution in your environment. Visit <http://www.doubletake.com/what-we-offer/services/> for more information.

---

## 2. Solution Overview

This document describes how you can use Double-Take to minimize downtime during a migration from SQL Server 2000 to SQL Server 2005. In addition, you can use the process described in this document to verify that your migration has completed successfully.

### 2.1. Advantages of using Double-Take<sup>®</sup> in a SQL migration

The following table identifies the most common supported methods that can be used to facilitate migration of databases from SQL Server 2000 to SQL Server 2005, along with the limitations of using each method.

Method	Limitations
Upgrade SQL binaries in-place using SQL 2005 installation media	<ul style="list-style-type: none"><li>◆ SQL server is unavailable during this process.</li><li>◆ Risk of negative impact to production database and SQL server if there is a problem during upgrade.</li></ul>
Perform traditional backup and restore to new SQL 2005 server	<ul style="list-style-type: none"><li>◆ Speed of tape backup and restore is not conducive to quick upgrades.</li><li>◆ Restored data will not be up to date if changes are made to original SQL 2000 database during backup/restore.</li><li>◆ To ensure that the new database has all of the legacy data, the original SQL 2000 database would need to be offline until the new SQL 2005 database was completely configured and deployed.</li></ul>
Use "Copy database Wizard" and "Detach and attach" to copy the databases to the new SQL 2005 server and attach them to the SQL instance	The source SQL server is unavailable during the database copy.

By using Double-Take to assist with the migration of databases from SQL Server 2000 to SQL Server 2005, you can overcome most of these limitations. The migration can be done in real time, and the amount of downtime during which clients cannot access the production servers is kept to a minimum.

By replicating changes to the data on the source to the target in real time, an up-to-the-second copy of the data is maintained on the target, ready for the migration to occur at the time most convenient for administrators and clients. This results in greater flexibility when planning the migration for administrators and reduced downtime during the migration for clients.

---

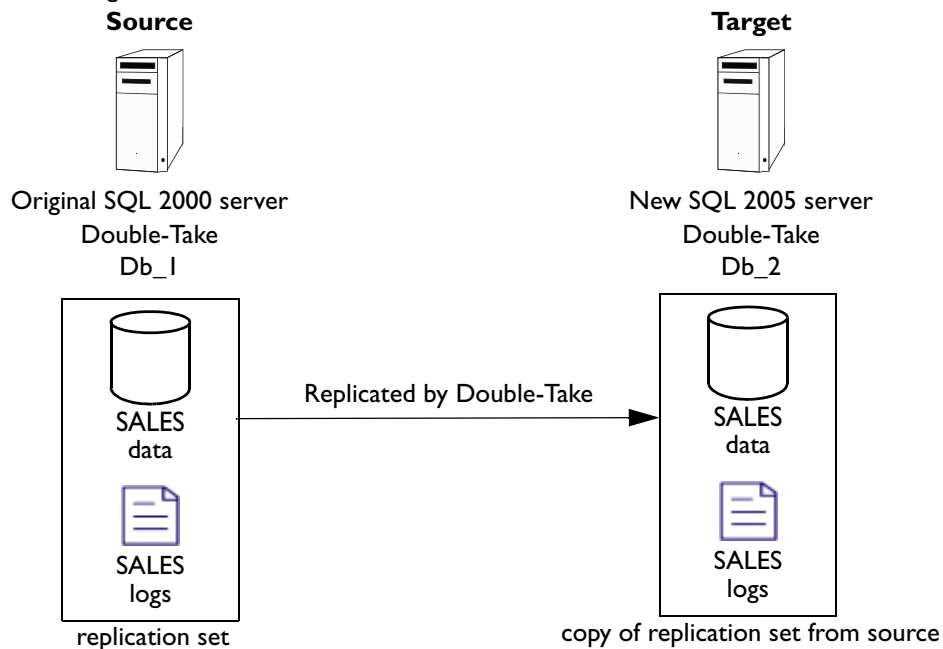
## 2.2. How the migration solution works

This section provides a high-level overview of the migration solution. For a specific example that includes a more detailed look at the steps involved in the process, see [Sample Implementation](#) on page 6.

1. A new SQL 2005 server is built on new or existing hardware.
2. Double-Take is installed on both the existing SQL 2000 server (source) and the new SQL 2005 server (target).



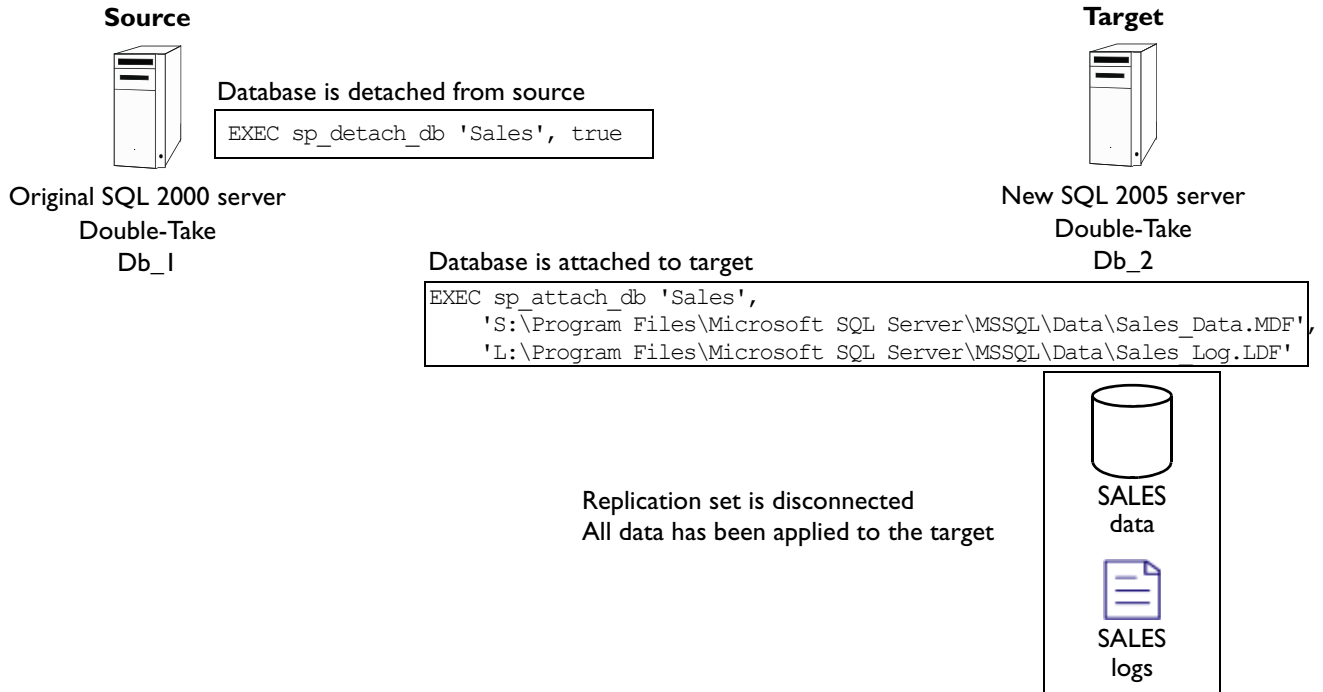
3. A replication set is created to replicate the SQL databases and logs from the SQL Server 2000 source server to the SQL Server 2005 target server. However, unlike a typical Double-Take high availability scenario, the SQL 2005 services on the target are left running.



4. Disconnect clients so that they cannot access the source.
5. Stop any applications from accessing SQL data on the source, and stop application and SQL services on source.
6. Verify that all data has been sent to target.
7. Disconnect the replication set between the source and target.

- Attach the databases to the SQL Server 2005 target using Query Analyzer and/or the OSQL utility with a SQL command file.

If you need more information about using Query Analyzer or the OSQL command line utility, refer to your SQL documentation.



- Detach the databases from the SQL Server 2000 source using the stored procedure `sp_detach_db` from within Query Analyzer, or using the OSQL command line utility on the source server.
- Move login permissions.
- Either change the target identity to match the SQL 2000 server, then restart the target (so that clients can connect using the original server identity), or redirect clients and applications to new SQL 2005 server.  
if you change the target name to the source name but retain the original IP on the target, then DNS entries may need to be updated.

## 2.3. How the testing solution works

- A new SQL 2005 server is built on new or existing hardware.
- Double-Take is installed on both the existing SQL 2000 server (source) and the new SQL 2005 server (target).
- A replication set is created to replicate the SQL databases and logs from the SQL Server 2000 source server to the SQL Server 2005 target server. However, unlike a typical Double-Take high availability scenario, the SQL 2005 services on the target are left running.
- Disconnect the replication set between the source and target.
- Attach the databases to the SQL Server 2005 target using Query Analyzer and/or the OSQL utility with a SQL command file.
- Verify that the databases attached successfully.
- Move login permissions.
- (Optional) Reconfigure client to attach to target to test.

---

## 2.4. Modifying the sample script files

You will need to edit the `DetachDB.sql` and `AttachDB.sql` command files, as well as the `AttachSQL.bat` script file, to include the correct database names and paths for your environment.

---

**NOTE:** These sample script and batch files are available on the Double-Take Software support web site at <http://support.doubletake.com>.

After you modify the sample scripts, save them with a new name to remove the `sample_` prefix. Copy the scripts to the directory where Double-Take is installed.

The sample batch files provided are only examples. Because no two environments or configurations are exactly the same, you **MUST** modify the sample scripts in order to make the solution work in your environment.

---

The following sample script files are used in this process:

- ◆ **DetachDB.sql**—Sample SQL command file to detach the SQL 2000 databases using Query Analyzer or the OSQL command line utility.
- ◆ **AttachDB.sql**—Sample script attach SQL databases, called from the `AttachSQL.bat` file.
- ◆ **AttachSQL.bat**—Sample batch file to attach SQL databases using the OSQL command line utility.

---

## 3. Sample Implementation

This section describes how this solution has been implemented to use Double-Take to assist with a migration from SQL Server 2000 to SQL Server 2005 .

---

**NOTE:** Because no two environments or configurations are exactly the same, you will need to modify the sample implementation in order to make the solution work in your environment.

---

### 3.1. Requirements

For this solution, your system must meet the following requirements:

Source server:

- ◆ Microsoft® Windows® Server™ 2000 with the most recent service pack or Windows Server 2003 with service pack 1 or later
- ◆ Double-Take version 4.4 with the most recent service pack
- ◆ SQL Server 2000 with service pack 4 or later

Target server:

- ◆ Microsoft Windows Server 2000 with the most recent service pack or Windows Server 2003 with service pack 1 or later
- ◆ Double-Take version 4.4 with the most recent service pack
- ◆ SQL Server 2005 with the most recent service pack

### 3.2. Install software on the source

1. Install Double-Take on the source server using the installation defaults. See the Double-Take *Getting Started* guide for details.
2. Record the drive and directory path(s) where the source SQL server database and log files are stored. The default directory for SQL 2000 is <drive>:\Program Files\Microsoft SQL Server\MSSQL\Data

Source SQL database and log file paths:

---

---

---

---

### 3.3. Install software on the target

1. Install SQL Server 2005 on the target, if it is not already installed.
2. If your SQL installation paths are not identical on both the source and target servers, or if you want to place the replicated source files in specific locations, record the drive and directory paths where you want the source database and respective log files to be placed on the target server. The default directory for SQL 2005 is <drive>:\Program Files\Microsoft SQL Server\MSSQL\Data

Target SQL database and log file paths:

---

---

---

---

3. Install Double-Take on the target server using the installation defaults. See the Double-Take *Getting Started* guide for details.

---

## 3.4. Configure mirroring and replication

1. Select **Start, Programs, Double-Take, Management Console**.
2. Double-click your source machine to log on.
3. Right-click on the source and select **Properties**.
4. On the Source tab, enable **Block Checksum All Files on a Difference Mirror** and click **OK**.
5. Right-click your source machine and select **New, Replication Set**. Enter the desired name for the replication set.
6. Using the list of file paths created in *Install software on the source* on page 6, expand the tree under the newly-created replication set and select the user-created database (.MDF) files and their associated log (.LDF) files for inclusion in the replication set.
7. Right-click the replication set name and select **Save** to save the replication set.
8. Drag and drop the replication set onto the target. The Connection Manager opens.
9. The **Source Server, Target Server, Replication Set, and Route** fields will automatically be populated. If you have multiple IP addresses on your target, verify the **Route** field is set to the correct network path. (For detailed information on connecting a source and target, see the *Double-Take User's Guide*.)
10. If your desired target path is identical to the source path, select the **One to One** option in the Mappings section.  
If you want all of the source files to go to a single location, select the **All to One** option in the Mappings section. You may edit the Target Path shown by clicking on it. Either type in the desired target path, or browse to it using the button next to the path edit box.
11. On the Orphans tab, verify that the **Move/Delete Orphan Files** checkbox is **NOT** selected.
12. On the Mirroring tab, select the type of mirror, either **Full** or **File Differences**, to perform.

---

**NOTE:** If the target has been previously mirrored to or restored, select **File differences** with the **Use block checksum** option so that only the changed data is sent across the network.

---

13. Click **Connect** to start the mirror and replication processes.

---

**NOTE:** If you start SQL Server and attach and mount the replicated databases on the target, or if the data on the target is otherwise modified, the data on the source and target will no longer match. If the updated data on the target is not needed, perform a full or difference with block checksum mirror from the source to the target. If the updated data on the target is needed, restore the data from the target to the source.

---

Your situation may be such that your source data files and log files are at various disk/path locations, and you want them to go to divergent disk/path locations on the target server. In this case, use the paths that you recorded in *Install software on the source* on page 6 to help create separate replication sets for each situation. Repeat steps 1-13 above to select the correct source file paths for the replication sets and use the disk/file paths recorded in *Install software on the target* on page 6 to set the correct target file paths for each situation.

## 3.5. Disconnect clients and verify that queues are empty

1. When you are ready to perform the migration, disconnect any clients that may be accessing the source databases. Stop any applications that may access the SQL data on the source, including any of the application's services. This should flush any data that may be stored in the `TEMPDB` database to the appropriate user-created database for that application.
2. Verify that all data has been sent to the target before continuing. You may use Performance Monitor (Perfmon) on the source server to check for any queued Double-Take data. See the *Double-Take User's Guide* for information on using Performance Monitor to monitor the source queue.

---

**NOTE:** The Double-Take source queue **MUST** be empty before proceeding.

---

3. Using the Double-Take Management Console, double-click on the source server in the left pane to log in. In the right pane, right-click on the connection(s) between the source and target servers and select **Disconnect**.

---

## 3.6. Attach the target databases

Attaching the replicated databases to the target SQL 2005 server is best accomplished using a combination of a SQL command file (.sql) and a batch file to call the OSQL utility, using the command file for input.

The attach procedure will upgrade the SQL 2000 databases to SQL 2005 databases during the attach process.

---

**NOTE:** Verify that all data has been written to the target before continuing. You may use Perfmon on the target server to check for any queued Double-Take data. See the *Double-Take User's Guide* for information on using Performance Monitor to monitor the target queues.

The Double-Take target queues **MUST** be empty before proceeding.

---

- I. Using Notepad or any text file editor, create a SQL command file and save it with a .sql extension. The file will utilize the stored procedure `sp_attach_db`. The following sample file illustrates the command syntax for attaching three sample databases, named *Sales*, *Accounting*, and *Marketing*. Note that the complete path and filespec to the database and corresponding log files on the target must be included. Refer to the file paths recorded in *Install software on the target* on page 6.

---

**NOTE:** These sample batch files are available on the Double-Take Software support web site at <http://support.doubletake.com>.

After you modify the sample scripts, save them with a new name to remove the `sample_` prefix. Copy the scripts to the directory where Double-Take is installed.

The sample batch files provided are only examples. Because no two environments or configurations are exactly the same, you **MUST** modify the sample scripts in order to make the solution work in your environment.

---

### SAMPLE\_AttachDB.sql

```
-- ***SAMPLE*** commands for attaching the SQL databases to the target server using Query Analyzer
-- or the OSQL command line utility.

-- This sample file is provided as an example only. Because no two environments or configurations are
-- exactly the same, you MUST modify this file in order to make the solution work in your environment.

-- This example shows divergent paths on the target server for the various databases and
-- corresponding transaction logs.

EXEC sp_attach_db 'Sales',
    'S:\Program Files\Microsoft SQL Server\MSSQL\Data\Sales_Data.MDF',
    'L:\Program Files\Microsoft SQL Server\MSSQL\Data\Sales_Log.LDF'
EXEC sp_attach_db 'Accounting',
    'T:\Program Files\Microsoft SQL Server\MSSQL\Data\Accounting_Data.MDF',
    'I:\Program Files\Microsoft SQL Server\MSSQL\Data\Accounting_Log.LDF'
EXEC sp_attach_db 'Marketing',
    'M:\Program Files\Microsoft SQL Server\MSSQL\Data\Marketing_Data.MDF',
    'L:\Program Files\Microsoft SQL Server\MSSQL\Data\Marketing_Log.LDF'
```

- 
2. Create a batch file on the target to execute the OSQL utility using the AttachDB command file created above as an input file. The following sample file illustrates the correct syntax.

#### SAMPLE\_AttachSQL.bat

```
rem ***SAMPLE*** script for attaching SQL databases.

rem This sample file is provided as an example only. Because no two environments or configurations are
rem exactly the same, you MUST modify this file in order to make the solution work in your environment.

rem In the following command, replace TargetServerName with the name of your target server,
rem and replace the path with the complete path to your SQL AttachDB command file.

OSQL -S <TargetServerName> -i "C:\SQL Scripts\AttachDB.sql" -E
pause
```

---

**NOTE:** The target server name must be included, as well as the complete path to the command file (.sql) created above.

---

3. Run the AttachSQL.bat file on the target server.

## 3.7. Detach the source databases

Using SQL Query Analyzer on the source, use the stored procedure `sp_detach_db` to detach the user-created databases from the source server. The following sample script illustrates the syntax for detaching three databases, named *Sales*, *Accounting*, and *Marketing*, from the source SQL 2000 server.

#### SAMPLE\_DetachDB.sql

```
-- ***SAMPLE*** script for detaching SQL 2000 databases using Query Analyzer or the OSQL command
-- line utility.

-- This sample file is provided as an example only. Because no two
-- environments or configurations are exactly the same, you MUST modify
-- this file in order to make the solution work in your environment.

EXEC sp_detach_db 'Sales', true
EXEC sp_detach_db 'Accounting', true
EXEC sp_detach_db 'Marketing', true
```

## 3.8. Post-upgrade recommendations

The following procedures should be performed after the upgrade is complete.

### 3.8.1. Update statistics

To ensure optimal performance of an upgraded database, Microsoft recommends running `SP_UPDATESTATS` (update statistics) against the upgraded database on the SQL Server 2005 server. Updating statistics using `SP_UPDATESTATS` is described in more detail in the SQL Server 2005 documentation at [http://msdn2.microsoft.com/en-us/library\(d=robot\)/ms173804.aspx](http://msdn2.microsoft.com/en-us/library(d=robot)/ms173804.aspx).

### 3.8.2. Update counts

In versions of SQL Server prior to SQL Server 2005, the values for the table and index row counts and page counts can become incorrect. Therefore, databases that were created on versions prior to SQL Server 2005 may contain incorrect counts. After a database is upgraded to SQL Server 2005, Microsoft recommends that the `DBCC UPDATEUSAGE` be run to correct any invalid counts. This `DBCC` statement corrects the rows, used pages, reserved pages, leaf pages, and data page counts for each partition in a table or index. `DBCC UPDATEUSAGE` is described in more detail in the SQL 2005 documentation at [http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts\\_dbcc\\_24rp.asp](http://msdn.microsoft.com/library/default.asp?url=/library/en-us/tsqlref/ts_dbcc_24rp.asp).

---

### 3.8.3. Change compatibility mode

When SQL 2000 databases are attached to a SQL 2005 server, their compatibility mode is set to 80. In this mode, only features available in SQL 2000 will function properly for that database. To take advantage of new SQL 2005 capabilities, administrators may need to set the database compatibility level to 90. Setting compatibility using `sp_dbcmptlevel` is described in more detail in the SQL 2005 documentation at

[http://msdn2.microsoft.com/en-us/library\(d=robot\)/ms178653.aspx](http://msdn2.microsoft.com/en-us/library(d=robot)/ms178653.aspx).

### 3.9. Move login permissions

There are many possible variations for how client permissions can be moved to the new SQL 2005 database. Due to the large number of possible configurations, instructions for how to configure client connections are out of the scope of this document.

For more information about moving user access permissions to the new SQL 2005 database, see the Microsoft document “HOW TO: Transfer Logins and Passwords Between Instances of SQL Server”, available at

<http://support.microsoft.com/kb/246133/>.

### 3.10. Configure client connectivity

After the databases are attached to the SQL 2005 server, you may remove the source server from the network, change the network name and IP address of the target server to those of the source, and reboot (other DNS changes may also be necessary). Clients should now be able to reconnect to the SQL 2005 server just as they did previously.

Another option would be to configure clients and applications to use the new target server. Configuring users and clients in this manner is beyond the scope of this document.