CHAPTER 24

Creating Database Objects with SQL Server

✦ ✦ ✦ ✦

In This Chapter

Introducing Enterprise Manager

Creating databases and tables

Constructing indexes and database diagrams

Controlling security

Managing logins and users

✦ ✦ ✦ ✦

In this chapter I'm going to show you how to use SQL Server 7's Enterprise Manager utility to create various objects in your database. These objects include databases, tables, indexes, database diagrams, and stored procedures. You can also use Enterprise Manager to manage security by creating logins, map them to users and assign security to the various database objects.

Introducing Enterprise Manager

Enterprise Manager is the primary utility that you use to manage an SQL Server database system. It has the ability to create databases and database objects, control the database's security, and perform routine operational activities like database backups and reorganizations. It can control the properties of the database server itself, as well as start and stop the server from a remote location. In short, it is a very powerful utility that is used by several different types of users to administer an SQL Server database.

Note

Security dictates function: Anyone with a valid login can use Enterprise Manager. However, the functions available at the database server level are based on the login's capabilities and functions on the corresponding user's system.

Enterprise Manager fundamentals

Enterprise Manager is a Microsoft Management Console (MMC) application, meaning that it shares its look and feel with many other system management tools provided by Microsoft (see Figure 24-1). To start Enterprise Manager, choose Start ➪ Programs ➪ Microsoft SQL Server 7.0 ➪ Enterprise Manager. Like many tools available from Microsoft, most of its operations can be performed by using a wizard or by manually configuring property windows.

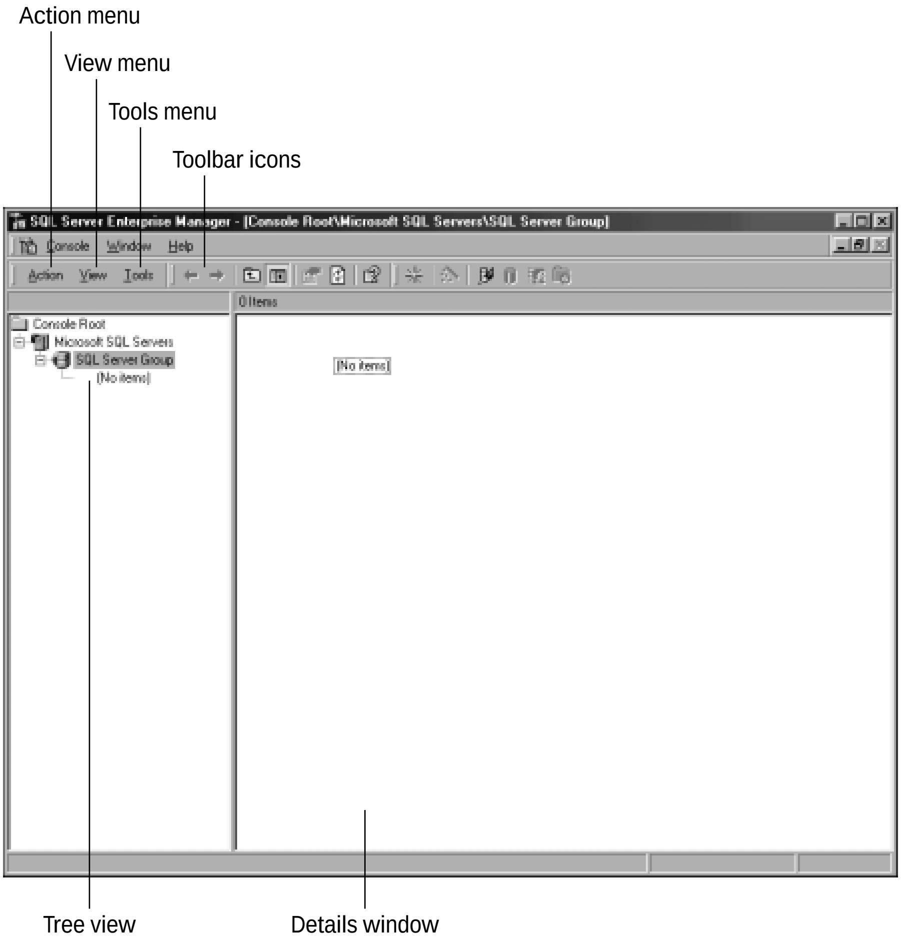


Figure 24-1: Viewing the Enterprise Manager window

- **Tree view** contains a hierarchical list of the database servers, databases, and classes of database objects that can be viewed or changed.
- **Details window** provides additional information for the selected icon in the tree view.
- **Action menu** contains a list of menu items that can be performed based on the context.
- **View menu** controls how the information will be displayed using MMC. Typically, all of the items will be selected including Console Tree, Description Bar, Status Bar, and the toolbars. You can also choose to display how the information is displayed in the details window using the Large, Small, Details, and List formats.
- **Tools menu** provides a list of shortcut menu items to help you perform most common tasks quickly.
- **Toolbar icons** provides a list of shortcut icons for the most popular pop-up menu items.

Registering a database server

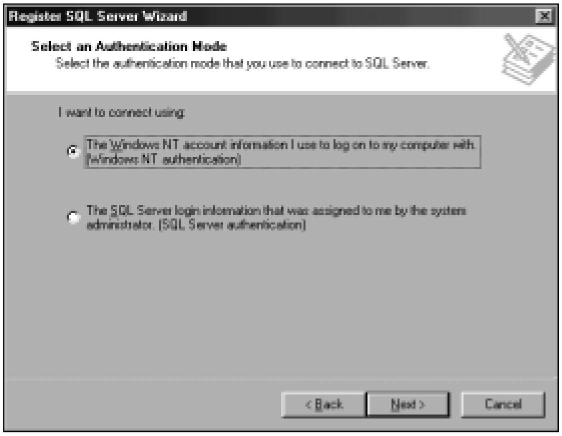
Enterprise Manager is capable of managing several SQL Server database servers across multiple remote computers. These computers are listed under the SQL Server Group icon in the tree view. However, before you can access a remote server, you must register the server with Enterprise Manager.

Note

Duh: In order to run Enterprise Manager on your computer, you first have to install it. If you are using the same computer on which you are running the database server, Enterprise Manager is already installed. If you are using a different computer, you can either choose to install the desktop version of SQL Server or just the SQL Server utilities from the server version.

Registering a server is very easy. Just follow these steps:

1. Right click on the SQL Server Group icon in the tree view and right click to display the pop-up menu. Choose New SQL Server Registration. You can also choose Action ➪ New SQL Server Registration. After choosing either option, the Register SQL Server Wizard will be displayed.
2. After pressing Next, the wizard will display a list of available database servers (see Figure 24-2). Select one and press the Add button. If the server you want to use isn't listed, type the name in the text box at the top of the list and press Add. Then press Next to continue.



Register SQL Server Wizard

Select an Authentication Mode

Select the authentication mode that you use to connect to SQL Server.

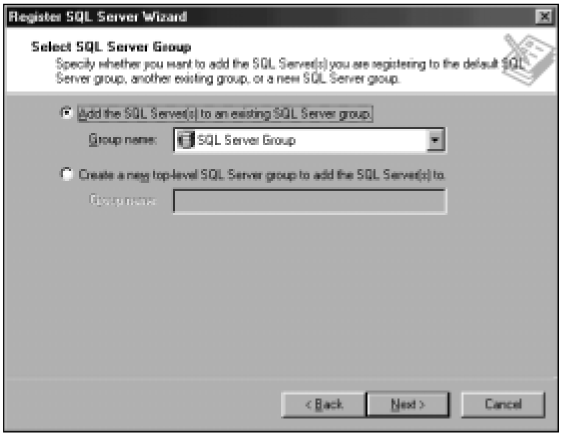
I want to connect using:

● The Windows NT account information I use to log on to my computer with. (Windows NT authentication)

○ The SQL Server login information that was assigned to me by the system administrator. (SQL Server authentication)

Figure 24-2: Choosing the database server to add

3. In the next step, you have a choice of which authentication system to use (see Figure 24-3). Since you are already using a Windows computer (it doesn't matter whether it's Windows 98 or Windows 2000/NT), you should probably use Windows NT account information. However, you must use a login that is already defined on the server. If you choose SQL Server authentication, the wizard will then prompt you for how you want to log in to the server. You can supply your login information, which Enterprise Manager will use each time it starts, or you can choose to be prompted for the information.



Register SQL Server Wizard

Select SQL Server Group

Specify whether you want to add the SQL Server(s) you are registering to the default SQL Server group, another existing group, or a new SQL Server group.

● Add the SQL Server(s) to an existing SQL Server group.

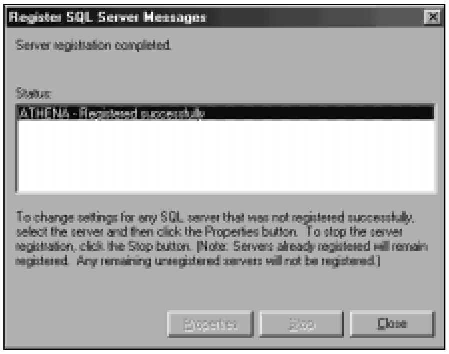
Group name: SQL Server Group

○ Create a new top-level SQL Server group to add the SQL Server(s) to.

Group name:

Figure 24-3: Selecting an authentication method

4. After you finish specifying how Enterprise Manager should log in to the database server, you can choose where the icon for the server will be stored (see Figure 24-4). By default, it will be stored under the SQL Server Group, but you can choose another group or create a new group.



Server registration completed.

Status:

ATHENA - Registered successfully

To change settings for any SQL server that was not registered successfully, select the server and then click the Properties button. To stop the server registration, click the Stop button. [Note: Servers already registered will remain registered. Any remaining unregistered servers will not be registered.]

Figure 24-4: Placing the icon for the database server in the tree view

5. When you reach the end of the wizard, you will see a list of database servers you want to register. Pressing Finish will start a process to connect to each of the database servers you specified (see Figure 24-5). This process will report any errors that are encountered. When the process is finished, press Close to finish the wizard and return to Enterprise Manager.

Cross-Reference

Refer to Chapter 23 for a discussion on Windows NT Authentication vs. SQL Server Authentication.

Viewing database servers and their objects

Once you have registered your database server, you can expand the icon tree to show the database servers you can access and the server-level objects available (see Figure 24-6).

- **Databases** contains the collection of databases on the server you can access. You can create, browse, edit, modify and delete tables, indexes, stored procedures, users, roles, and other database objects.
- **Data Transformation** contains information about data transformation packages and the data repository. You can view and modify data transformation jobs, view and edit metadata, and search for information in the repository.
- **Databases** contains the collection of databases on the server you can access. You can create, browse, edit, modify and delete tables, indexes, stored procedures, users, roles, and other database objects.

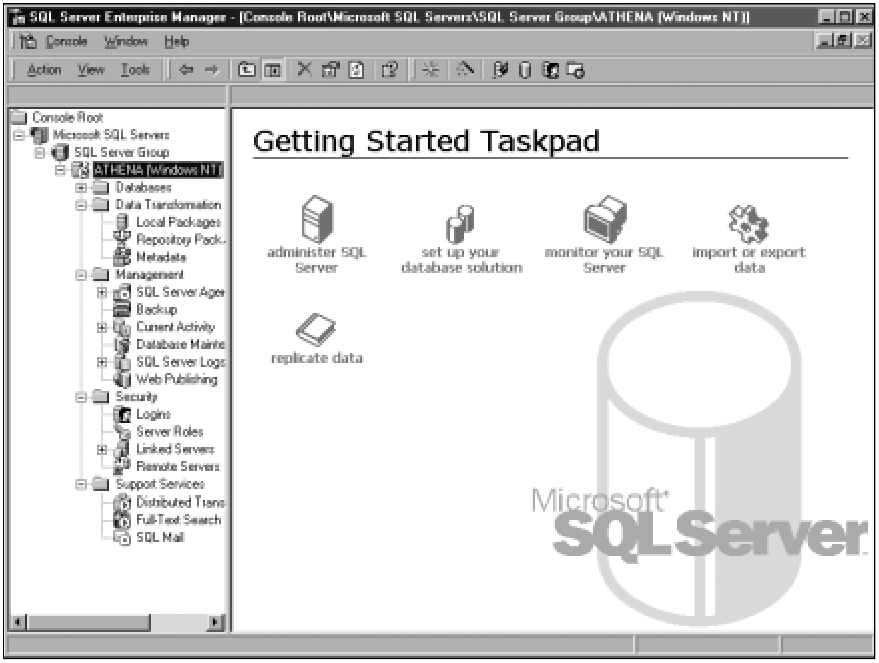
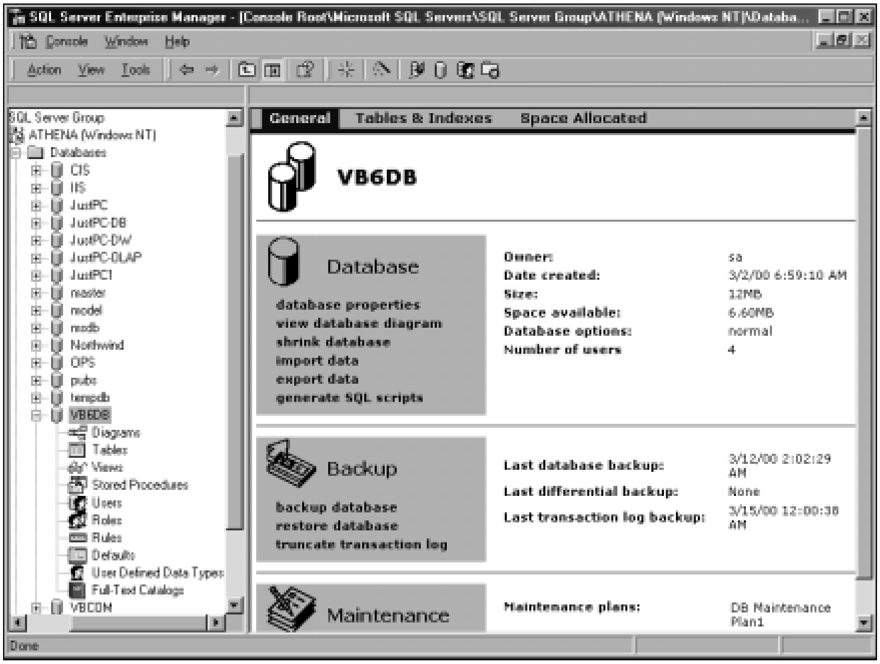


Figure 24-5: Registering the database servers



Owner: sa
Date created: 3/2/00 6:59:10 AM
Size: 12MB
Space available: 6.60MB
Database options: normal
Number of users 4

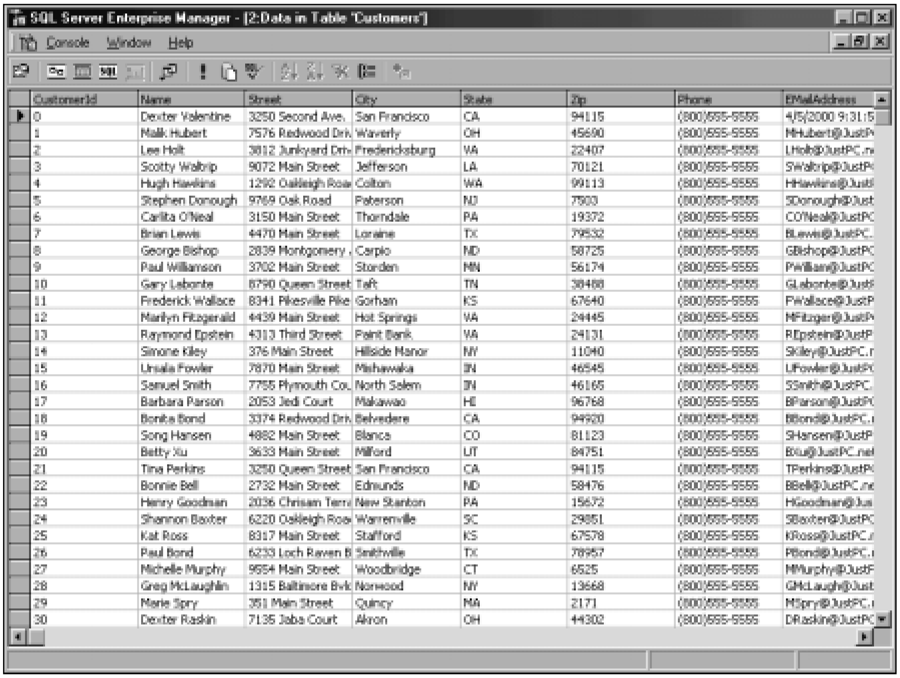
Last database backup: 3/12/00 2:02:29 AM
Last differential backup: None
Last transaction log backup: 3/15/00 12:00:38 AM

Maintenance plans: DB Maintenance Plan1

Figure 24-6: Browsing database servers and database objects

- ✦ **Data Transformation** contains information about data transformation packages and the data repository. You can view and modify data transformation jobs, view and edit metadata, and search for information in the repository.
- ✦ **Management** contains information about operational activities performed by the database server. This includes things such as database backups, log files, maintenance jobs, and SQL Server Agent activities.
- ✦ **Security** contains information about login and server roles, plus information about how to access remote database servers and linked database servers.
- ✦ **Support Services** contains information about the Distributed Transaction Coordinator, the Full Text Search feature, and SQL Mail.

Expanding the icon associated with a database reveals the objects inside that can be managed by Enterprise Manager (see Figure 24-7). These objects exist only in the specified database, and not at the database server level. In order to access the database, your login must be mapped to a user in the database.


SQL Server Enterprise Manager - [2:Data in Table 'Customers']
Console
Window
Help

CustomerId	Name	Street	City	State	Zip	Phone	EMailAddress
0	Dexter Valentine	3250 Second Ave.	San Francisco	CA	94115	(800)555-5555	4/5/2000 9:31:5
1	Malik Hubert	7576 Redwood Driv	Waverly	OH	45690	(800)555-5555	MHubert@JustP
2	Lee Holt	3812 Junkyard Driv	Fredericksburg	VA	22407	(800)555-5555	LHolt@JustPC.n
3	Scotty Waltrip	9072 Main Street	Jefferson	LA	70121	(800)555-5555	SWaltrip@JustP
4	Hugh Hawkins	1292 Oakleigh Roa	Colton	WA	99113	(800)555-5555	HHawkins@Just
5	Stephen Donough	9769 Oak Road	Paterson	NJ	7503	(800)555-5555	SDonough@Just
6	Carlita O'Neal	3150 Main Street	Thorndale	PA	19372	(800)555-5555	CO'Neal@JustPC
7	Brian Lewis	4470 Main Street	Loraine	TX	79532	(800)555-5555	BLewis@JustPC.
8	George Bishop	2839 Montgomery .	Carpio	ND	58725	(800)555-5555	GBishop@JustPC
9	Paul Williamson	3702 Main Street	Storden	MN	56174	(800)555-5555	PWilliam@JustPC
10	Gary Labonte	8790 Queen Street	Taft	TN	38488	(800)555-5555	GLabonte@Just
11	Frederick Wallace	8341 Pikesville Pike	Gorham	KS	67640	(800)555-5555	FWallace@JustP
12	Marilyn Fitzgerald	4439 Main Street	Hot Springs	VA	24445	(800)555-5555	MFitzger@JustP
13	Raymond Epstein	4313 Third Street	Paint Bank	VA	24131	(800)555-5555	REpstein@JustP
14	Simone Kiley	376 Main Street	Hillside Manor	NY	11040	(800)555-5555	SKiley@JustPC.n
15	Ursala Fowler	7870 Main Street	Mishawaka	IN	46545	(800)555-5555	UFowler@JustPC
16	Samuel Smith	7755 Plymouth Cou	North Salem	IN	46165	(800)555-5555	SSmith@JustPC.
17	Barbara Parson	2053 Jedi Court	Makawao	HI	96768	(800)555-5555	BParson@JustPC
18	Bonita Bond	3374 Redwood Driv	Belvedere	CA	94920	(800)555-5555	BBond@JustPC.
19	Song Hansen	4882 Main Street	Blanca	CO	81123	(800)555-5555	SHansen@JustP
20	Betty Xu	3633 Main Street	Milford	UT	84751	(800)555-5555	BXu@JustPC.net
21	Tina Perkins	3250 Queen Street	San Francisco	CA	94115	(800)555-5555	TPerkins@JustPC
22	Bonnie Bell	2732 Main Street	Edmunds	ND	58476	(800)555-5555	BBell@JustPC.ne
23	Henry Goodman	2036 Chrisam Terra	New Stanton	PA	15672	(800)555-5555	HGoodman@Jus
24	Shannon Baxter	6220 Oakleigh Roa	Warrenville	SC	29851	(800)555-5555	SBaxter@JustPC
25	Kat Ross	8317 Main Street	Stafford	KS	67578	(800)555-5555	KRoss@JustPC.
26	Paul Bond	6233 Loch Raven B	Smithville	TX	78957	(800)555-5555	PBond@JustPC.
27	Michelle Murphy	9554 Main Street	Woodbridge	CT	6525	(800)555-5555	MMurphy@JustP
28	Greg McLaughlin	1315 Baltimore Bvk	Norwood	NY	13668	(800)555-5555	GMcLaugh@Just
29	Marie Spry	351 Main Street	Quincy	MA	2171	(800)555-5555	MSpry@JustPC.
30	Dexter Raskin	7135 Jaba Court	Akron	OH	44302	(800)555-5555	DRaskin@JustPC

Figure 24-7: Browsing the objects in a database

- ✦ **Diagrams** contains a set of database diagrams showing how the tables are related to each other.
- ✦ **Tables** contains the set of tables in your database.

- **Views** contains the collection of views in your database.
- **Stored Procedures** contains all of the stored procedure definitions for your database.
- **Users** contains information about all of the users that may access your database.
- **Roles** contains the description of the security roles used in the database.
- **Rules** is maintained for backwards compatibility with older versions of SQL Server. They have been superceded by **Check** constraints, which are maintained in the Design Table window.
- **Defaults** contains the collection of default values that may be referenced by name or associated with a particular column in a table.
- **User Defined Data Types** contains the set of user defined data types that may be used as the type for a particular column.
- **Full-Text Catalogs** contains the definitions associated for the full-text catalogs, which hold the set of full-text indexes in the database.

Browsing data

When you select a particular table or view, you can choose to browse its contents by right clicking on its icon and choosing Open Table or Open View from the pop-up menu. You can choose to return all rows in the table or specify the number of rows you wish to see starting from the top of the table. In either case, you'll see a display similar to that shown in Figure 24-8.

The information in the table is displayed using a grid, where each row in the grid corresponds to a row in the table, and each column corresponds to a column in the table. You can scroll through the values using the scroll bars and adjust the widths of each column to best display the information. The current row will be indicated with an arrow in the row prefix area, while the name of each column will be display in the column prefix area.

In addition to viewing the data from the table, you can also modify the data, assuming that you have the proper security permissions. To change a value in an existing row, simply edit the value in the cell. When you move to another row, the data in the table will be updated.

To add a new row, scroll to the end of the table and look for the last row. An asterisk (*) will be displayed in the row prefix area, indicating that this row has not been added to the table. As with updating a row, simply move to another row to commit the changes.

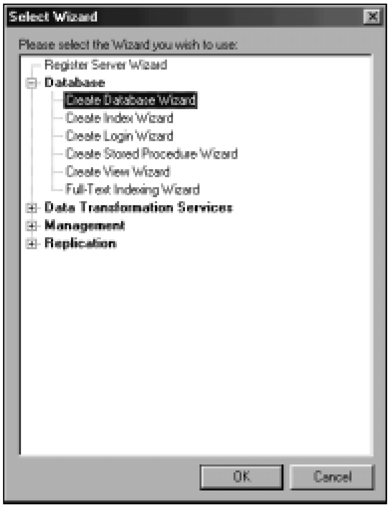


Figure 24-8: Browsing the contents of a table

Other operations, including delete, may be performed on a row by right clicking on the row and choosing one of the functions listed in the pop-up menu. Note you can use the pop-up menu to perform the same function on multiple rows by clicking on the row prefix area, holding the left mouse button down, and dragging the cursor to mark the rows as selected, then right clicking on the selected rows to display the pop-up menu.

Note

A short term lock: When you browse rows in your table or view, you may be holding locks in the database. If there has been no activity for a while, the Enterprise Manager will ask you if you want to continue working with the results pane. If you respond No, or don't respond within a minute, Enterprise Manager will close the pane, and any uncommitted changes you have made will be discarded. You can refresh the data by right clicking anywhere on the pane and choosing Run from the pop-up menu.

Tip

There's more than meets the eye: Open Table and Open View functions exist on top of a table-oriented query facility. You can modify the underlying query to change the rows returned either by changing the SQL **Select** statement directly, or by using the grid and diagram graphical tools. You can view the panes that contain this information by right clicking on the results pane and selecting the panes you wish to view under the Select Panes pop-up menu item.

Databases and Tables

Databases exist in SQL Server to hold tables, and tables exist in the database to hold your data. Creating them with Enterprise Manager's wizards makes the job much easier than in earlier versions of SQL Server.

Creating a database

To create a new database, you must use a login that is assigned to either the sysadmin or dbcreater roles. Typically, you'll create your database using the `sa`, login, but depending on how your database security is organized, you may use another login.

Before you create a database

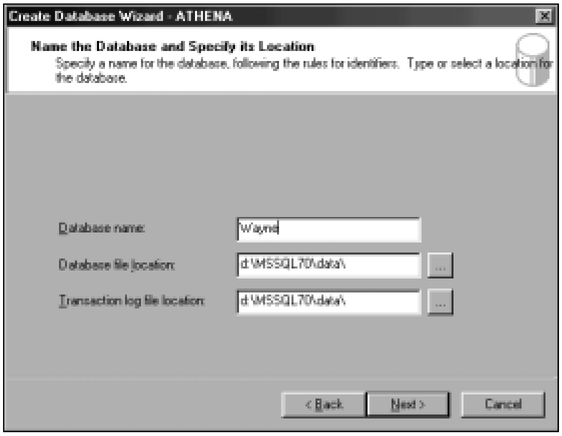
When you create a database, you should have the following information ready:

- The name of the database server where you want to create the database.
- The name of the database itself. While you can use nearly any combination of characters in the name, I suggest beginning the name with a letter, followed by any combination of letters, numbers, and special characters such as @, $, #, and _ up to a maximum of 123 characters. If you don't follow this suggestion, you may have to surround the name of the database with quotes or square brackets to ensure that the name is properly understood.
- The location of the database files. This is a reference to a directory somewhere on the database server where the files containing the objects inside the database will be held.
- The location of the log files. This is a reference to a directory somewhere on the database server where the file containing the information used to hold changes made to the database is kept.
- The initial size of the database files and how they should grow over time. You should choose initial values that are sufficient to cover the space you need in the beginning, and a growth method that allows your database over time.
- The initial size of the log files and how they should grow over time. The initial size determines how much space is originally allocated for the log, and the growth method determines how the log should be increased in case the server fills it up.

Creating your database

To create a database using Enterprise Manager, follow these steps:

1. Start Enterprise Manager (choose Start ➪ Program Files ➪ Microsoft SQL Server 7.0 ➪ Enterprise Manager) and select the database server where the new database will reside. Select Tools ➪ Wizards. This will display the Select Wizard dialog box. Expand the Database node, choose Create Database Wizard, and press OK (see Figure 24-9).



Name the Database and Specify its Location

Specify a name for the database, following the rules for identifiers. Type or select a location for the database.

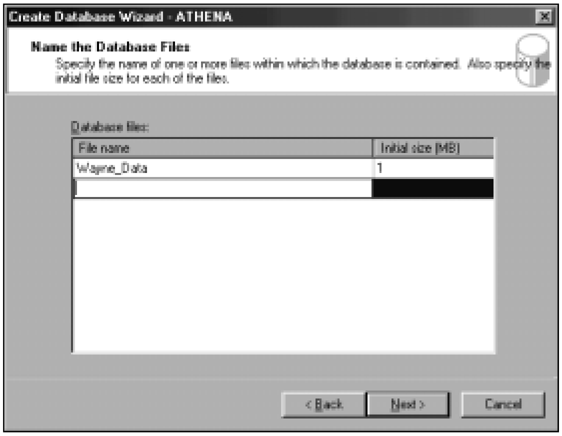
Database name: Wayne

Database file location: d:\MSSQL70\data\

Transaction log file location: d:\MSSQL70\data\

Figure 24-9: Choosing the Create Database wizard

2. When the wizard starts, press Next to go to the first step (see Figure 24-10). In this step, you'll have to provide a name for the database, plus the location of the files that hold the database.



Name the Database Files

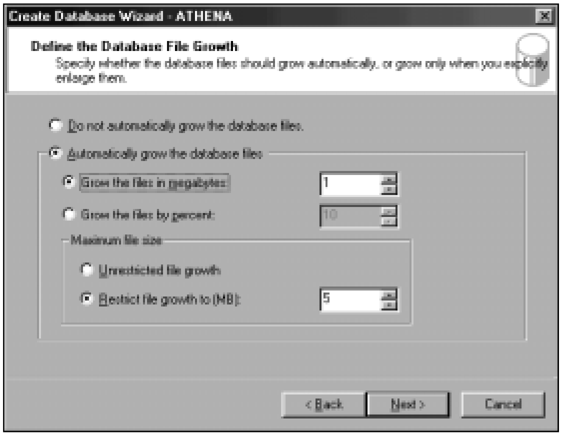
Specify the name of one or more files within which the database is contained. Also specify the initial file size for each of the files.

Database files:

File name	Initial size (MB)
Wayne_Data	1

Figure 24-10: Naming your database and locating where its information will be stored

3. In the next step of the wizard, you'll be prompted to enter the name of the file or files you want to use, and their initial size (see Figure 24-11). By default, the wizard will create one file with the name of the database, followed by `_Data` and an initial size of one megabyte.



Create Database Wizard - ATHENA

Define the Database File Growth

Specify whether the database files should grow automatically, or grow only when you explicitly enlarge them.

○ Do not automatically grow the database files.

● Automatically grow the database files

● Grow the files in megabytes: 1

○ Grow the files by percent: 10

Maximum file size

○ Unrestricted file growth

● Restrict file growth to (MB): 5

Figure 24-11: Specifying the files to hold your database

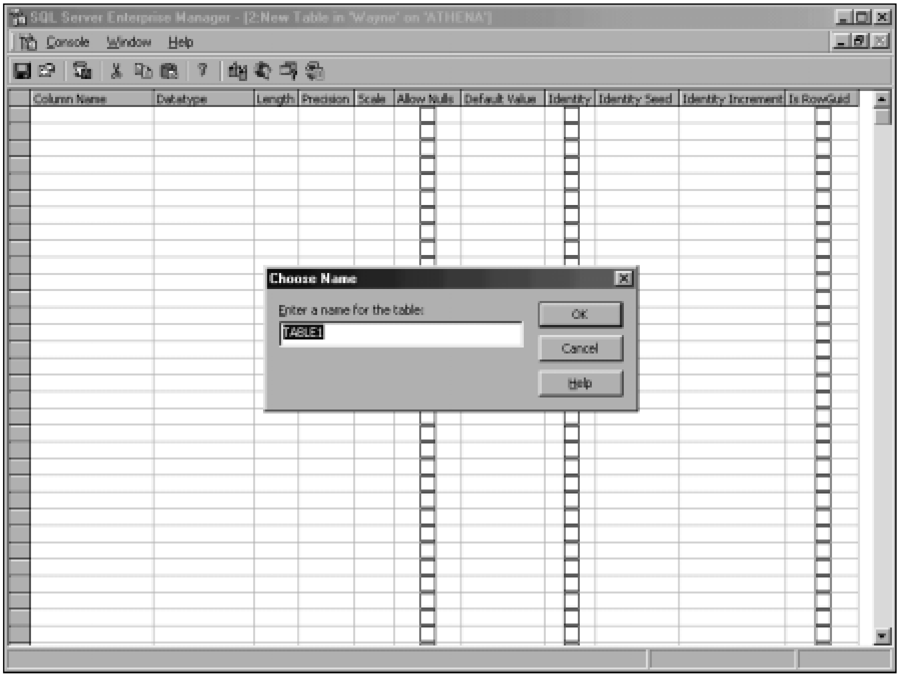
4. After specifying the file or files that will hold the database, you'll be prompted to choose how the database server should grow your files (see Figure 24-12). You can choose to grow your database file, in fixed sized chunks or by adding a fixed percentage of the currently allocated space. You can also place a maximum file size to prevent unlimited file growth.

Tip

Limits are good: When allocating space for a database file, you should place a limit on the file size. Unless you carefully monitor the amount of space your database uses, and periodically reorganize and compress the files, a ten-megabyte database may end up using ten gigabytes of disk space.

5. The wizard will repeat steps three and four, but for the transaction log files. The default file name is the name of the database followed by `_Log`.

6. In the last step of the wizard, all of the information collected will be displayed. If you want to change any of these parameters, press Back Otherwise, pressing Finish will begin the process to create the database. Because SQL Server physically writes binary zeros over every byte of space allocated for the database files and logs, the actual creation process may take a while if you are building a big database. A message box will be displayed letting you know if the database was successfully created. Press No to return to Enterprise Manager.



Enter a name for the table: TABLE1

Figure 24-12: Choosing file limits

Note

Covering your butt: If you don't have a maintenance plan for your database server that covers your new database, I strongly recommend that you create one when the Create Database Wizard finishes. See the *SQL Server 7 for Dummies* by Anthony T. Mann for details on how to create a maintenance plan.

Tip

Property pages: While this may be obvious, it's worth mentioning anyway. All of the information entered in the Create Database Wizard can also be entered using a database property page by right clicking on the Databases icon beneath the database server icon and choosing New Database from the pop-up menu. Once the database has been created, you can review and change this information by right clicking on the database and selecting Properties from the pop-up menu.

Creating a table

A database is pretty useless without any tables, so after creating your database, you'll probably want to create some tables. While the Enterprise Manager doesn't include a wizard to help you create a table, the Design Table window is very easy to use.

Database Performance

Believe it or not, having a faster CPU will not necessarily make your database server faster. A database server is very I/O intensive. Anything that allows your database to retrieve data faster from the disk will help the server's performance.

Before looking at hardware solutions to improve your database server's performance, you should ensure that the database server isn't competing with other processes on the server for memory and disk I/O. Dedicating a computer for the database server is highly recommended. If you can't dedicate a computer, you should at least try to place your database files on dedicated disk drives.

Adding memory to your server allows the database server to cache more data in memory; this is the biggest change you can make to improve database performance. After all, retrieving data from memory is much faster than retrieving it from disk. This is why Microsoft has gone to the trouble of making special versions of Windows NT to support more than the standard 2 gigabytes of main memory and to create the Data Center version of Windows 2000 which can support even more main memory.

After adding memory to your system, using SCSI disk drives is the next most important improvement. They not only allow you to manage up to 15 disk drives on a single card, but they also support concurrent operations on each drive. Thus, you can have multiple disk drives performing seeks, while others are transferring data. SCSI-III can transfer data faster than SCSI-II or SCSI-I and should be used for best performance.

Finally, using faster disk drives themselves will also improve performance. Disks that spin at 7,200 revolutions per minute (RPM) will transfer data faster than those that transfer data at 5,400 RPM, although two 5,400 RMP disk drives will probably perform better than one 7,200 RPM disk drive (assuming that you are using SCSI disk drives and that you can spread the I/O activity evenly between the two disk drives).

Before you create your table

Before you create your table, you should have a list of columns that the table will contain, along with values for the following attributes for each column:

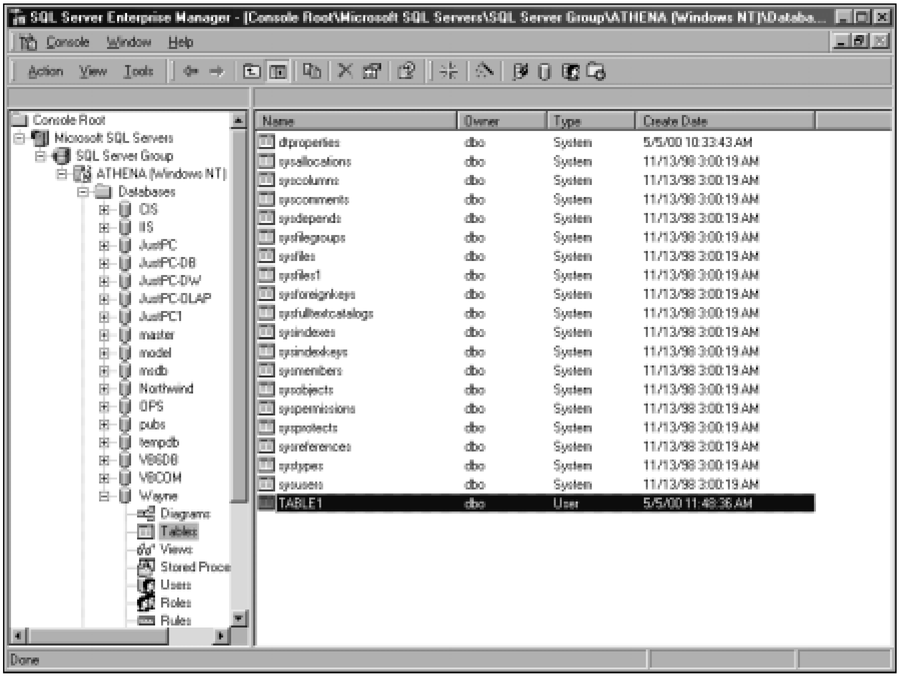
- The name of each column. The name should begin with a letter and can be followed by any combination of letters, numbers, and special characters such as @, $, #, and _ up to a total length of 128 characters. While spaces and other special characters may be used, you may have to surround the column name with quotes or square brackets when coding SQL statements.
- The data type for each column. The data type should be one of the data types listed in Chapter 23. Depending on the exact data type, you may have to adjust values for Length, Precision, and Scale.
- Whether the column will accept **Null** values. Note that **Null** values should not be used in any column that is used as part of a primary or foreign key.

- The default value for the column. This is an optional attribute that allows you to specify a default value rather than assigning a value of **Null** when adding a new row to the table.
- Whether to make the column an identity column. An identity column contains a value that can uniquely identify a row in the table. Only one column per table may be an identity column. This value is an integer value, which increases with each new row. It can be used with **Int**, **Decimal(4,0)** and any other data type that represents a whole number. You can specify the initial value in Identity Seed, and the increment added after each value is used under Identity Increment.
- Whether to make the column a row GUID column. A row GUID column is similar to an identity column, but rather than storing an integer value, a GUID value is stored instead. As with identity columns, only one row GUID column is permitted per table. Typically, you would use a row GUID column only if you were planning to implement database replication.

Creating your table

Follow these steps to create a table:

1. Expand the icon tree to reveal the Tables icon in the database where you want to build the table. Then right click on the Tables icon and select New Table from the pop-up menu. This will display the Design Table window, plus a message box that prompts you for the name of the table (see Figure 24-13).
2. Enter the name of your table and press OK.
3. Fill out the grid by entering the information for each column in your table as a row in the form. You must enter values for Column Name and Datatype. Selecting a data type will automatically fill in values for the Length, Precision, and Scale columns. If the data type is a variable length data type such as **Char** or **Decimal**, then you should adjust these values as appropriate.
4. Insert additional columns if you need them.
5. To mark a column or group of columns as the primary key, select the rows containing the columns that make up the key, right click to display the pop-up menu, and choose Set Primary Key. Each column that is part of the primary key will have an icon of a key displayed in the row header. Also, remember to remove any check marks in the Allow Nulls column for each of the fields that are included with the primary key.
6. When you have finished entering all of the columns for the table, you can press the Save icon at the top of the form, or simply close the window to save your table definition. If you close the window, you’ll be asked if you want to save your change. Press Yes if you do and No if you don’t.



Name	Owner	Type	Create Date
dtproperties	dbo	System	5/5/00 10:33:43 AM
sysallocations	dbo	System	11/13/98 3:00:19 AM
syscolumns	dbo	System	11/13/98 3:00:19 AM
syscomments	dbo	System	11/13/98 3:00:19 AM
sysdepends	dbo	System	11/13/98 3:00:19 AM
sysfilegroups	dbo	System	11/13/98 3:00:19 AM
sysfiles	dbo	System	11/13/98 3:00:19 AM
sysfiles1	dbo	System	11/13/98 3:00:19 AM
sysforeignkeys	dbo	System	11/13/98 3:00:19 AM
sysfulltextcatalogs	dbo	System	11/13/98 3:00:19 AM
sysindexes	dbo	System	11/13/98 3:00:19 AM
sysindexkeys	dbo	System	11/13/98 3:00:19 AM
sysmembers	dbo	System	11/13/98 3:00:19 AM
sysobjects	dbo	System	11/13/98 3:00:19 AM
syspermissions	dbo	System	11/13/98 3:00:19 AM
sysprotects	dbo	System	11/13/98 3:00:19 AM
sysreferences	dbo	System	11/13/98 3:00:19 AM
systypes	dbo	System	11/13/98 3:00:19 AM
sysusers	dbo	System	11/13/98 3:00:19 AM
TABLE1	dbo	User	5/5/00 11:48:36 AM

Figure 24-13: Creating a new table

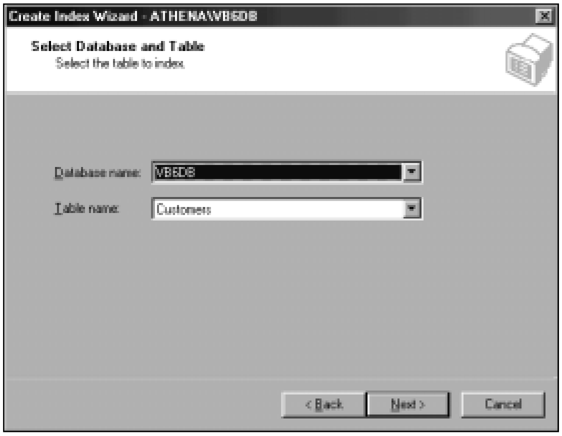
Modifying your table

Once you've finished creating your table, you can always go back and modify it. Simply right click on the Tables icon under the appropriate database icon and choose Design Table (see Figure 24-14). The same window you used to create the table will be displayed containing your table's current definition. Other options on the same pop-up menu will allow you to rename and delete the table. Use these options with care because if your table participates in a foreign key or referenced by a stored procedure, the name change may not be propagated.

Caution

Don't do it: Do not change or delete any of the system tables or you will corrupt your database. If you do, the only way to recover your database will be to completely restore it from a database backup.

As long as your table is empty, you can make any changes you want to the table's definition. However, this isn't true if the table contains data. If you add a column to a table with data, the column's value will be set to **Null** unless you define a default value for the column, in which case the default value will be used. If you delete a column, all of the data stored in the column will be lost. However, you will be prevented from deleting the column if it is part of an index, used in a constraint, part of a **Default** definition, bound to a rule, or used as part of a full-text index.



Create Index Wizard - ATHENA\VB6DB

Select Database and Table

Select the table to index.

Database name: VB6DB

Table name: Customers

Figure 24-14: Selecting a table

You can change certain attributes of a column, such as making a column that previously rejected **Nulls** to accept **Null** values, without impacting the data already in the table. However, most changes will force Enterprise Manager to process each row physically and cause the table to apply the change. If you switch data types, Enterprise Manager will attempt to convert the value from the old data type to the new data type. Likewise, if you change the precision or length, Enterprise Manager will also have to perform the appropriate conversion.

Tip

Safety first: Before you make a change to a table, make sure that you have a good backup copy of your database before you begin. No matter how careful you are you can always recover your database to the point before you made your first change.

Tip

A better way to reformat a table: If you must change the data type for a column, and you're not comfortable with how Enterprise Manager will perform the function, consider using the Data Transformation Services. It allows you to code a VBScript macro to handle the actual conversion process.

Cross-Reference

See Chapter 23 for more information about DTS.

Indexes and Diagrams

The database server uses indexes to locate information in your database more quickly than they could be found without indexes. Whether indexes are present or not is irrelevant to the logical design of the database. They only affect the database's performance.

Not only do database diagrams provide a logical view of your database design, they provide you with the capability to design your database interactively. Most of the functions to create and modify objects that are available to you in Enterprise Manager are also available in the database diagram facility. This means that you don't have to learn yet another set of database design tools.

Creating an index

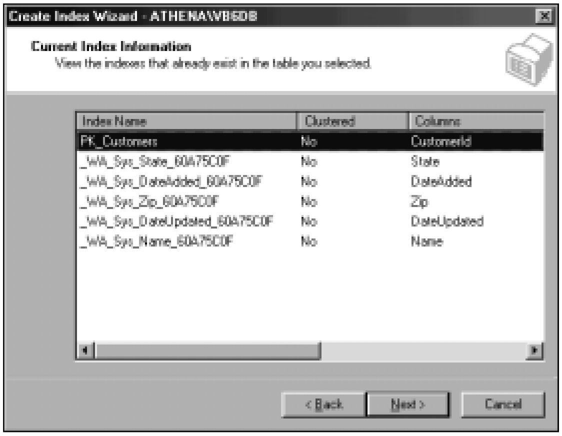
After creating your tables, the next step is to add indexes to your tables. Without indexes, SQL Server will have to scan every record in the table to find the rows you specify in a query. With an index, SQL Server can go directly to the rows you specify, assuming of course that your query took advantage of the index.

There are two main ways to create an index. The first is to create the index manually by using the Create Index wizard, while the second is to run the Index Tuning wizard that examines a typical set of queries to determine the optimal set of indexes.

Running the Create Index wizard

To run the Create Index wizard, follow these steps:

1. Expand the icon tree to expose the database where you want to add the index. Then select Wizards from the Tools menu. Choose Database ➪ Create Index Wizard to start the Create Index wizard.
2. Press Next to move from the introduction screen to the first step in the wizard, which asks you to select the database and table to be indexed (see Figure 24-15). By default, the currently selected database and the first user table in the database will be displayed.



Create Index Wizard - ATHENA\VB6DB

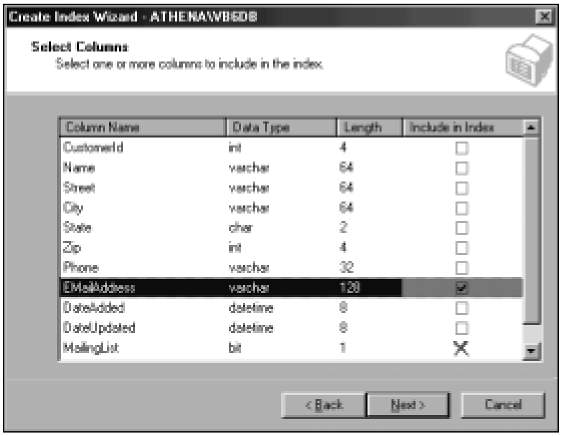
Current Index Information

View the indexes that already exist in the table you selected.

Index Name	Clustered	Columns
PK_Customers	No	CustomerId
_WA_Sys_State_60A75C0F	No	State
_WA_Sys_DateAdded_60A75C0F	No	DateAdded
_WA_Sys_Zip_60A75C0F	No	Zip
_WA_Sys_DateUpdated_60A75C0F	No	DateUpdated
_WA_Sys_Name_60A75C0F	No	Name

Figure 24-15: Selecting the database and table to be indexed.

3. In the next step of the wizard, all of the indexes associated with the table will be listed, along with the columns indexed (see Figure 24-16), Note that the primary key should always have an index and any foreign key constraints will also have an index. You should make sure that you don't duplicate an index that already exists.



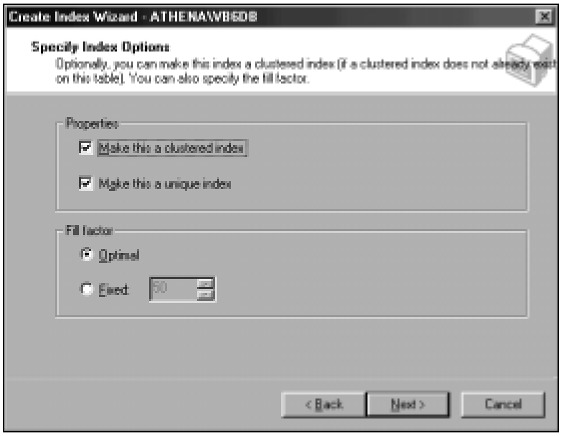
Select Columns

Select one or more columns to include in the index.

Column Name	Data Type	Length	Include in Index
CustomerId	int	4	□
Name	varchar	64	□
Street	varchar	64	□
City	varchar	64	□
State	char	2	□
Zip	int	4	□
Phone	varchar	32	□
EMailAddress	varchar	128	☑
DateAdded	datetime	8	□
DateUpdated	datetime	8	□
MailingList	bit	1	✗

Figure 24-16: Viewing the existing indexes

4. After viewing the existing indexes, you need to choose the column or columns that will make up your new index by placing a check mark in the appropriate box (see Figure 24-17). Note that some columns may have data types that can't be indexed, in which case a big red X will be displayed in place of the check box.



Specify Index Options

Optionally, you can make this index a clustered index (if a clustered index does not already exist on this table). You can also specify the fill factor.

Properties

☑ Make this a clustered index

☑ Make this a unique index

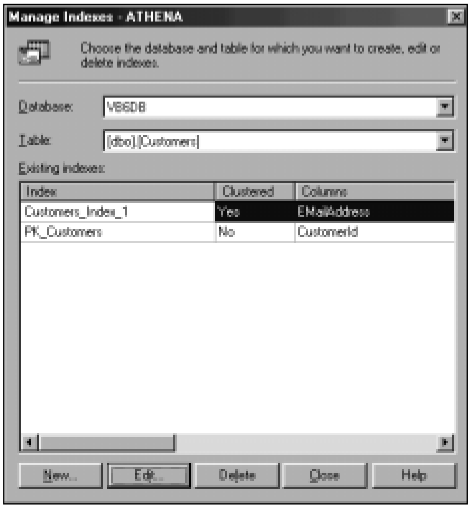
Fill factor

● Optimal

○ Fixed: 50

Figure 24-17: Selecting columns for the index

5. In the next to last step of the wizard, you can specify some options for the index (see Figure 24-18). By selecting Make this a clustered index, you can instruct SQL Server to keep the rows in the table in the same physical order as the index. This can improve performance dramatically when you retrieve a set of records that have a common key value. Only one clustered index is permitted per table however. You can ensure that only one row in the table can have a particular value in the key by selecting Make this a unique index. You can also influence the performance of the index by specifying how much space, or fill, SQL Server should leave in the individual index pages for adding new index values. Unless you really understand how this works, you should choose Optimal Fill Factor.


Manage Indexes - ATHENA

Choose the database and table for which you want to create, edit or delete indexes.

Database: VB6DB

Table: [dbo].[Customers]

Existing indexes:

Index	Clustered	Columns
Customers_Index_1	Yes	EMailAddress
PK_Customers	No	CustomerId

Figure 24-18: Specifying index options

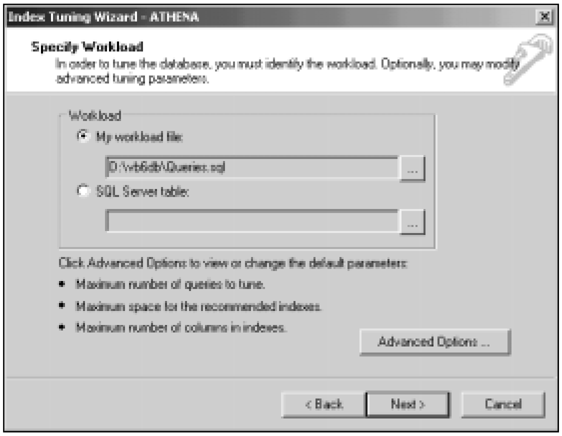
6. In the last step of the wizard, you assign the index a name and adjust the order of the columns included in the table. When you're finished, press Finish to create the index. Since the index is created in real-time, you may have to wait a few minutes for this process to finish, depending on the size of your table and the speed of your database server. A message box will be displayed when the index has been created.

Note

Order in the index: The order of the columns in the index is important, since an index on the columns A, B, and C can be searched three different ways: on column A by itself, on columns A and B, and on columns A, B and C. This index is useless if you want to search on columns B, C, or B and C together, since it is impossible to locate a particular value for B or C without searching through the entire index.

Managing your indexes

You can get a list of the indexes you created for a particular table by right clicking on a table in the Details window and choosing All Tasks ➪ Manage Indexes from the pop-up menu. The Manage Indexes dialog box will be displayed, as shown in Figure 24-19. This window will display the set of the user created indexes. Using the buttons at the bottom of the dialog box, you can create a new index, edit, or delete an existing index.



Index Tuning Wizard - ATHENA

Specify Workload

In order to tune the database, you must identify the workload. Optionally, you may modify advanced tuning parameters.

Workload

● My workload file: D:\vb6db\Queries.sql

○ SQL Server table:

Click Advanced Options to view or change the default parameters:

- Maximum number of queries to tune.
- Maximum space for the recommended indexes.
- Maximum number of columns in indexes.

Figure 24-19: Displaying the Manage Indexes dialog box

Running the Index Tuning Wizard

Choosing the proper indexes for a database can be more of an art than a science. Trying to achieve a balance between too many indexes and too few is difficult. With too few indexes, you will waste time trying to retrieve rows from your table, while too many indexes increases the work needed to insert a new row.

The Index Tuning Wizard will analyze trace data collected by the SQL Server Profiler or a list of specific queries that you specify. It also takes into consideration the current set of indexes. Then, based on its analysis, the Index Tuning Wizard will make a list of recommendations that you can apply immediately or save to a disk file to be applied later with Query Analyzer.

To create the workload file, enter the SQL Queries you wish to optimize into a normal text file using Notepad. Make sure you save the file with a file type of `.SQL`.

To capture trace information, run the SQL Server Profiler utility and follow these steps:

1. Select New ➪ Trace from the main menu to display the Trace Properties dialog box.

2. Enter a name for the trace in the Trace Name field and select the name of the database server where the trace information will be generated on the General tab. Then specify the name of a file or database table where the trace information will be stored.

3. On the Events tab, add the TSQL event to the list of selected events and press OK to start the trace.

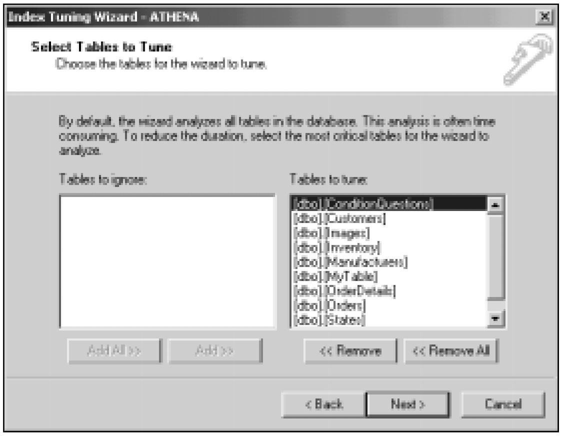
Note

Tracing can be hazardous to your database server's performance: Running a trace increases the amount of work for your database server to perform, and it may adversely affect your server's performance.

To analyze trace data or analyze a few specific queries, follow these steps:

1. In Enterprise Manager, select Wizards from the Tools menu, then select the Management ➪ Index Tuning Wizard and press OK. After the initial screen of the wizard is displayed, press Next to begin the tuning process.

2. On the Select Server and Database step of the wizard (see Figure 24-20), select the name of the database server and database you wish to analyze. Also, you need to choose whether you want to keep your existing indexes. If you're only analyzing a few problem queries, you should keep your current indexes, but if you are analyzing a large volume of trace data, then you may want to replace your current indexes. Choose Perform thorough analysis if you want to choose the optimal set of indexes. Note that this option will significantly increase the amount of time required to run the analysis.



Index Tuning Wizard - ATHENA

Select Tables to Tune

Choose the tables for the wizard to tune.

By default, the wizard analyzes all tables in the database. This analysis is often time consuming. To reduce the duration, select the most critical tables for the wizard to analyze.

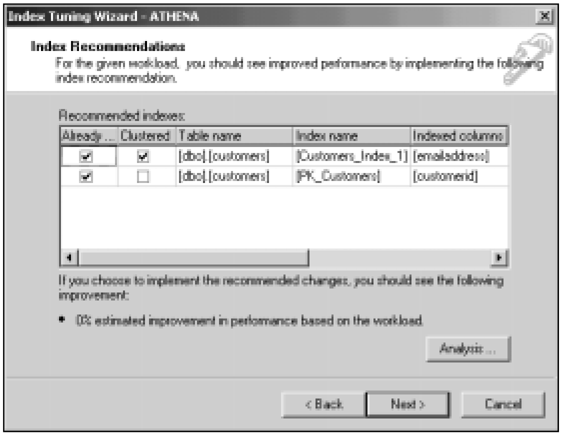
Tables to ignore:

Tables to tune:

[dbo].[ConditionQuestions]
[dbo].[Customers]
[dbo].[Images]
[dbo].[Inventory]
[dbo].[Manufacturers]
[dbo].[MyTable]
[dbo].[OrderDetails]
[dbo].[Orders]
[dbo].[States]

Figure 24-20: Select the database server and database you want to analyze

3. In the next step of the wizard, save a workload file and press Next. Then you will be prompted for the name of your workload file or the name of the database table where the trace information is stored (see Figure 24-21).



Index Tuning Wizard - ATHENA

Index Recommendations

For the given workload, you should see improved performance by implementing the following index recommendation.

Recommended indexes:

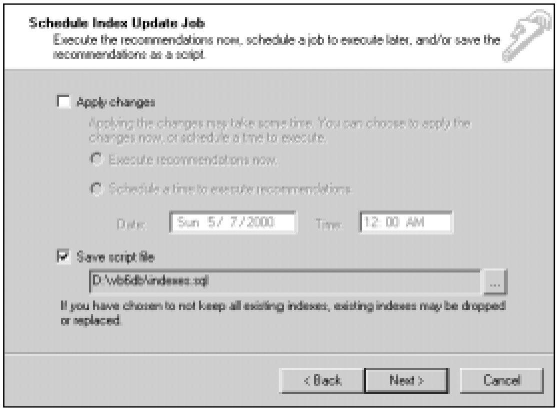
Already ...	Clustered	Table name	Index name	Indexed columns
☑	☑	[dbo].[customers]	[Customers_Index_1]	[emailaddress]
☑	□	[dbo].[customers]	[PK_Customers]	[customerid]

If you choose to implement the recommended changes, you should see the following improvement:

- 0% estimated improvement in performance based on the workload.

Figure 24-21: Specify the location of your workload

4. You can choose the tables you wish to tune in the next step of the wizard (see Figure 24-23). Unless you are worried about the amount of time to run the analysis or want to focus on a subset of the tables, you should select all of the tables in the database.



Schedule Index Update Job

Execute the recommendations now, schedule a job to execute later, and/or save the recommendations as a script.

□ Apply changes

Applying the changes may take some time. You can choose to apply the changes now, or schedule a time to execute.

○ Execute recommendations now.

○ Schedule a time to execute recommendations.

Date: Sun 5/ 7/2000 Time: 12: 00 AM

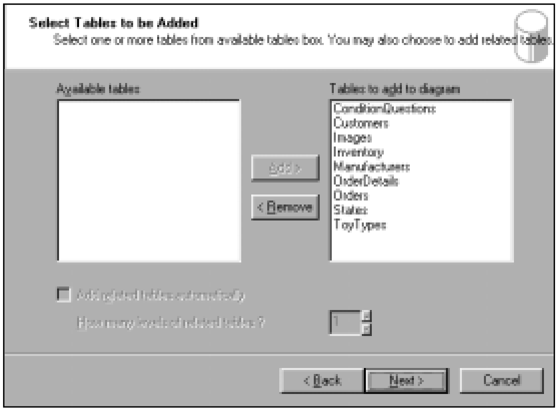
☑ Save script file

D:\vb6db\indexes.sql

If you have chosen to not keep all existing indexes, existing indexes may be dropped or replaced.

Figure 24-22: Selecting the tables to tune

5. Pressing Next will start the analysis process. A dialog box will be displayed that tracks the analysis process. When the analysis is complete, the results will be displayed in the Index Recommendations step shown in Figure 24-23. Pressing the Analysis button will show the detailed results of the analysis.



Select Tables to be Added

Select one or more tables from available tables box. You may also choose to add related tables.

Available tables:

Tables to add to diagram:

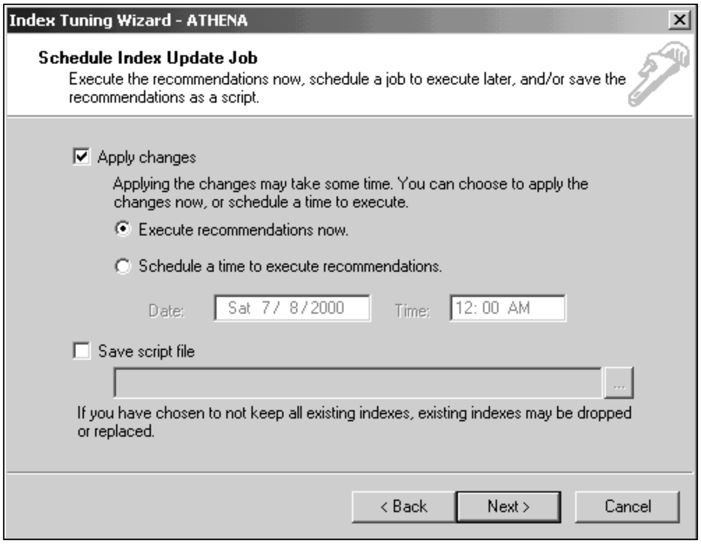
ConditionQuestions
Customers
Images
Inventory
Manufactuers
OrderDetails
Orders
States
ToyTypes

□ Add related tables automatically

How many levels of related tables? 1

Figure 24-23: Reviewing the index recommendations

6. In the next to last step of the wizard, you can choose to apply the recommendations immediately, schedule them to be applied as a batch job, or save the recommendations as a script file that you can apply with Query Analyzer (see Figure 24-24). After making your choice, press Next to display the final step of the wizard, and Finish to complete the process.



Index Tuning Wizard - ATHENA

Schedule Index Update Job

Execute the recommendations now, schedule a job to execute later, and/or save the recommendations as a script.

☑ Apply changes

Applying the changes may take some time. You can choose to apply the changes now, or schedule a time to execute.

● Execute recommendations now.

○ Schedule a time to execute recommendations.

Date: Sat 7/ 8/2000 Time: 12: 00 AM

□ Save script file

If you have chosen to not keep all existing indexes, existing indexes may be dropped or replaced.

Figure 24-24: Choosing how to apply the recommendations

Creating a database diagram

Another useful database object you can create with Enterprise Manager is a *database diagram*. A database diagram contains a logical view of your database similar to an Entity/Relationship diagram. You can add new tables directly in the database diagram, as well as modify existing tables. You can also create relationships between tables by dragging and dropping column references. It is a powerful tool that makes it easy to design your database by allowing you to visualize your database and the relationships between its tables.

Cross-Reference

Refer to Chapter 3 for a discussion of Entity/Relationship diagrams.

Creating your database diagram

To create a database diagram, follow these steps:

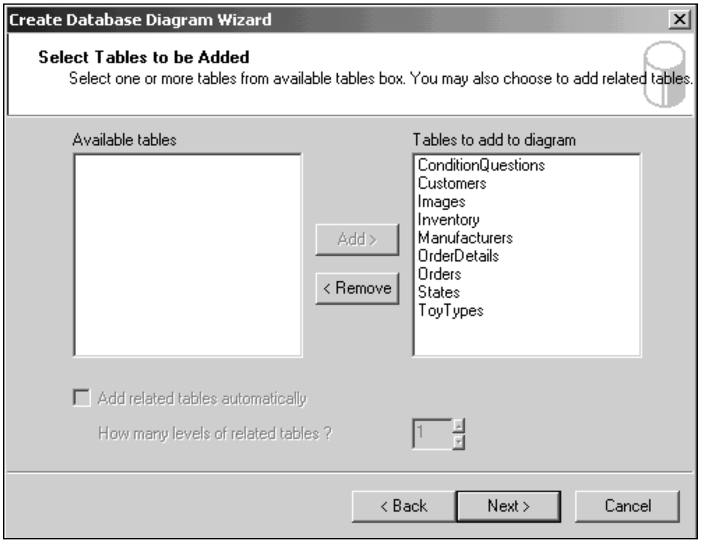
1. Expand the icon tree to select the database server and database where you want to create the database diagram. Right click on the Diagrams icon and choose New Database Diagram from the pop-up menu. This will create a blank database diagram and start the Create Database Diagram Wizard. Press Next to begin creating your database diagram.

2. In the Select Tables to be Added step, you can choose which tables in the database will be included in your database diagram (see Figure 24-25). Sometimes it's useful to have a diagram that includes only a small subset of the tables in the database that focus on a particular function. Checking the Add related tables button, and then adding the table you want to focus on, will automatically add all of the tables that have a relationship to the original table. As tables are added to the Tables to add to diagram list, they will be removed from the Available tables list.

3. In the last step of the wizard, you see the list of tables you selected. Press Finish to add the tables to the diagram and automatically arrange them. Behind the wizard, you'll see your tables appear and then be arranged. When the diagram is complete, a message box will appear and you'll be left with a brand new database diagram to edit and save (see Figure 24-26).

Using the database diagram

Once your database diagram is complete, you should notice that each table is similar to the form you used to design a table discussed in the Creating a Table section earlier in this chapter. That's because it is the same, but only the first column from the form is shown in the diagram. You change the columns displayed by right clicking on the table and selecting one of the following views: Column Properties, Column Names, Keys, Name Only, and Custom View from the pop-up menu. To choose the information displayed in Custom View, choose Modify Custom View.



Select Tables to be Added

Select one or more tables from available tables box. You may also choose to add related tables.

Available tables

Tables to add to diagram

ConditionQuestions
Customers
Images
Inventory
Manufacturers
OrderDetails
Orders
States
ToyTypes

□ Add related tables automatically

How many levels of related tables ? 1

Figure 24-25: Selecting the tables for the database diagram

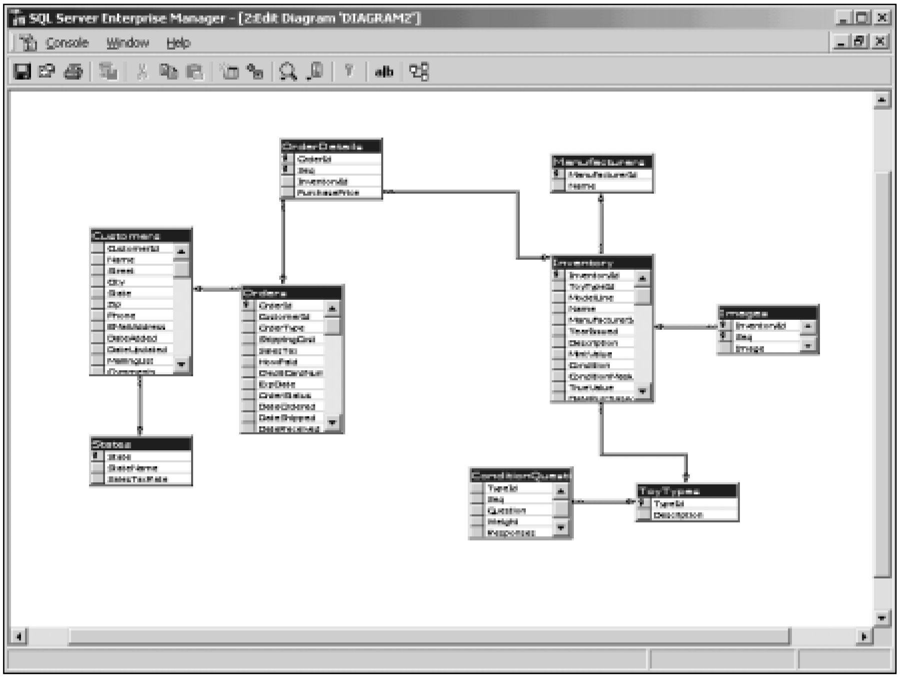
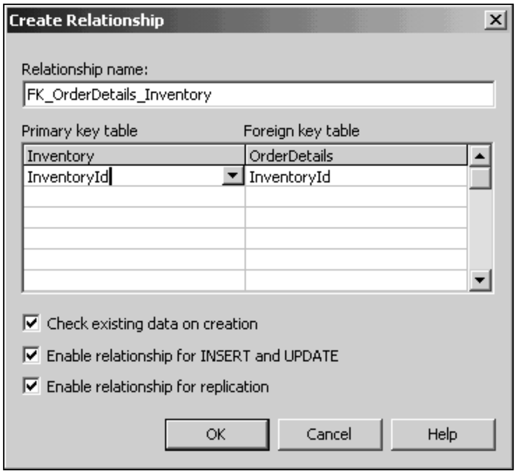


Figure 24-26: Viewing your new database diagram

The second thing you should notice about the diagram are the lines drawn between the tables. These lines represent foreign key constraints. To define a foreign key constraint, click on the prefix area in front of a column in one table, drag it to the other table, and drop it. A dialog box will be displayed that shows both the primary and foreign key tables and the columns that form the foreign key (see Figure 24-27). You can then choose the columns in each table that reflect the columns that comprise the foreign key.



Create Relationship

Relationship name: FK_OrderDetails_Inventory

Primary key table	Foreign key table
Inventory	OrderDetails
InventoryId	InventoryId

☑ Check existing data on creation

☑ Enable relationship for INSERT and UPDATE

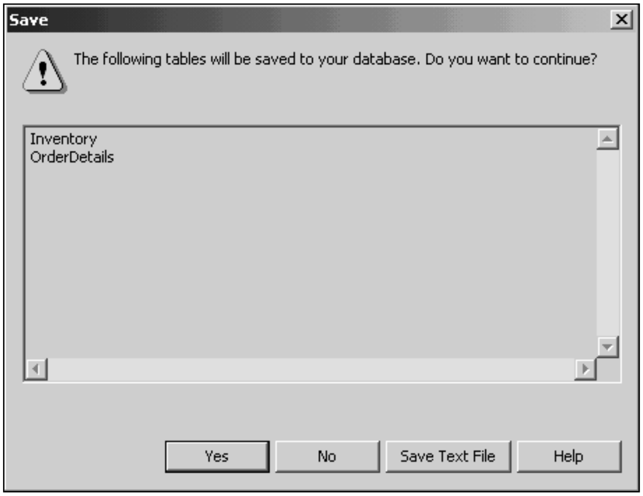
☑ Enable relationship for replication

Figure 24-27: Creating a foreign key relationship

One nice thing about database diagrams is that you can print them. When programming, I usually have a printed copy of the database diagram handy to refer to table and column names and to identify relationships. Since a database diagram can get rather complicated, and may not fit on a single sheet of paper, you can zoom the diagram to create the best fit.

You can also drag the tables around on the diagram to arrange them the way you want. You should notice that the foreign key constraints will automatically follow the table. If you don't like where the constraints are placed, you can simply drag them around until you are happy with them.

As you make changes to your database diagram, you are not actually making any changes to the database. The diagram tool tracks the changes and applies them only when you save the changes or exit the diagram. In either case, a dialog box will be displayed listing the tables that you have changed (see Figure 24-28). You have the option to make the changes, discard the changes, or create an SQL script file containing the statements necessary to make the changes.



Save

The following tables will be saved to your database. Do you want to continue?

Inventory
OrderDetails

Figure 24-28: Saving the changes to the database diagram

Managing Security

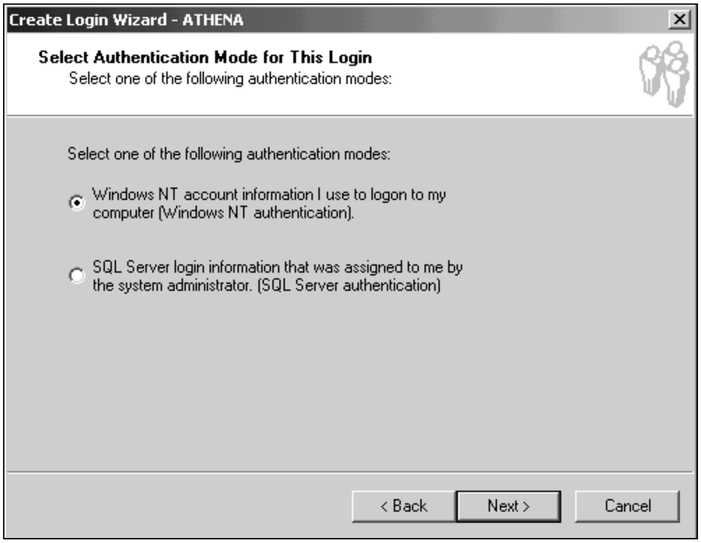
Managing security is both easy and hard at the same time. Understanding the commands you use to apply security is easy. Making sure that you don't have any holes in your security is difficult.

Creating a login

As you might expect, Enterprise Manager includes wizards for creating logins. The Create Login Wizard not only creates a login id, but it also lets you pick database server roles and create a user with the same name in one or more databases.

To create a login, follow these steps:

1. Choose Tools ➪ Wizards and then select Database ➪ Create Login Wizard. Press OK to start the wizard. After skipping over the introduction to the wizard by pressing Next, you'll be prompted to choose the authentication mode for the user (see Figure 24-29). Select the desired authentication method and press Next.
2. If you chose Windows NT Authentication, you'll see the form shown in Figure 24-30. Specify the domain name and the account for the user and then either grant access to the database server or deny access. Press Next to continue.



Create Login Wizard - ATHENA

Select Authentication Mode for This Login

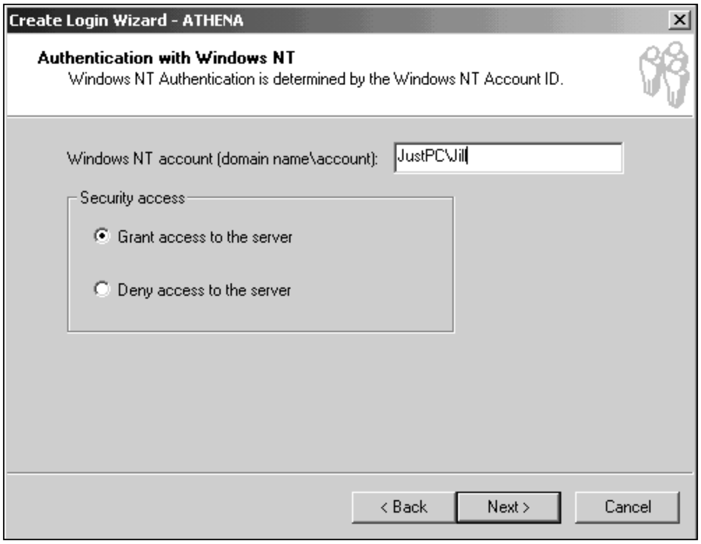
Select one of the following authentication modes:

Select one of the following authentication modes:

● Windows NT account information I use to logon to my computer (Windows NT authentication).

○ SQL Server login information that was assigned to me by the system administrator. (SQL Server authentication)

Figure 24-29: Choosing an authentication mode for your login



Create Login Wizard - ATHENA

Authentication with Windows NT

Windows NT Authentication is determined by the Windows NT Account ID.

Windows NT account (domain name\account): JustPC\Jill

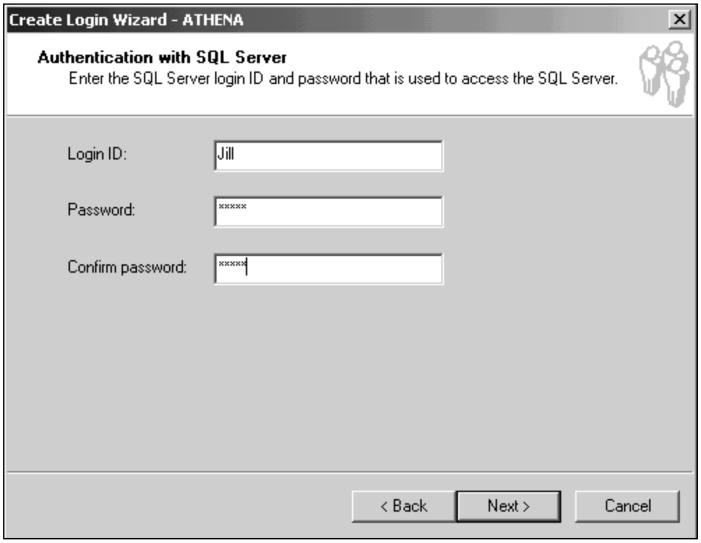
Security access

● Grant access to the server

○ Deny access to the server

Figure 24-30: Entering information for a Windows NT Authentication login

3. If you chose SQL Server Authentication, the form shown in Figure 24-31 will be that you have to enter the password information twice to ensure that the password was entered properly.



Create Login Wizard - ATHENA

Authentication with SQL Server

Enter the SQL Server login ID and password that is used to access the SQL Server.

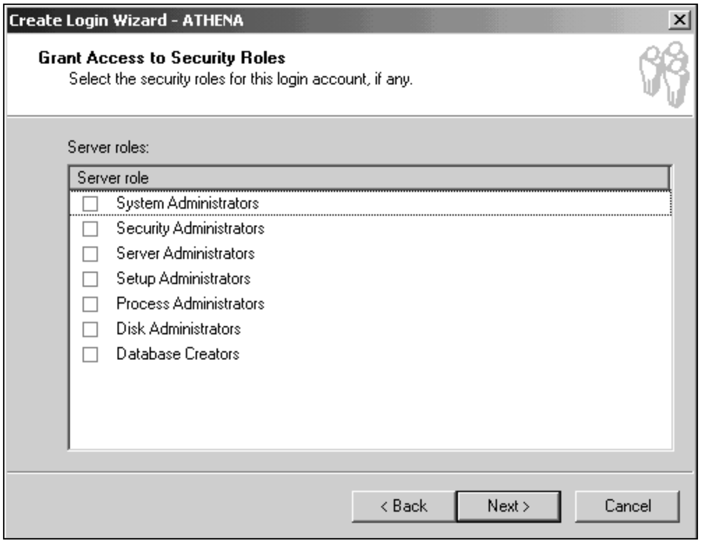
Login ID: Jill

Password: *****

Confirm password: *****

Figure 24-31: Entering information for an SQL Server Authentication login

4. No matter which authentication method you choose, the next step will ask you to choose which server security roles should be assigned to the login (see Figure 24-32).



Create Login Wizard - ATHENA

Grant Access to Security Roles

Select the security roles for this login account, if any.

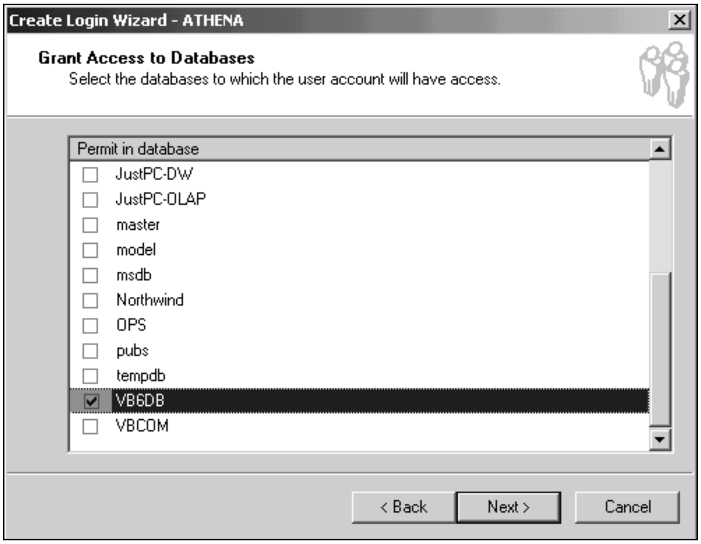
Server roles:

Server role

□ System Administrators
□ Security Administrators
□ Server Administrators
□ Setup Administrators
□ Process Administrators
□ Disk Administrators
□ Database Creators

Figure 24-32: Choose server security roles for your new login

5. In the next to last step of the wizard, you'll be prompted to select the databases in which the new login should be granted access (see Figure 24-33). Press Next to review the information you entered and press Finish to create the login. Note that a user with the same name as the login will be created in each database you selected.



Create Login Wizard - ATHENA

Grant Access to Databases

Select the databases to which the user account will have access.

Permit in database

□ JustPC-DW
□ JustPC-OLAP
□ master
□ model
□ msdb
□ Northwind
□ OPS
□ pubs
□ tempdb
☑ VB6DB
□ VBCOM

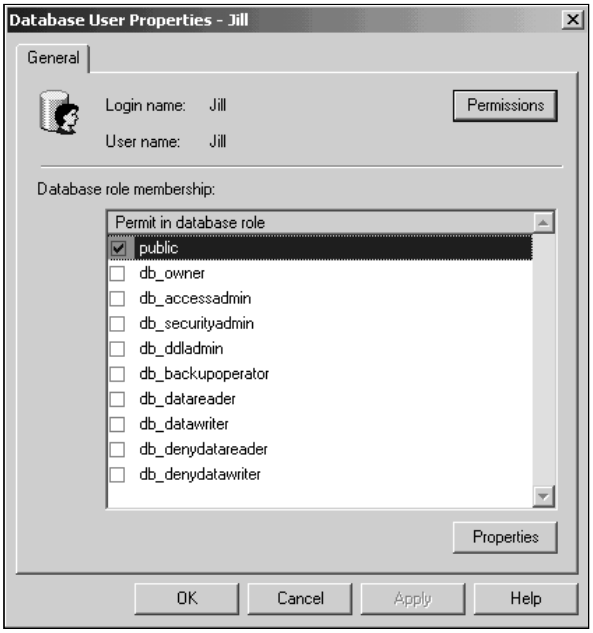
Figure 24-33: Grant your new login access to the databases on the server

Tip

What did I do?: You can see the list of logins on the database server by expanding the icon tree for the database server to show Logins, which is found under the Security icon. Selecting this icon will display all of the logins in the Details window. The list of users for a database can be displayed in the Details window by selecting the Users icon under the appropriate database icon.

Granting permissions in a database

Just because you have granted a user access to a database doesn't mean that they have complete access to your database. By default, they will be assigned to the public role, which means that any resources granted to the public will automatically be granted to the new user. You can review and change their assigned roles by expanding the icon tree to show the Users icon beneath the database you wish to manage. Then, by right clicking and choosing Properties from the pop-up menu, you can specify which roles they have been permitted (see Figure 24-34). On the General tab, you can add new roles by placing a check mark next to the desired role or revoke a role by removing the check mark. You can see the user's specific security permissions by pressing the Permissions button.



Login name: Jill

User name: Jill

Database role membership:

Permit in database role

☑ public
□ db_owner
□ db_accessadmin
□ db_securityadmin
□ db_ddladmin
□ db_backupoperator
□ db_datareader
□ db_datawriter
□ db_denydatareader
□ db_denydatawriter

Figure 24-34: Reviewing security roles in a database

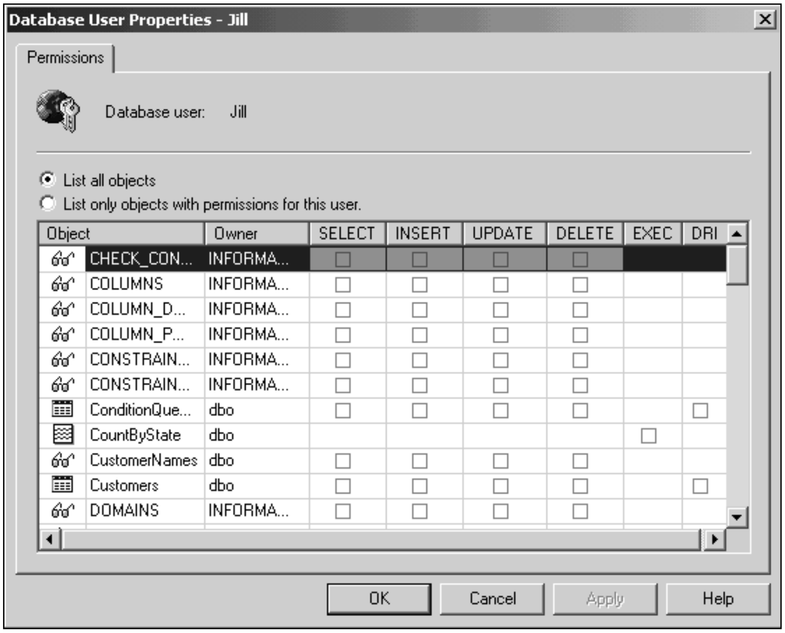
By clicking on the List only objects with permissions for this user on the Permissions tab, you can quickly determine which database objects, if any, this user has access to. The check marks indicate which commands the users may issue against which database objects.

You can modify the permissions associated with a role in the same way you modify them for a user. The only difference is that you have to select the role from the list of roles displayed in the Details window when you select the Roles icon for the desired database in the tree view area of Enterprise Manager.

A list of users assigned to that particular role will be displayed in the Role Properties dialog box (see Figure 24-36). You can then display the same detailed Security Permissions dialog box shown in Figure 24-35 by pressing the Permissions button. You can also get a list of users assigned to the role by selecting a role in the User Properties dialog box and pressing the Properties button.

Tip

Too much security can cause problems: While you can modify the permissions for each individual user, you will be much better off assigning the user to a role and modifying the permissions for the role. When you maintain security permissions at the user level, it is very easy to make a mistake and grant the user permission to something they should be denied. While it is still possible to do this with a role, double checking the roles will be a much easier job than double checking each user.



Database User Properties - Jill

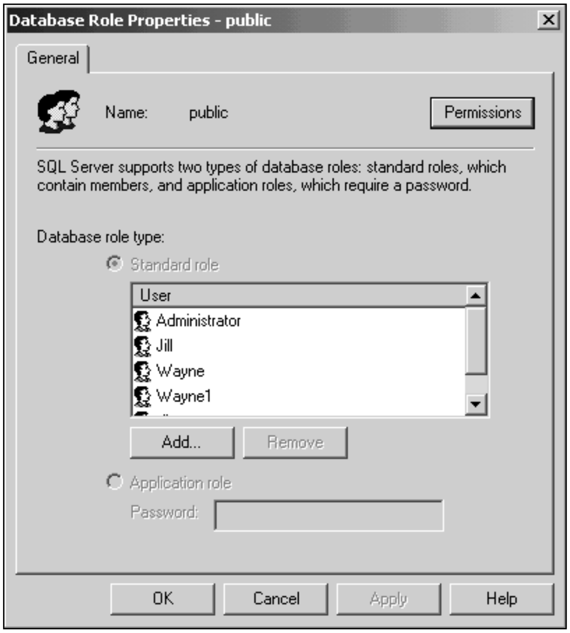
Permissions

Database user: Jill

● List all objects
○ List only objects with permissions for this user.

Object	Owner	SELECT	INSERT	UPDATE	DELETE	EXEC	DRI
CHECK_CON...	INFORMA...	□	□	□	□		
COLUMNS	INFORMA...	□	□	□	□		
COLUMN_D...	INFORMA...	□	□	□	□		
COLUMN_P...	INFORMA...	□	□	□	□		
CONSTRAIN...	INFORMA...	□	□	□	□		
CONSTRAIN...	INFORMA...	□	□	□	□		
ConditionQue...	dbo	□	□	□	□		□
CountByState	dbo					□	
CustomerNames	dbo	□	□	□	□		
Customers	dbo	□	□	□	□		□
DOMAINS	INFORMA...	□	□	□	□		

Figure 24-35: Reviewing the detailed security permissions for a user



Database Role Properties - public

General

Name: public

SQL Server supports two types of database roles: standard roles, which contain members, and application roles, which require a password.

Database role type:

● Standard role

User
Administrator
Jill
Wayne
Wayne1

○ Application role

Password:

Figure 24-36: Reviewing the properties of a role

Thoughts on Enterprise Manager

Enterprise Manager is a very powerful tool that simplifies the administration of an SQL Server database. In this chapter, I've barely begun to communicate what you can do with Enterprise Manager. I strongly recommend that you take the time to explore its capabilities, especially if you plan to do a lot of work with SQL Server.

If you look carefully, many of the tools that are used in Enterprise Manager look like those available in the Enterprise Edition of Visual Basic. This is no coincidence. The same code base is used for both. I feel that Enterprise Manager does a better job of implementing the tools than Visual Basic, but it is nice to be able to use some of the same tools with other databases also.

It's important to note that anything you can do with Enterprise Manager can also be done using the appropriate SQL statements. Enterprise Manager is nothing more than a very complex program that issues ADO requests to an SQL Server database, just like your application would.

Summary

In this chapter you learned:

- about the various features of SQL Server Enterprise Manager.
- how to browse data in your database.
- how to create a new table.
- how to add an index to a table.
- how to create a database diagram.
- how to create a login and assign security permissions.

✦ ✦ ✦