



[Connect dialog box](#)

[Transfer dialog box](#)



To connect to source and destination servers

1. Under Source Server, in the Server box type the name of the SQL Server to transfer data from. Then, in the Login ID box, type your login identification and in the Password box, type your password for that server.

The source server can be either a Microsoft-based SQL Server for Windows NT or OS/2 or a non-Microsoft-based SQL Server. You must have SELECT permission in the source database. If the source server is a non-Microsoft-based SQL Server and you have not installed Microsoft SQL Object Manager scripts on that server, you must also have CREATE PROCEDURE permission in the source database.

2. Under Destination Server, in the Server box, type the name of the Windows NT-based Microsoft SQL Server to transfer the data to. Then, in the Password box, type the SA password for that server.

Note that it is possible for the source server and the destination server to be the same server, if that server is a Windows NT-based SQL Server. You must be the database owner of the destination database.

3. The ANSI to OEM box is checked if the DB-Library "AutoAnsiToOem" option is set "on" in your system. If this box is checked, TEXT values will be converted during the transfer. Clear the check box if you do not want TEXT values to be converted.
4. Choose the Connect button.

SQL Server must be started on both the source and destination servers.



To transfer data

1. Under Source/Destination, in the From Database box, select the database to transfer data from.
2. In the To Database box, select the database to transfer data to.

Note that the database must already be created and have enough space to contain the objects and data you are transferring. The destination database must be a different database than the source database.

3. To change the location where the scripts are created, in the Script directory box, type a new directory. If the directory is not already created on the computer, SQL Transfer Manager will create it.

Several files are created in the script directory, including a .SCH file, a .TAB file, an .IDX file, a .LOG file, a .DRP file, a .TRG file and any error files that are created as a result of the data transfer. The extensions of these error files are of the form .0 to .FFF. The .LOG file will indicate which table is associated with which error file.

4. To generate the scripts that will be used to perform the data transfer (without actually transferring the data or objects) choose the Export Only button. To view segment and size usage information about tables without actually performing a transfer, you can use this option and then view the information about the tables in the .TAB file that is created in the script directory.

5. Under Objects to Transfer, select the objects to transfer.

- To include all objects in the database, select the All Objects check box.
- To include all objects of a specific type, select the object type's check box.
- To specify objects to transfer, clear the All Objects check box. Then select an object in the Available Objects box and choose the Add button. (You can add multiple objects one at a time, or you can select multiple objects and then choose the Add button.) If you add an object by mistake, you can cancel the addition by selecting the object in the Selected Objects box and then choosing the Remove button.
- To include all objects that depend on the specific object(s) or object type(s) you selected, select the Include Dependencies check box.
- To drop existing objects in the destination database, select the Include Drops check box.
- To transfer segment information to the destination database, select the Include Segments check box. Note that the segments must already be created in the destination database.
- To transfer the data in the source database to the destination database, select the Include Data check box.
- To create unique clustered indexes after the data transfer (by using the WITH SORTED DATA clause), select the Sorted Data check box. If this box is not selected, unique clustered indexes are created prior to data transfer. Nonunique clustered indexes are always created prior to data transfer, whether or not this box is selected. Creating the clustered indexes prior to data transfer is slower during the actual data transfer process, but it eliminates the full table sort that is required if the clustered index is created after the transfer. Nonclustered indexes and triggers are always created after the data is transferred.

If the destination server's sort order does not match that of the source server, using the SORTED DATA clause is likely to fail. If this happens there will be an error message in the .LOG file and after the transfer you must build the indexes yourself without the SORTED DATA clause.

Note that for any clustered index that will be created after the data transfer, the transfer process requires twice the table's size in free space to accommodate the full table sort.

- To add the users from the source database to the destination database, select the All Users check box.

Note that if you have a guest user created in the source database, you must also create a guest user in the destination database.

Overview To add the groups from the source database to the destination database, select the All Groups check box.

Overview To add the login IDs for the users from the source server to the destination server, select the All Logins check box.

Overview To include all the user's permissions from the source database, select the Include Permissions check box.

6. Choose the Transfer button.

Several files are created in the script directory, and the data and objects you specified are transferred. However, if you have chosen the Export Only box, the files will be created in the script directory but no data or objects are actually transferred. You can interrupt the transfer once it has begun by pressing CTRL+C.

Overview

SQL Transfer Manager provides an easy, graphical way to transfer the objects and data from a Microsoft-based SQL Server or a non-Microsoft based SQL Server to a Windows NT-based Microsoft SQL Server. (For example, transferring data from Microsoft SQL Server running on an Intel-based SQL Server to a different processor architecture.)

You can also use this tool to transfer data from a server with one sort order to a server with a different sort order. Note, however, that this tool does not convert extended characters, so you cannot use it for conversion from one code page to another. Also note that if you are transferring data between servers with different case sensitivity, errors will occur if you were not consistent in case when you created your objects and added your data. For example, on a case-insensitive server if you create a table called ABC and then create a view called xyz as `SELECT * FROM abc`, the `CREATE VIEW` statement will work. However, if you transfer the information to a case-sensitive server, the `CREATE VIEW` statement will fail.

To transfer the data, you must have `SELECT` permission in the source database and you must be the database owner of the destination database. If the source server is a non-Microsoft-based SQL Server and you have not installed Microsoft SQL Object Manager scripts on that server, you must also have `CREATE PROCEDURE` permission in the source database.

Before using this tool to transfer data, you must create the destination database and it must be large enough to contain all objects and data that you will transfer. If you are planning to include segments in your transfer, you must also create the segments in the destination database, and there must be sufficient space on the segment(s) to accommodate the tables you will be transferring. If you are not sure of the segment space requirements, you can check them by using the Export Only option in the Transfer dialog box and then reading the generated .TAB file.

.SCH file

The .SCH file contains the database scripts that will be used for transferring the data and objects.

.TAB file

The .TAB file contains summary information about the tables in the database, including their names, segment information, size usage, and the number of rows in each table.

.IDX file

The .IDX file contains the scripts for index creation.

.LOG file

The .LOG file contains all server messages and errors, and informative messages that will help you recover from errors.

.DRP file

If you have selected the Include Drops check box in the Transfer SQL Database dialog box, the .DRP file contains the DROP statements for the objects .

.TRG file

The .TRG file contains the scripts for trigger creation.

