

Setting up Databases on a Microsoft SQL 7.0 Server

Overview of the Installation Process

To set up *GoldMine* properly, you must follow a sequence of steps to install *GoldMine*'s program files, and the other programs required by *GoldMine* Enterprise Edition. The following list of steps shows the procedure that you must perform for SQL Server 7.0. This guide provides most of the necessary information, and references will direct you to any needed outside documentation.

1. Install *GoldMine* on your network drive—see “Installing *GoldMine* on a Network” in the *GoldMine Reference Manual*. Your system must run *GoldMine* version 4.00.9626.

★ **Before you start:**

1. For best results from your server when running *GoldMine*, use the recommended versions of the following Microsoft products:
 - MS SQL 7.0 Service Pack 1 or later
 - NT Server 4.0 Service Pack 5 or later
2. Check that your system is running *GoldMine* version 4.00.9626 or greater. To display information about the version from within *GoldMine*, from the Main Menu, select **H**elp|**A**bout.

If your *GoldMine* version is less than 4.00.9626, *you must update before continuing*. To update the version on a system with Internet access: from *GoldMine*'s Main Menu, select **H**elp|**N**et Update.

2. Install your SQL server software—see your SQL Server documentation. You do not have to install both *GoldMine* and the SQL software on the same server.

3. Create a database on the SQL server.
4. Modify your SQL 7.0 configuration for use with *GoldMine*.*
5. Set up a BDE alias.
6. Create SQL contact set(s) with *GoldMine*.
7. Create user login(s) for the SQL Server for *either* each *GoldMine* user, *or* for all *GoldMine* users.
8. Modify user properties in *GoldMine* to link *GoldMine* users with SQL Server 7.0 users. If you have not yet created *GoldMine* users, you must do so before continuing—see “Setting up Security and Access Rights” in the *GoldMine Reference Manual*.
9. Install SQL Server client utilities on workstations.

The following documentation provides details for steps 3–9. For the remaining steps, see the resources listed in the step.

Creating an MS SQL 7.0 Database

For each *GoldMine* contact set that you wish to rehost to SQL Server, you must create a database in the **SQL Enterprise Manager**. Global *GoldMine* databases, such as the **Calendar** and **Mailbox**, should only be rehosted once to one MS SQL database. If you implement only one contact set, all *GoldMine* files can be rehosted to the same SQL database. For details on rehosting, see “Rehosting *GoldMine* Data to SQL Server 7.0” on page ###.

To create a database in SQL Server 7.0 for use with *GoldMine*:

1. To start the Enterprise Manager, from the Windows’ task bar, select **Start|Programs|Microsoft SQL Server 7.0|Enterprise Manager**.
2. In the left pane, search the list of SQL servers to locate the server that will receive the rehosted database(s).

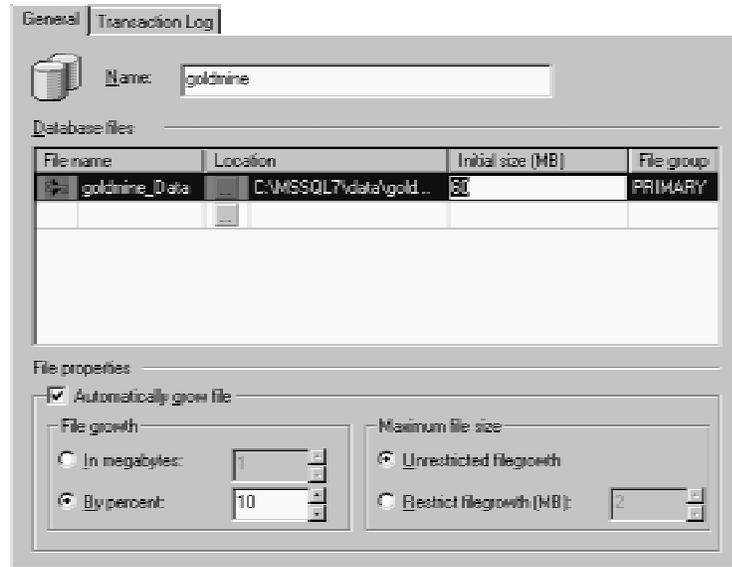
If you find the server you want, highlight the server. Go to step 4.

If not, continue with step 3.

* The *GoldMine Enterprise Administrator’s Guide* provides the procedures and recommendations for the most essential settings, but comprehensive tuning of your database server is beyond the scope of this guide. Further customization of your MS SQL 7.0 configuration should be performed by either a database administrator or certified MS SQL technician.

3. Right-click on **SQL Server Group** to display the local menu. Select **New SQL Server Registration**. The **Server Registration Wizard** appears. You can select a server from a list of candidate MS SQL servers. For details on registering a server, see your MS SQL Server 7.0 documentation. *You must register your server before proceeding to the next step.*
4. To establish a connection with the selected server, click  next to the server entry.
5. Right-click on the **databases** folder to display the local menu, then select **New Database**. The **Database Properties** dialog box appears. By default, the **Enterprise Manager** displays the **General** tab.

General tab



6. Type a **Name** for the database. Since you must refer to this name at a later time in the configuration process, be sure to keep a record of the name.
7. Check that the **Location** field contains the drive and directory in which you want to create the database file on the server. *Unless you plan to create the database on a different physical drive than where MS SQL resides, do not change this setting from the default.*
8. Set the **Initial size (MB)** for your SQL database. If you plan to rehost a preexisting dBase database, enter a value that is approximately 15% larger than the size of your total dBase data.



By default, SQL Server 7.0 automatically expands database size, which directs SQL Server to monitor the free space in a database and expand the size as needed. Unless special needs or circumstances require otherwise, leave the default setting to allow SQL Server to manage the size of your databases and thereby reduce the amount of database maintenance required.



Unless you are experienced with database management and fully understand the implications, do not modify any other default settings when creating a database.

9. Select the **Transaction Log** tab. MS SQL Server generates a transaction log for each hosted database to record each change made to that database. Check that the **Location** field contains the drive and directory in which you want to create the transaction log file on the server machine.

Creating a database may be a somewhat lengthy process.

10. When the database is created, select the database entry, then right-click to display the local menu. Select **Properties**. The **Database Properties** dialog box appears.
11. Select the **Options** tab.
12. Select **Truncate log on checkpoint** and **Select into/bulk Copy**.
13. Select **OK**.

Your database has been created. Repeat steps 4–13 for each database you must create. Keep a record of the name given to each database, as you need this information to create an alias in the Borland Database Engine (BDE). If you plan to rehost only one *GoldMine* contact database, you must create only one SQL database.

Modifying your MS SQL Server Configuration to Run GoldMine

Please read this section carefully before performing the procedure. *Only technically experienced individuals should attempt making changes to an MS SQL server. Any mistakes made in following these instructions can render your MS SQL server unusable.* If you are not technically qualified, ask your database administrator or an MS SQL-certified Authorized GoldMine Solutions Partner perform these steps.

To run *GoldMine* on an MS SQL server, there must be:

- Sufficient number of user connections to accommodate the total number of *GoldMine* users
- and*
- Minimum amount of memory must be available to support each user connection

User Connections

GoldMine opens a *user connection* to access a table on the MS SQL server. The maximum number of user connections allowed by the server is specified in the MS SQL server configuration settings. The server must allow two user connections per *GoldMine* user accessing the data. For example, if you have 20 *GoldMine* users, your *user connections* must be at least 40.

Allocating Memory

In addition to setting the number of user connections, you must allocate sufficient memory to support each user connection. The memory available for SQL Server is limited both by the settings specified and by the amount of RAM on the server. For each user connection, you must allocate 37 KB. You can configure MS SQL 7.0 *either* to either dynamically allocate memory up to a specified maximum, *or* you can allocate a fixed amount of memory for MS SQL 7.0 to use.



The 37-KB memory requirement is the *minimum requirement* to maintain the connections desired. However, this is *not* enough memory to run *GoldMine* optimally.

GoldMine Software recommends a *minimum* of 128 MB of RAM on the server. The exact amount of memory you need for your server will depend on the number of users accessing the data and the number of contact records in *GoldMine*, as well as activities run by the server in addition to *GoldMine*. Allocating additional memory through the server configuration enhances overall MS SQL performance. For assistance in determining an adequate amount of memory for your server, contact an Authorized GoldMine Solutions Partner certified in MS SQL.

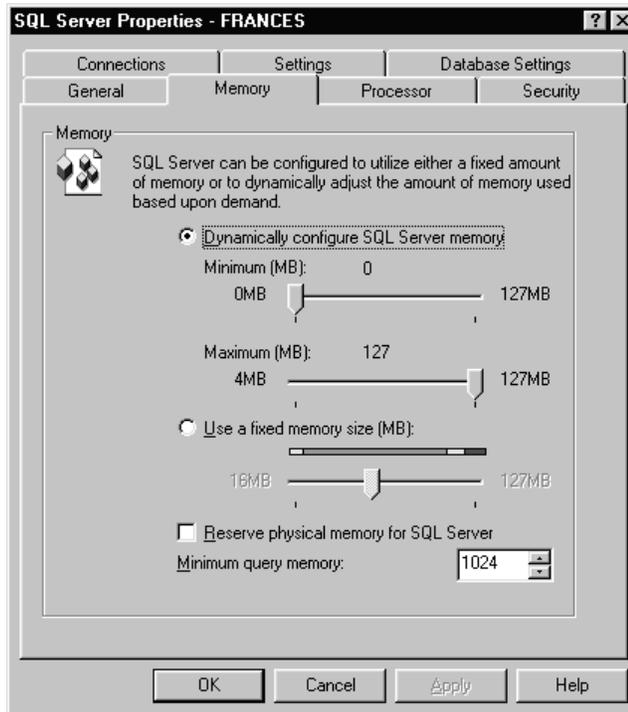


The following modifications are *not necessary* unless the settings have already been modified from their default. It is recommended that you leave these settings as they were when SQL Server was installed.

To change the user connections and memory allocation from SQL Enterprise Manager:

1. Click  next to your server name to establish a connection to the server.
2. Right-click on the server name to display the local menu. Select **Properties**. The SQL Server Properties dialog box appears. By default, the SQL Enterprise Manager displays the **General** tab.
3. Select the **Memory** tab.

SQL Server Properties box



The memory tab contains the following options:

- **Dynamically configure SQL Server memory** (recommended): specifies a minimum and maximum amount of memory in megabytes that MS SQL 7.0 can allocate for its memory needs. To change the minimum and maximum settings, drag the sliding memory indicator to the desired level of memory allocation. Selecting this option will reduce or eliminate memory allocation errors because MS SQL Server can determine and allocate memory demanded by an operation.
- **Use a fixed memory size**: specifies a fixed amount of memory for MS SQL 7.0 to use for memory. If you select this option, be sure to allocate sufficient memory for the user connections (37 KB per user connection). To change the allocation, drag the sliding memory indicator to the desired level of memory allocation. Selecting this option runs the risk of encountering memory allocation errors.*

4. Select the **Connections** tab.

5. For **Maximum Concurrent User Connections**, specify 0 to allow MS SQL Server to automatically decide how many user connections are needed (no maximum). If you must limit the number of concurrent *GoldMine* users, specify the maximum number of user connections in this field. The server must allow two user connections per *GoldMine* user accessing the data, so if you have 20 *GoldMine* users, your *user connections* must be at least 40.

6. Select **OK**.

7. Right-click on your server to display the local menu, then select **Stop**.

8. Right-click on your server to display the local menu. Select **Start**.

* If the available memory does not meet the 37-KB/user-connection requirement, an error message appears to indicate that the server cannot run with the current memory allocation. Increase the size of the memory allocation until the message does not appear.

Configuring the BDE Administrator for use with SQL Server 7.0

The next step in setting up *GoldMine* for a Microsoft SQL server is to create the database alias in the BDE. *GoldMine* works with the BDE to communicate with the servers. Creating an alias provides *GoldMine* with a means to identify the SQL database.



Each workstation that will access the *GoldMine* SQL data must use the same, shared copy of the Borland Database Engine. For details on administering your BDE, see FactsBack #500, which you can download from the GoldMine Software Web site at <http://www.goldminesw.com>.

To establish a BDE connection between *GoldMine* and MS SQL Server 7.0:

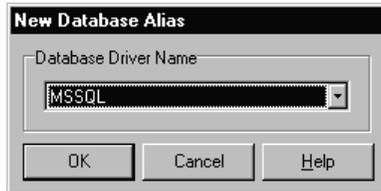
1. To start the **BDE Administrator**, run BDEADMIN.EXE from *either*:

- GoldMine/Setup/BDEShare/ folder
- or*
- Windows' Control Panel (Start|Settings|Control Panel)

If you cannot locate the **BDE Administrator**, your BDE may be incorrectly configured. For details, see FactsBack document #500 as described above.

2. From the Main Menu, select **Object|New**.

*New Database
Alias
dialog box*



3. From the **Database Driver Name** drop-down list, select **MSSQL**. Select **OK**.

4. In the left pane of the **BDE Administrator**, type a name for the alias, then press **ENTER**. To indicate that changes have been made,  appears next to the alias name. *Do not save changes at this time*, but continue with "Required Edits to the Default Entry for MS SQL."

Required Edits to the Default Entry for MS SQL

1. In the Definition folder, type the DATABASE NAME created in Microsoft SQL Enterprise Manager.

Defining the BDE setup for MS SQL

Definition	
Type	MSSQL
APPLICATION NAME	
BATCH COUNT	200
BLOB EDIT LOGGING	
BLOB SIZE	64
BLOBS TO CACHE	256
DATABASE NAME	IGMDATABASE
DATE MODE	0
ENABLE BCD	FALSE
ENABLE SCHEMA CACHE	FALSE
HOST NAME	
LANGDRIVER	
MAXDBPROCESSES	31
MAXQUERYTIME	900
MAXROWS	-1
NATIONAL LANG	NAME
OPEN MODE	READ/WRITE
SCHEMA CACHE DIR	
SCHEMA CACHE SIZE	8
SCHEMA CACHE TIME	-1
SERVER NAME	gmsserver
SQLPASSTHRU MODE	SHARED/AUTO COMMIT
SQLSRV MODE	
TDS PACKET SIZE	4096
USER NAME	sa

Edit the fields as shown by following steps 1-6

2. Change **SERVER NAME** to the computer name of your MS SQL server.

3. Change **USER NAME** to the user name that will log in to MS SQL, as shown in the figure. This login will be used *only* if no login is entered in the **Logon name** field in the **Access** tab of *GoldMine's [username] Properties* window.

4. In the **BLOB SIZE** field, type 64.

5. You can increase the **MAX QUERY TIME** if the size of your contact set is very large. This value limits the number of seconds that a client (workstation) will wait after requesting data from the server before timing out. Setting the entry to 300 (seconds) is usually sufficient, even for a database of several hundred thousand contacts.

6. Type a **BLOBS TO CACHE** value representing the approximate number of activities your users will view in *GoldMine's Activity List*. Acceptable values include 256, 512, and 1024.



If the **BLOBS TO CACHE** value is set too low, *GoldMine* may display the following error:

Invalid Blob Handle in Record Buffer

To correct this error, increase the setting. *Only increase this setting if the error occurs.* Raising the setting requires more memory on the workstation running *GoldMine*.

7. To apply the changes, *either* click  in the upper left-hand corner, *or*, from the Main Menu, select **Object|Apply**. A prompt asks:

Save all edits to [*database alias*]?

8. To set up the BDE connection to the new database alias, select OK.

9. Select the **Configuration** tab.

10. To expand the tree, select **Configuration|Drivers|Native|MSSQL**.

11. In the right pane, increase the **TIMEOUT** value to 7200.

12. To apply the changes, *either* click  *or*, from the Main Menu, select **Object|Apply**. A prompt asks:

Save all edits to [*database alias*]?

13. Select OK.

You have finished creating the alias. Be sure to test the connection because you must be able to connect successfully to continue the rehosting process.

To test the connection, return to the **Databases** tab, then click  next to the new alias.

If you connect successfully, a green box will appear around the alias icon.

If not, see the “Troubleshooting Guide” on page 46.

Rehosting GoldMine Data

Once the BDE is configured to access an SQL database you can rehost *GoldMine* data. Close the **BDE Administrator**, then start *GoldMine*.

Using **GoldMine's Database Wizard**, you can *either*:

- Create a database for new data
- or*
- **Rehost** (copy and convert) an existing database existing onto a different server

Creating a Database

You can create a blank database structure to store the *GoldMine* table structure. Once created, you can later manually enter or import data into the database.

Rehosting a Database

You can rehost an existing *GoldMine* contact database or other *GoldMine* files onto the MS SQL Server. This process does *not* affect the original database, but instead, copies and converts a file from one format type, such as dBASE, to another, such as SQL.

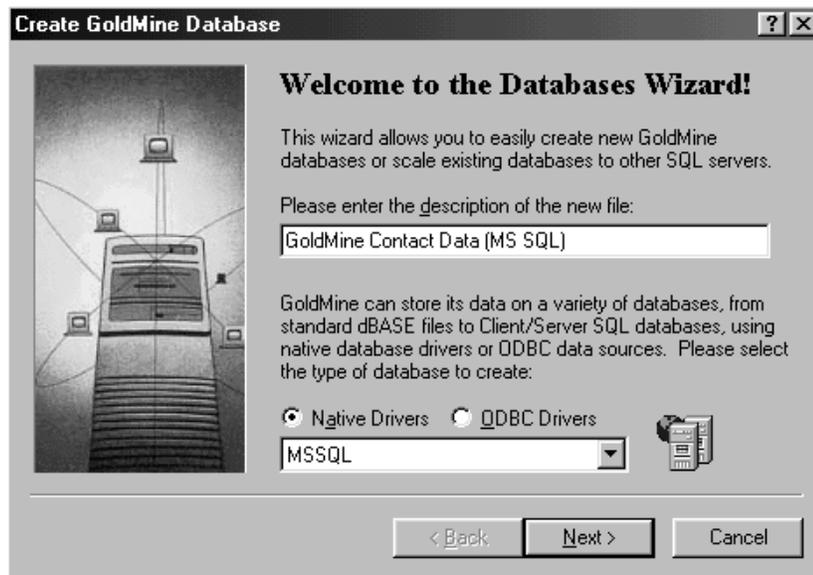
The **Create GoldMine Database Wizard** can rehost either one or two databases:

- **Contact set database:** contains all of the contact tables that store contact information, such as names, addresses, phone numbers, and other contact-related information. When you rehost a contact set, *GoldMine* adds a contact set record to the **Contact Files** dialog box.
- **Other GoldMine files:** includes the **Calendar**, forms, **InfoCenter™**, etc. When you rehost these files, *GoldMine* does *not* add records to the **Contact Files** dialog box. Instead, the **GoldDir** line of *GoldMine's* GM.INI file is modified to set *GoldMine* to use the new SQL database when accessing these files.

To create a database in *GoldMine*:

1. From the Main Menu, select **File|Open File**, then select **New** from the **Contact Files** dialog box.

*Create GoldMine
Database Wizard*

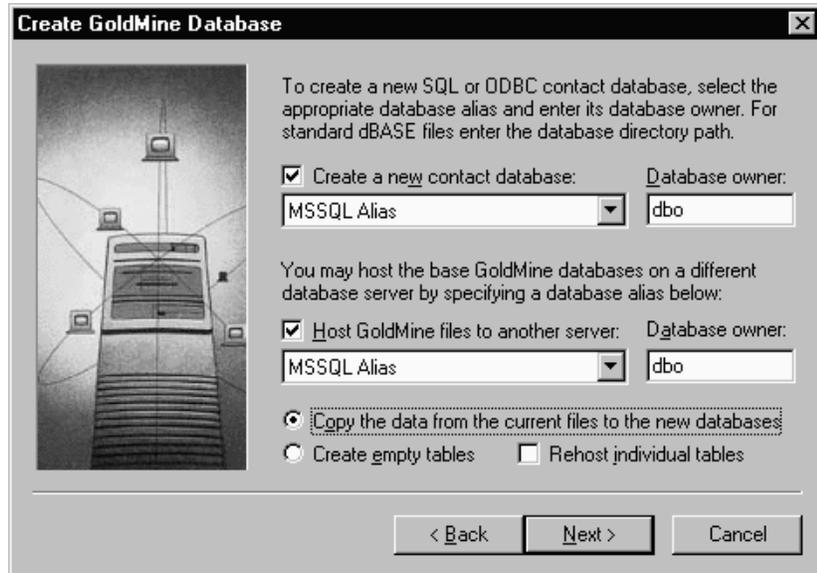


2. Below **Please enter the description of the new file**, type a descriptive title for the contact set. When creating a database or rehosting a contact set, *GoldMine* will use this title when listing the file in the **Contact Files** dialog box. However, when you *only* rehost other *GoldMine* files such as the **Calendar**, **InfoCenter™**, etc., this field can be left blank. *GoldMine* does *not* list these files in the **Contact Files** dialog box.
3. Designate a driver and database type as follows:
 - a. Select **Native Drivers**.
 - b. Select **MSSQL** from the drop-down list.
4. When done, select **Next>**. The second dialog box of the **Create GoldMine Database Wizard** appears. Continue with “Setting Database Options.”

Setting Database Options

Once you select **Next>** from the first dialog box of the **Create GoldMine Database Wizard**, the second dialog box appears. The following figure shows the second dialog box of the wizard that appears when you select an SQL database type, such as **MSSQL**.

*Create
GoldMine
Database
Wizard*



To create or rehost a database on SQL server 7.0:

1. To **Create a new contact database**, select the BDE alias you created earlier in this document from the drop-down list.
2. To rehost your *GoldMine* files, including the **Calendar**, **InfoCenter™**, forms, and other tables, in addition to your contact data, specify a location by selecting any listed database alias from the **Host GoldMine files to another server** drop-down list.

If you are only rehosting a contact set, uncheck the **Host GoldMine files to another server**.

4. Specify the **Database owner**. This should be the user in SQL that you will use to rehost your database(s). Do *not* edit this field unless you have a specific reason, and have created an administrative user in SQL Server for this purpose. If you have not modified your SQL server's security, leave the **Database owner** at **dbo**.



The first time that you rehost databases from dBASE to an SQL server, you will want to select *both* the contact set database *and* the global *GoldMine* data files so that all data is located on the SQL server. Both *GoldMine* files and contact set files can exist in the same SQL database.

However, you should only rehost the *GoldMine* files once. If you rehost more than one contact database, do *not* check Host *GoldMine* files to another server during each additional rehost. You do *not* have to rehost both at the same time.

4. Specify whether you want to *either* copy data from the currently open contact set to the newly created database, *or* create an empty database structure to receive data at a later time.

a. Select one of the following options:

- **Copy the data from the current files to the new databases:** Copies data from the currently open contact set. If necessary, *GoldMine* will convert the data to the format appropriate for the database type specified on the first dialog box of the Create *GoldMine* Database Wizard.

or

- **Create empty tables:** Creates a database structure that can receive data at a later time. *Never* select this option if you select Host *GoldMine* files to another server.

b. To specify one or more tables for *GoldMine* to copy, select **Rehost individual tables**. *GoldMine* displays the **Rehost Selected Tables** dialog box, from which you can select one or more tables to copy. Check this option only if you want to specify the tables to be copied. For example, you might want to copy the table(s) needed to complete rehosting if the process was interrupted.

The individual tables will *either* be empty, *or* contain data from the currently open contact file or root *GoldMine* data, depending on the option selected in step 4a.

5. When done, select Next>. To gain access to the database server, *GoldMine* may prompt you for the SQL login.

6. To enable access to the database server, type `sa` for the **Username**. Bypass **Password**, and select **OK**.* This login request will *not* accept your *GoldMine* username and password. Although *GoldMine* is prompting the user for an entry, this password is required by the SQL database, not by *GoldMine*.

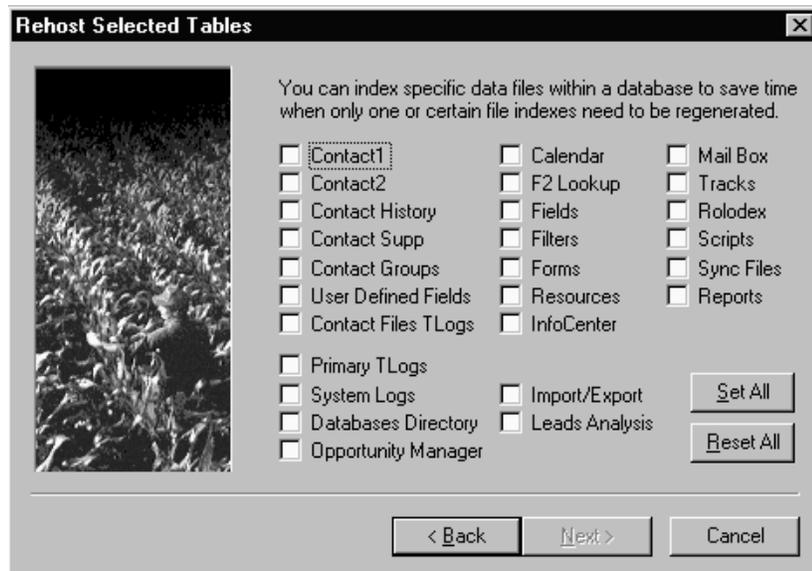
If you selected *either Copy the data from the current files to the new databases or Create empty tables* and did not select **Rehost individual tables**, the Database Access dialog box appears. Go to “Setting Access Options” on page 27.

If you selected **Rehost individual tables**, the Rehost Selected Tables dialog box appears. Continue with “Selecting Individual Tables to Rehost.”

Selecting Individual Tables to Rehost

If you select **Rehost individual tables** from the second dialog box of the Create GoldMine Database Wizard, then select **Next>**, the Rehost Selected Tables dialog box appears.

Rehost Selected Tables dialog box



* If you cannot log in using the username `sa` and without a password, the SQL Server's security has been modified. Check with your database administrator for the username and password for an SQL Server user that has the security rights to create tables.

1. Check each box corresponding to the data file(s) that you want to select.

Selecting individual tables for rehosting is useful when you want to:

- Restore a table from a dBASE backup
- Transfer a table to another *GoldMine* system
- Restart *from the point of failure* after a failed attempt to rehost

2. When done, select Next>. Go to “Starting to Create Database Files” on page 29.

Setting Access Options

If you select *either* **C**opy the data from the current files to the new databases *or* **C**reate empty tables from the second dialog box of the Create GoldMine Database Wizard, then select Next>, the Database Access dialog box appears.

*Database
Access
dialog box*



To grant access to the database:

1. To grant access to specified users, select a user or user group from the **A**llow **a**ccess to: drop-down list. By default, *GoldMine* will grant access to **(public)**. If unsure, leave the default setting.
2. To specify a *database* that *GoldMine* can use when associating transfer set data, type a unique **C**ontact set **c**ode value for the *database*. This code is necessary only if you plan to synchronize multiple contact sets.



For details about contact set codes, see “Setting Database Options” in the *GoldMine Reference Manual*.

3. To allow *only* the currently running *GoldMine* (as determined by serial number) to open the database, select **Allow database access only from this GoldMine installation**. A *GoldMine* system with a different serial number cannot open the database.

4. Select **Next>**. Continue with “Starting to Create Database Files.”

Starting to Create Database Files

Once you have finished defining options for the database, the **Creating Database Files** dialog box appears.

When ready to start creating the database, select **Finish**. If you are copying a large number of records as part of the process, creating a database can be a relatively time-consuming process.

To stop the process at any time, select **Cancel**.

GoldMine displays the **Creating GoldMine Database Files** status window to provide status information during the creation of the database.

If you have rehosted the global *GoldMine* databases, a window will prompt you to make them default. Select **Yes** to instruct *GoldMine* to open the newly rehosted SQL databases whenever the program is run.

When done creating the database, *GoldMine* closes the **Creating Databases Files** status window.

Creating MS SQL Server 7.0 Users

To access the MS SQL Server, each user must have a login to the database. You can use one of the following methods to create logins for use with *GoldMine*:

- ***Create a single MS SQL Server login for all GoldMine users:*** simpler method that minimizes administration within MS SQL Server, but may reduce the amount of manageability and accountability. Using only one login will not allow an administrator to separately track changes to the database by different users.
- ***Create an MS SQL login for each user:*** requires more initial administration, but allows greater flexibility in security and accountability. Using this method, users can be assigned unique permissions, and their changes can be tracked on a per user basis.

Either approach will work with *GoldMine*. Depending on the demands and requirements of your environment, such as the number of *GoldMine* users, the

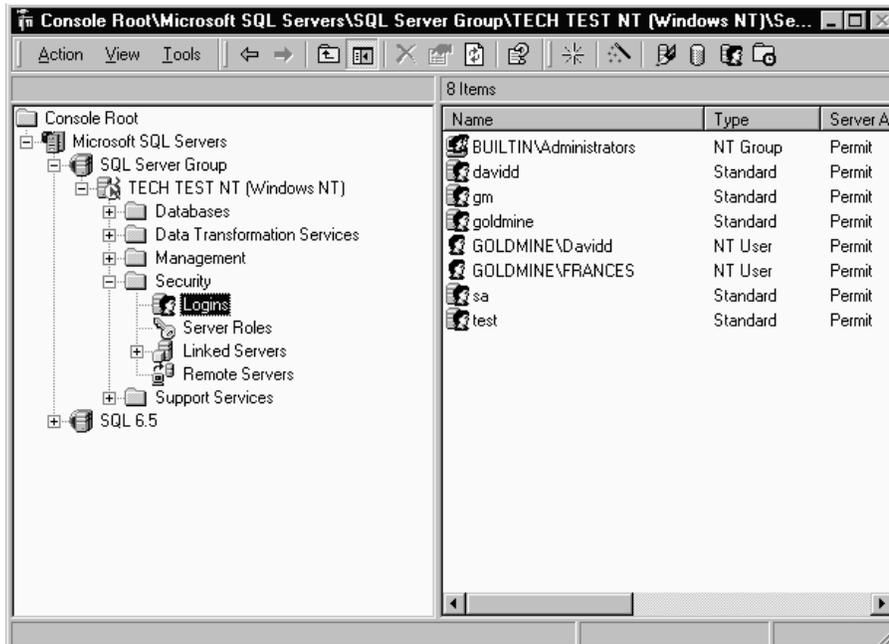
required security for those users, and the amount of time allocated for the rehosting process, one approach will be more advantageous than the other. Your database administrator can determine which method will work best for your environment.

Creating Logins

To create logins:

1. To start the SQL Enterprise Manager, from the Windows' task bar, select Start|Programs|Microsoft SQL Server 7.0|Enterprise Manager.
2. In the left pane, search the list of MS SQL servers to locate the server on which you have rehosted your *GoldMine* data.
3. Click **+** next to the server with the rehosted data.
4. Click **+** next to the **Security** folder.
5. Select **Logins**, as shown in the following figure, then right-click to display the local menu.

*MS SQL Server
7.0 Enterprise
Manager*



6. Select **New Login**.

SQL Server
Login
Properties
window
(General tab)

The screenshot shows the 'SQL Server Login Properties - New Login' dialog box with the 'General' tab selected. The 'Name' field contains 'DAVIDD'. Under the 'Authentication' section, 'SQL Server authentication' is selected with a radio button. The 'Domain' dropdown is empty, and 'Security access' has 'Grant access' selected. The 'Password' field contains asterisks. Under the 'Defaults' section, 'Database' is set to 'goldmine' and 'Language' is set to '<Default>'. The 'OK', 'Cancel', and 'Help' buttons are at the bottom.

7. Select *one* of the following methods of **Authentication** for this user:

- **Windows NT Authentication:** allows you to specify a user on your NT network to be imported into MS SQL Server. The user can log into MS SQL Server with the same username and password he uses to log in to the Windows NT network. Continue with step 8.
- **SQL Server Authentication:** prompts you to *create* a username and password for a user, and will *not* import users from Windows NT. Go to step 9.

8. Select the **Domain** where the user exists on your Windows NT network from the drop-down list. For **Security Access**, select **Grant Access**. Go to step 10.

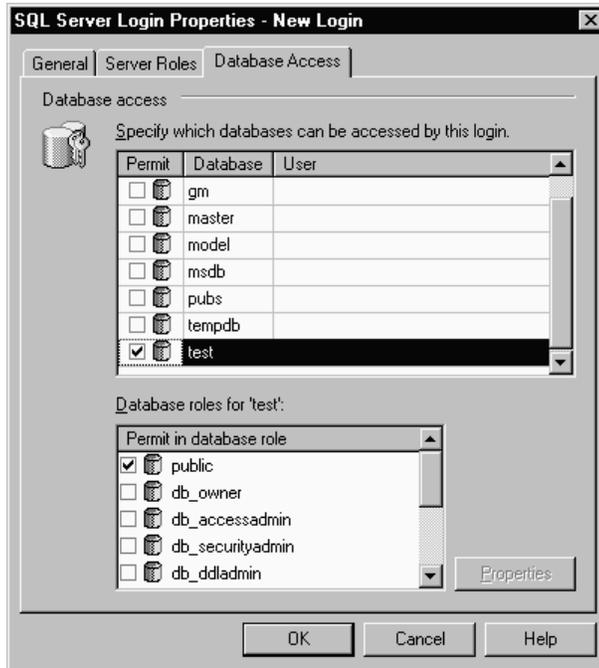
9. Type a **Password** for your **SQL Server authentication**. The entry appears as asterisks.

10. Type a user **Name**. If you selected **Windows NT Authentication** in step 7, this entry is typically the user name of an existing Windows NT account. If you selected **SQL Server Authentication**, this entry is a new username for MS SQL Server.

11. To select a default **Database** or **Language** (other than English) for the user, you can select available options listed in the corresponding drop-down lists.

12. Select the Database Access tab.

SQL Server Login Properties - New Login window (Database Access tab)



13. Identify the *GoldMine* database(s) to which you want to grant access for the new user, then click the appropriate checkbox(es).

14. Select OK.

You have finished creating a user, and granting access privileges to *GoldMine* database(s). To create additional users, repeat steps 4–14. After creating all users, continue with “Assigning Permissions and Granting Access.”

Assigning Permissions and Granting Access

To ensure the security appropriate to maintain the integrity of your data, you can work with the following settings:

- **Granting Table Access** is required to assign the appropriate security rights to each user who will access the *GoldMine* tables. You *must* grant table access to allow users to access the *GoldMine* data.
- **Assigning Server Roles** is one of two methods by which specific users can be granted permissions to create or remove objects, run system processes, and perform general administration on the SQL Server. If any users in your MS SQL Server implementation will perform these operations, they must be assigned appropriate rights. Assigning roles will give the user a predefined group of permissions for simultaneous use.
- **Setting Database Permissions** is the other method by which specific users can be granted administration permissions. Setting the database permissions allows you to pick individual rights for each user, instead of assigning a group of permissions with an administrative role.



CREATE TABLE rights in MS SQL are required for a user to perform the following operations in *GoldMine*:

- Rehosting data
- Rebuilding a database
- Creating transfer sets for synchronization

The CREATE TABLE privilege can be assigned with *either* a server role, *or* by setting database permissions for the SQL user who will perform these *GoldMine* operations. However, users who only perform routine *GoldMine* operations only need the options set in the following section, “Granting Table Access to All Users.”

Granting Table Access to All Users

To grant access to the *GoldMine* tables to all users:

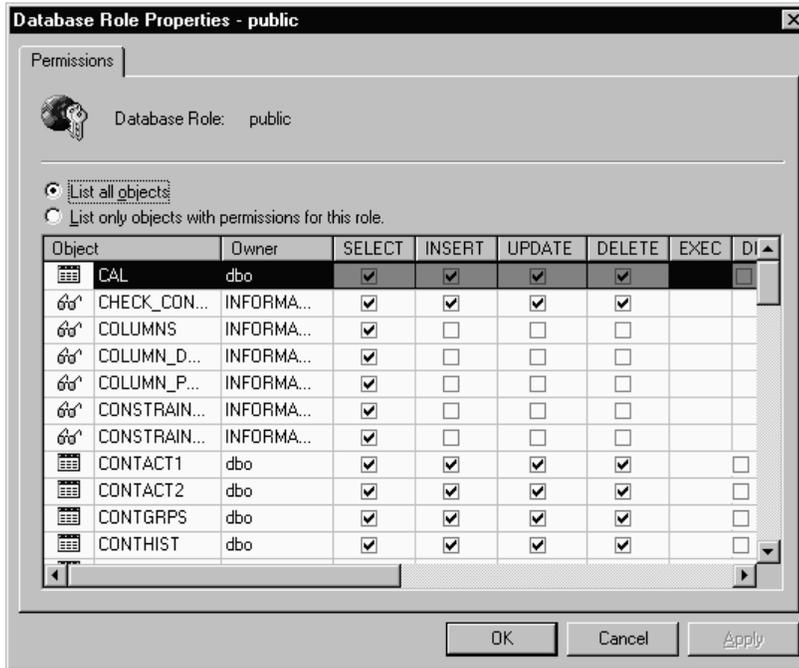
1. To start the SQL Enterprise Manager, from the Windows' task bar, select Start|Programs|Microsoft SQL Server 7.0|Enterprise Manager.
2. In the left pane, search the list of MS SQL servers to locate the server on which you have rehosted your *GoldMine* data.
3. Click next to the server with the rehosted data.
4. Click next to the Databases folder.

5. Click  next to the database to which you want to grant access.
6. Click **Roles**. The available roles appear in the detail pane.
7. Right-click the **Public** role to display the local menu, then select **Properties**. The **Database Role Properties** window appears.
8. Check that all users are displayed in the **User** pane.

If not, you will need to *either* **Add** them, *or* follow the procedure described in “Creating Logins” on page 30.

If all users are displayed, continue with step 9.
9. Click **Permissions**.

Permissions tab



The **Object** column of the **Permissions** tab lists  (tables) and  (views) available in the selected database, and the level of access your users have to these tables and views.

10. For each *GoldMine* table, check the boxes for **Select**, **Insert**, **Update**, and **Delete** to assign access right. Each *GoldMine* table appears in upper-case letters next to . For a list of the *GoldMine* tables in your database, see “Tables Created for *GoldMine* Databases” on page 44.
11. Click **OK** to enter settings from the **Permissions** tab. The **General** tab appears.
12. Click **OK** to enter settings from the **General** tab.

Your users have been granted access to the tables in this *GoldMine* database. To implement more than one *GoldMine* database, repeat steps 4–9 for each additional database.

When done, you can either:

- Go to “Configuring *GoldMine* Usernames for SQL Servers” on page 38
or
- Continue with “Granting Additional Permissions” to assign administrative rights

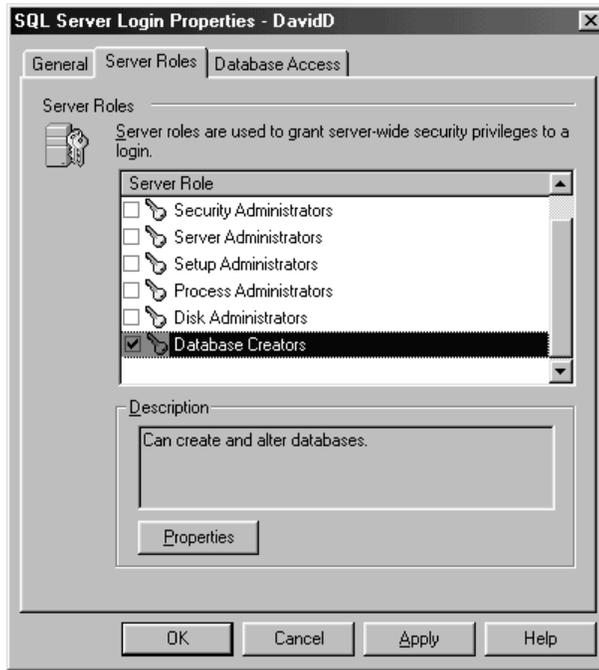
Assigning Server Roles

You may need to assign administrative rights to specific users. You can do this by assigning a group of permissions at once with a **role**.

To assign a role:

1. To start the **Enterprise Manager**, from the Windows’ task bar, select **Start|Programs|Microsoft SQL Server 7.0|Enterprise Manager**.
2. In the left pane, search the list of SQL servers to locate the server on which you have rehosted your *GoldMine* data.
3. Click  next to the server with the rehosted data.
4. Click  next to the **Security** folder.
5. Click **Logins**. The SQL Server users appear in the right pane.
6. Right-click on the user that you want to edit to display the local menu. Select **Properties**. The **SQL Server Login Properties** window appears. By default, the **General** tab appears.
7. Select the **Server Roles** tab. SQL Server displays the **Server Role** pane.

Server Roles tab



8. Check the appropriate boxes to assign specific administrative roles to the user. The following table shows the operations granted to each administrative role.

Statement	Database Creators	Process Administrators	Security Administrators	Server Administrators
Alter Database	yes	no	no	no
Create Database	yes	no	no	no
DBCC	no	no	no	yes
Deny	no	no	yes	no
Grant	no	no	yes	no
Kill	no	yes	no	no
Reconfigure	no	no	no	yes
Restore	yes	no	no	no
Revoke	no	no	yes	no
Shutdown	no	no	no	yes

The **Disk Administrators** and **Setup Administrators** roles grant privileges to perform only specified system-stored procedures. They do not have permissions to execute any Transact-SQL statements. However, the **System Administrators** fixed server role grants permissions to execute all Transact-SQL statements.



For details on specific roles or statements, see the [SQL Server Books Online](#), which is available from the Windows' task bar by selecting **Start|Programs|Microsoft SQL Server 7.0|Books Online**.

9. Click **OK**.

10. To grant server roles to additional users, repeat steps 5–9.

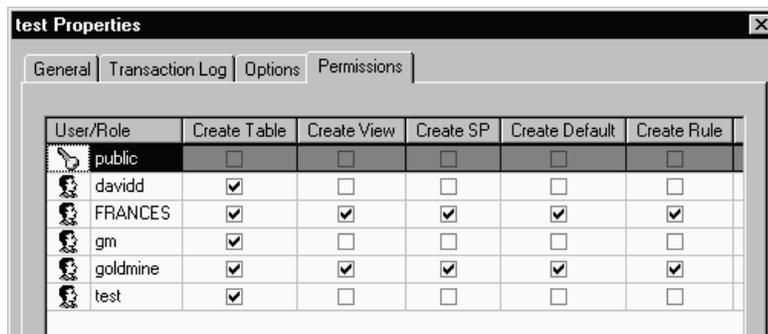
Assigning Database Permissions to Individual Users

A more flexible way to grant permissions can be accessed through the **Database Properties**. Assigning permissions in this manner allows specific statements to be allowed to that user while others are denied.

To grant permissions individually:

1. To start the **Enterprise Manager**, from the Windows' task bar, select **Start|Programs|Microsoft SQL Server 7.0|Enterprise Manager**.
2. In the left pane, search the list of SQL servers to locate the server on which you have rehosted your *GoldMine* data.
3. Click **+** next to the server with the rehosted data.
4. Click **+** next to the **Databases** folder.
5. Right-click the database to which you want to grant permissions to display the local menu. Select **Properties**.
6. Select the **Permissions** tab.

Permissions tab



7. Check the boxes corresponding to the statements and the users or roles to which you want to grant statements. For details about the statements listed in this tab, see your SQL Server 7.0 documentation.



The *only* statement relevant to *GoldMine* is the CREATE TABLE statement. *GoldMine* uses the CREATE TABLE statement when rehosting data, rebuilding databases, and creating transfer sets for synchronization. A *GoldMine* user performing these operations must have rights to execute the CREATE TABLE statement. Other options in the **Permissions** tab are never used by *GoldMine*, but may be useful for the purposes of database administration.

8. Click **Ok**.

9. To assign individual permissions to another database, repeat steps 4–9.

Configuring *GoldMine* User Names for MS SQL Servers

As an administrator, you have the responsibility of ensuring the security of your organization's data while permitting appropriate access to each user. To grant user access to *GoldMine* data, you must assign a unique user name in *GoldMine*.



For details on assigning unique user names, see "Setting up *GoldMine* Users" in the *GoldMine Reference Manual*.

However you assign SQL logins, the login to the SQL database server can be transparent to the users when the database server login and password is specified in the properties of the *GoldMine* user name.

To specify the SQL login for each user:

1. Within *GoldMine*, from the Main Menu, select **File|Configure GoldMine|User's File**. The **Users' Master File** dialog box appears.
2. In the **Username** column of the browse window, select the user for whom you wish to assign an SQL database login. Select **Properties**. The **[username] Properties** window appears. By default, *GoldMine* displays the **Profile** tab.
3. Select the **Access** tab.

Entering an SQL login and password

The screenshot shows a dialog box titled "SQL Server Logon" with a key icon in the top right corner. It contains two text input fields: "Logon name:" and "Password:". The "Logon name:" field is currently empty, and the "Password:" field is also empty.

4. In the **SQL Server Logon** section, specify the **Logon Name** of up to 15 alphanumeric characters. This entry is the login to the SQL Server created for this user.

5. Type the **P**assword you created for this SQL Server login.

GoldMine uses the **L**ogon name entry to attempt a connection to the SQL database server. If no **L**ogon name is specified, *GoldMine* attempts to log in to the database server using the *GoldMine* user name and password.

6. Click **O**K. The *GoldMine* username has been configured. Repeat steps 2–6 for each *GoldMine* user who will access the SQL database.



If you have contact sets residing on multiple SQL database servers, you can enter an additional SQL login for the contact set in *GoldMine's* **C**ontact **F**ile **P**rofile dialog box. To access the **C**ontact **F**ile **P**rofile dialog box, from *GoldMine's* Main Menu, select **F**ile|**O**pen File. In the browse window, select the contact set, then select **P**roperties. The **C**ontact **F**ile **P**rofile dialog box has an **S**QL **S**erver **L**ogon section identical to the figure shown above. To enter an additional login, follow steps 4–6 in this section.

Your *GoldMine* user configuration is complete. To complete the implementation of SQL Server 7.0, continue with “Installing MS SQL Client Utilities.”

Installing MS SQL Client Utilities

For a workstation to access an SQL database, the MS SQL client utilities must be installed. The installation requires approximately 35 MB available on each workstation. The following steps were designed for use with the SQL Server 7.0 Standard Edition on a Windows 95/98 workstation. Before proceeding, you should uninstall any earlier versions of the MS SQL client on the workstation.

To install the client utilities:

1. Close all open applications.
2. Place the SQL Server 7.0 CD-ROM in the workstation's CD drive. If the workstation does not have a CD drive and you are installing through a network, use Windows Explorer to open the network CD drive, then double-click **autorun.exe**.

If Internet Explorer 4.01 SP1 or later is not installed, you must **I**nstall **S**QL **S**erver **7.0** **P**requisites, and return to this menu when done.

If Internet Explorer 4.01 SP1 or later is installed on the workstation, continue with step 3.

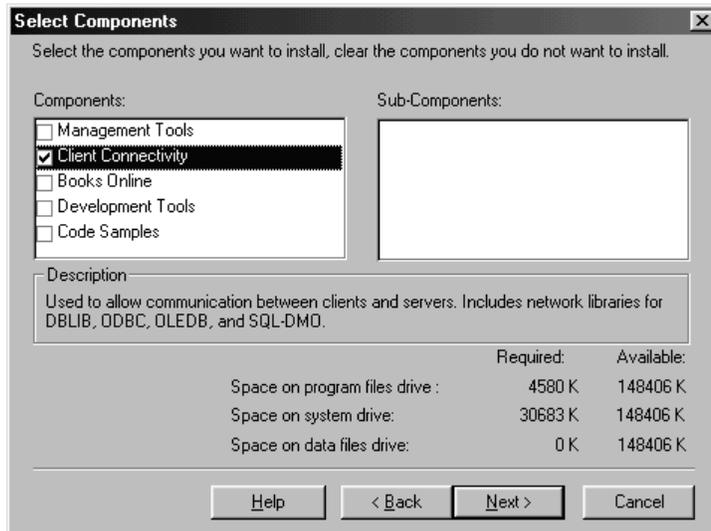
3. Click **I**nstall **S**QL **S**erver **7.0** **C**omponents.

4. Click **Database Server - Standard Edition**. A message may appear stating that the server component cannot be installed to this workstation. If so, click **OK**. The **Welcome** screen of the **Microsoft Server 7.0** installation wizard appears.
5. Click **Next>**. The **License Agreement** appears.
6. Read the **License Agreement**, then select **Yes** to continue.
7. Type your **Name** and **Company** name, then click **Next>**.
8. Type the **CD Key** for your copy of **SQL Server 7.0**, then click **OK**.
9. Read the details on the product ID, then click **OK**.
10. In the **Setup Type** dialog box, you can specify a folder to install to with the **Browse** button. By default, MS SQL installs the client utilities in **C:\MSSQL7**. Click **Next>**. The **Select Components** dialog box appears.



The following installation options are the *minimum* required for *GoldMine's* purposes. You *must* install the listed components, but can install additional components if desired.

Select Components dialog box



11. Check **Client Connectivity**. You can uncheck any other options, as they are unnecessary for *GoldMine*.



For details about the optional components, see your **SQL Server** documentation.

12. Click **Next>** to display the final screen of the wizard.

13. Click **Next>** to begin copying files.

14. Click **Finish** when prompted.

You have finished installing the MS SQL client utilities on this workstation. Continue with “Configuring Workstations.”

Configuring Workstations

After installing the client utilities on a workstation, you must configure the Client Network Utility to access your database server, and create a shortcut to *GoldMine* for the workstation.

To configure the Client Network Utility:

- 1.** To start the **Enterprise Manager**, from the Windows’ task bar, select **Start|Programs|Microsoft SQL Server 7.0|Client Network Utility**. The **SQL Server Client Network Utility** dialog box appears.
- 2.** Click **Add**. The **Add Network Library Configuration** screen appears.
- 3.** Specify the name and address of the database server in the **Server alias** field.
- 4.** Select an appropriate network library from the list, and specify connection parameters for the selected library. Your network and database server determine these settings. For details on these entries, see your network administrator and MS SQL documentation.
- 5.** Click **OK** to return to the **SQL Server Client Network Utility** window.
- 6.** Click **OK**.

You are done configuring the Client Network Utility for this workstation.

Creating the Shortcut and Starting GoldMine

The shortcut to start *GoldMine* is the same as the other shortcuts already on your desktop. Set the shortcut to point to GMW4.EXE installed on the network. If you are prompted for the location of the idapi32.dll, navigate to the location of that DLL on the network.



For details on installing *GoldMine* on a single-user system, and starting *GoldMine* on any system, see “Installing and Starting *GoldMine*” in the *GoldMine User’s Guide*.

For details on installing *GoldMine* on your network, see “Installing *GoldMine* on a Network” in the *GoldMine Reference Manual*.

If BDE errors appear when you start *GoldMine*, the most likely cause is that one of the steps in this guide was overlooked.

To correct the errors, you can:

- Review the procedures contained in this document.
- See the “Troubleshooting Guide,” on page 46.