

Welcome – SQLShell™ Accounting Source Code – SQL Server Notes - 10/24/2006

This document contains notes specific to the MS-SQL version of the SQLShell™ Accounting source code for SQL Server databases.

The SQLShell™ accounting source code has been modified to utilize Microsoft SQL Server as the database backend from its original design, which utilized Topspeed data files.

Microsoft SQL Server is a Relational Database Management System (RDBMS) that allows you to create, manage, administer, and deploy databases. It has four main editions: Enterprise, Standard, Developer, and Personal. When you install SQL Server on a Microsoft operating system (OS), the following services are created: MSSQLSERVER, SQLSERVER AGENT, and Distributed Transaction Coordinator (DTC).

Once the installation is complete, SQL Server provides you with a set of administration tools. Enterprise Manager gives you a graphical user interface (GUI) to work with, while Query Analyzer gives you the ability to run T-SQL queries. SQL Profiler allows you to analyze data and queries in real time or from a flat file.

To minimize changes to the source and allow for maximum compatibility with other relational database servers via ODBC (ODBC version also available) all code controlling referential integrity, data validation (constraints), processes, etc. have been left in the applications rather than migrated to the RDBMS. Clarion developers who are productive in Clarion but may lack RDBMS experience should find the applications much easier to modify to their own requirements as a result. Migration of various code from the applications to the back end RDBMS may be performed by the developer to suit their specific needs at any time, in whole or in part without effect on the applications assuming the changes are properly implemented. The suite of applications will perform quite well running against MS-SQL Server or for that matter any RDBMS via ODBC.

Some Specifics about the code changes

A new dictionary was created for MSSQL and all applications have been updated to use this new dictionary. A single script has been created that will add all required tables, indexes, etc. to the new database once created on the SQL Server. See the script header for installation notes.

Two database backup files have been created and are included with the applications. One is a blank database named DukeSQL.bak and another is a database with some sample data and is named DukeSQLDemo.bak, both of which can simply be restored to an existing SQL Server installation. See SQL server notes below for instructions on how to restore a backup if needed.

A conversion program has been created and source is included to allow the developer to use the program as a data migration tool for existing installations utilizing the Topspeed data files. The developer will need to verify the column definitions of the existing Topspeed files match their own since they may have been modified and adjust the code to their specific needs. Additionally, the source must be changed to either point to the target SQL Server and specify connection information or could be modified with ease to get this information from an INI file so that the program could be distributed as is to end users along with an INI file for the end user to modify.

- All STRING columns changed to CSTRING (original Length + 1)
- Dates modifications:
 - If only date column was used in the TPS version, then the date was converted from datename LONG to datename with a type of DATE in the dictionary, but DATETIME in the SQL create script. The Clarion RTL is smart enough to handle just the date portion.
- IF both a date and time was used then the following is used to convert instead.

```

DateName    STRING(8)
DateName_GRP  GROUP,Over(DateName)
Date_Date    ! Date
DATE_Time    ! Time
            END

```

- A SysID column (type LONG) has been added to all tables except the Company table. This column is automatically incremented by the SQL Server using IDENTITY() and was added to insure that each table has a unique value as required by all RDBMS platforms.
- The COMPANY table remains as a TPS (Topspeed) data file that should be loaded to the default directory where the applications run from and stores the company names and server connection information, allowing the end user to select and connect to various databases to support multiple companies via one installation.
- All browses, processes, report procedures, etc. have had all required columns that are utilized in a record filter added to the Hot Field list (added the BIND() attribute). Additionally, each has had the unique SysID column added as an additional sort columns entry.
- All procedures have been examined and minor code changes made as required to insure compatibility with SQL Server.
- Various code throughout the applications utilized dimensioned fields (arrays) for handling data. Since the MS-SQL Accelerator driver (nor any of the SLQ drivers including ODBC for that matter) support the DIM() attribute, all arrays have been modified to basically create a new column for each dimension of the array, then a group over has been added. This allows the original method of accessing the array to be used without modification.

Other items of interest...

All applications were compiled and tested with Clarion 5.5H. All applications should compile and run using any release of Clarion 5.5 though the latest release is of course recommended. The applications will also open and compile properly under Clarion 6x, of course the latest release is recommended. If using Clarion 6.x, it is *strongly* recommended that the Global Cooperative Threading Extension be used in all apps as the applications have not been modified and tested to be 100% thread safe using the new Clarion threading model. In general, you should discover that the applications function better if compiled using Clarion 6.3x, the ABC libraries have greatly improved since Clarion 5.5x releases.

After compiling the applications, you will need to open the switchboard app at least once to configure your default SQL connection settings for your installation. While all apps will run properly, some procedures may return a driver error (usually from within a couple of process procedures) stating that an invalid driver action has been performed if compiled using Clarion 5.5. There is no actual error, but at least one record is required for some operations. Clarion 6.2 or greater will not report this error and will simply continue. Apps compiled under Clarion 5.5x will also continue to function properly despite the error message. Once a record has been added, this will no longer appear.

Developers new to MS SQL Server will need to become acquainted with the aspects of installing, configuring and managing databases along with developing a basic skill set for working with tables, views, stored procedures, etc. To get the basic installation up and running, some notes have been provided below. The applications will run correctly with MS SQL Server v7 or newer. The notes below assume a SQL 2000 installation. Detailed information on all versions is available via books online or at www.microsoft.com/sql

Installing SQL Server 2000 (SQL Server v7 is nearly identical - visit www.microsoft.com/sql for SQL 2005 quick start instructions)

To Install SQL Server 2000 Basic Local Installation

1. Insert the Microsoft SQL Server 2000 compact disc in your CD-ROM drive (if the compact disc does not run automatically, double-click Autorun.exe in the root directory of the compact disc), select SQL Server 2000 Components, and then select Install Database Server. Setup prepares the SQL Server Installation Wizard. At the Welcome page, click Next.

2. In the Computer Name dialog box, Local Computer is the default option, and the local computer name appears in the text box. Click Next.

3. In the Installation Selection dialog box, click Create a new instance of SQL Server, or install Client Tools, and then click Next. Follow directions on the User Information, Software License Agreement and related pages. In the Installation Definition dialog box, click Server and Client Tools, and then click Next.

4. In the Instance Name dialog box, if the Default check box is available, you can install either the default or a named instance. If the Default check box is not available, a default instance has already been installed, and you can install only a named instance. • To install the default instance, click to select the Default check box, and then click Next.

(To install a named instance, click to clear the Default check box, type a new named instance in the Instance Name box, and then click Next.)

5. In the Setup Type dialog box, click Typical or Minimum, and then click Next.

6. In the Service Accounts dialog box, accept the default settings, type your domain password, and then click Next. In the Authentication Mode dialog box, accept the default setting, and then click Next. When you finish specifying options, click Next in the Start Copying Files dialog box.

7. In the Choose Licensing Mode dialog box, make selections according to your license agreement, and then click Continue to begin the installation. In the Setup Complete dialog box, click Yes, I want to restart my computer now, and then click Finish.

To Install Client Tools Only for SQL Server 2000

1. Insert the Microsoft SQL Server 2000 compact disc in your CD-ROM drive (if the compact disc does not run automatically, double-click Autorun.exe in the root directory of the compact disc), select SQL Server 2000 Components, select Install Database Server, and then click Next at the Welcome page of the SQL Server Installation Wizard.
2. In Computer Name dialog box, Local Computer is the default option, and the local computer name appears in the edit box. Click Next.
3. In the Installation Selection dialog box, click Create a new instance of SQL Server, or install Client Tools, and then click Next.
4. Follow the directions on the User Information, Software License Agreement, and related pages.
5. In the Installation Definition dialog box, click Client tools only, and then click Next.
6. In the Select Components dialog box, accept the defaults or select the components you want, and then click Next. You can select an item in the Components list, such as Management Tools, and then select items from the related Sub-Components list, such as Enterprise Manager. Click to select items that you want to install, and click to clear the check box for the items you do not want to install. For information about each component, select the item, and view the Description box.
7. In the Start Copying Files dialog box, click Next to complete the installation of the client tools.

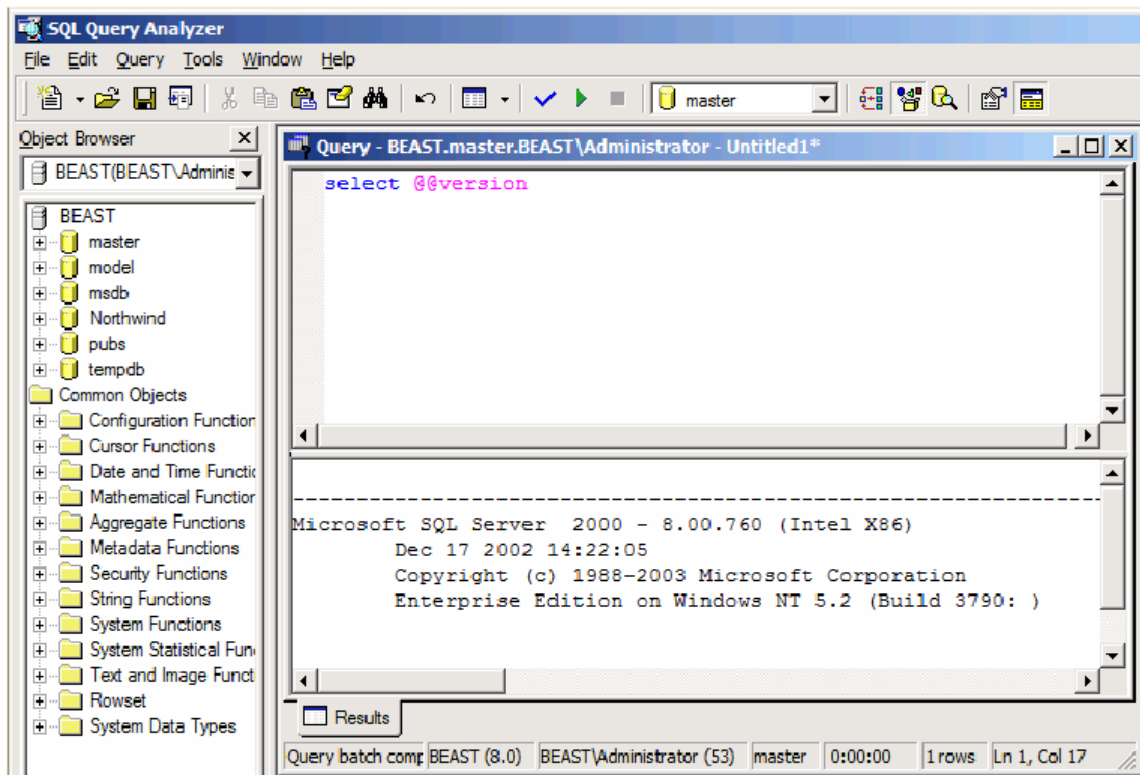
To Install Connectivity Only for SQL Server 2000

1. Insert the Microsoft SQL Server 2000 compact disc into your CD-ROM drive (if the compact disc does not run automatically, double-click Autorun.exe in the root directory of the compact disc), and then select SQL Server 2000 Components.
2. Select Install Database Server. Setup prepares the SQL Server Installation Wizard. At the Welcome page, click Next.
3. In the Computer Name dialog box, Local Computer is the default option, and the local computer name appears in the text box. Click Next.
4. In the Installation Selection dialog box, click Create a new instance of SQL Server, or install Client Tools, and then click Next.
5. Follow the directions on the User Information, Software License Agreement and related pages.
6. In the Installation Definition dialog box, click Connectivity Only, and then click Next.
7. In the Start Copying Files dialog box, click Next to complete the installation.

Common Tasks You Will Perform

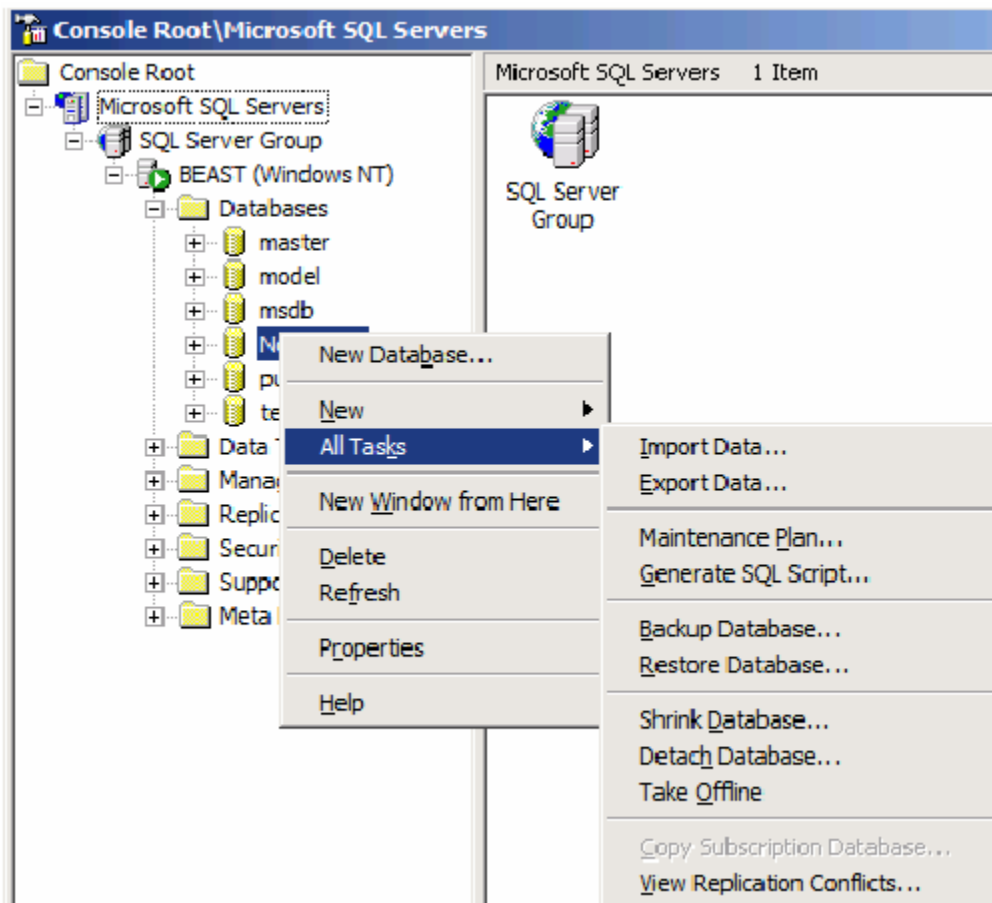
Run a query in Query Analyzer:

1. Click Start | Programs | Microsoft SQL Server | Query Analyzer.
2. Choose your Authentication type: Windows Authentication or SQL Server Authentication.
3. Type `select @@version` and click Go.



Back up a Database

1. Click Start | Programs | Microsoft SQL Server | Enterprise Manager.
2. Right-click a database and choose All Tasks | Backup Database.
3. Click Add to add a destination where your backup will reside.
4. Click the ellipsis under Filename and type the name of your backup.
5. Click OK and Finish.



Restore a backup

1. Click Start | Programs | Microsoft SQL Server | Enterprise Manager.
2. Right-click a database and choose All Tasks | Restore Database.
3. Choose the backup you want to restore and click OK.

Additional SQL Server Resources

[Clarion Magazine](#)

Online magazine for Clarion developers with many articles pertaining to SQL Server and Clarion.

[Microsoft SQL Server home page](#)

Microsoft's designated home Web site for the SQL Server.

[The Database Journal](#)

The online knowledge center for database professionals.

[The Professional Association for SQL Server](#)

This non-profit organization is dedicated to the advancement of SQL Server.

[SQL Server Central](#)

A community Web site that provides SQL Server articles, scripts, and forums to help your professional development as a database administrator or user of SQL Server.

Books...

[Professional SQL Server Reporting Services](#)

By Paul Turley, Todd Bryant, James Counihan, George McKee, and Dave DuVarney. Wrox, 2004, ISBN: 0764568787.

[Microsoft SQL Server 2000 Bible with CD-ROM](#)By Paul Nielsen. John Wiley & Sons, 2002, Book and CD-ROM edition, ISBN: 0764549359.

[Microsoft SQL Server 2000 Database Administrator's Guidebook](#)By Carl H. Speshock. Prentice Hall PTR, 2001, Book and CD-ROM edition, ISBN: 0130614300.

[Microsoft SQL Server 2000 Administrator's Pocket Consultant](#)By William R. Stanek. Microsoft Press, 2000, ISBN: 0735611297.

[Microsoft SQL Server 2000 Database Administrator's Guidebook](#)By Carl H. Speshock. Prentice Hall PTR, 2001, ISBN: 0130614300.

Other resources...

[Perform a secure SQL Server installation](#)

[Step-By-Step: An introduction to SQL Server Profiler](#)

[TechRepublic Tutorial: Creating new databases in SQL Server 2000](#)

[Set up SQL Mail to send SQL Server alerts](#)

[Take advantage of the SQL Server 2000 security tools](#)

Get the apps up and running FAST – A Quick Start Guide

1. After completing the **SQLShell™** accounting source code installation, open and compile all applications.
2. Assuming you have successfully installed SQL Server 2000 on a machine, restore the included DukeSQL.bak and DukeSQLDemo.bak databases to your SQL Server and remove the DukeSQL user from both databases all via Enterprise Manager.
3. Using Enterprise Manager, go to security for your server and add a user named DukeSQL, make the password DukeSQL, then grant it access to the DukeSQLDemo and the DukeSQL database, then give it owner permissions and of course the default of public may be left alone on both databases.
4. Modify the included 'DukeSQL.ini' file in the login settings section to point to your server and the database of your choice, the DukeSQLDemo or the DukeSQL.
5. Start the Switchboard application, click on Set Default SQL Login button and verify all looks correct.
6. Now launch any application from the switchboard, you are now up and running.

Errata

Please refer to the included documentation for set up and usage of the accounting applications.

DISCLAIMER

The **SQLShell™** Application Shells are licensed "as is" and we cannot and do not guarantee you uninterrupted service or protection from errors or lost data. You assume responsibility for selecting and modifying software to achieve your intended results. Comsoft7 does not warrant that the functions contained in the programs will meet your requirements or that the operation of the programs will be uninterrupted or error free.

In no event will Comsoft7 be liable to you for any lost profits, lost savings, or other incidental or consequential damages arising out of the use or inability to use any program or for any claims by any third party.

See our license agreement for additional information.