

[Figures are not included in this sample chapter]

MCSE Training Guide: SQL Server 6.5 Administration

- 3 -

Enterprise-Wide Database Administration

Server administration and configuration (that is, enterprise-wide database administration) is accomplished with a number of tools that provide centralized methods by which various aspects of a SQL Server can be controlled. This chapter will focus on the use of each of these tools and their respective functions. Specifically, this chapter covers the following:

- Distributed Management Framework (DMF)
- SQL Enterprise Manager
- SQL Service Manager
- SQL Client Configuration Utility
- SQL Security Manager
- SQL Setup
- SQL Trace
- SQL Performance Monitor
- SQL/W

As always, this chapter begins with a list of the exam objectives relating to the topic at hand, moves on to the chapter pretest, offers valuable information directly addressing the objectives and additional relative discussion topics, and finishes with exercises, chapter review questions, and answers to both the chapter review questions and the chapter pretest questions.

This chapter focuses on enterprise-wide database administration. It helps you prepare for the exam by addressing and fully covering the following objectives:

- Configure Servers in the Enterprise
- Manage Servers in the Enterprise

- Administer Servers in the Enterprise

Since the objectives for this portion of the exam are somewhat high-level, this chapter instead focuses on how the objectives of server management and administration are accomplished. To do this, the content highlights the tools and utilities included with SQL Server that are used to perform common administrative tasks such as database creation, client connectivity configuration, modifying security options, or alert management.

Rather than being organized around the three exam objectives, this chapter is structured by each SQL Server management and administrative utility. The sections are presented in no specific order other than to try to establish a foundation for server management prior to exploring a particular utility in greater detail. All utilities are presented with the exception of the SQL Server Web Assistant; this tool relates more to advanced server functionality rather than server administration, and thus is outside the scope of this chapter.

-
1. What are three methods that can be used to stop SQL Server?
 2. How can I increase the amount of memory SQL Server allocates upon startup?
 3. How can the SQL Server security mode be modified?
 4. How can Windows NT Security be integrated with SQL Server?

Answers are located at the end of the chapter...

Distributed Management Framework (DMF)

With the release of SQL Server 6.0, and now version 6.5, Microsoft introduced the *Distributed Management Framework*, or *DMF*, as the primary means of SQL Server administration. DMF is comprised of a set of methods, services, and components that expose functionality of SQL Server and its related services through common interfaces and applications (see fig. 3.1).

Essentially, DMF provides *OLE (Object Linking and Embedding)* interfaces to SQL services for use by the included administration tools, as well as any user-written VB 4.0 or even MS Excel VB application. These OLE objects are referred to as *Distributed Management Objects*, or *DMOs*. SQL system files that provide this functionality are illustrated in table 3.1. Table 3.1

System Files that Comprise the SQL Server Distributed Management Framework

File Name	Description
SQLOLE.REG	This file contains the required Windows NT Registry Entries for OLE functionality to SQL Server.
SQLOLE65.TLB	This OLE Type Library file is used by Visual Basic Applications.
SQLOLE65.SQL	This file contains Transact-SQL statements required for the execution of SQL stored procedures utilizing DMO functions. It is installed by default during SQL Server setup.

SQLOLE.HLP	This Primary help file contains information on available DMO methods and properties.
------------	--

Figure 3.1 *The SQL Distributed Management Framework exposes common server functions to administrative applications.*

NOTE SQL-DMF contains over 60 objects (Database Management Objects, or DMO) and 1000 methods and properties. Microsoft publishes a number of books on SQL Server Programming through DMO. Note that sample Visual Basic code and Excel 5.0 modules are included in the SQL Server installation. Complete descriptions of these files and their functions can be found by viewing the README.TXT file located in the MSSQL\SQLOLE\SAMPLES directory.

The Distributed Management Framework is important because it is the basis for all tools that are used in SQL Server Management and Administration. Prior to DMF, only tools and utilities supplied with SQL Server could be used in its administration. This restriction made it difficult for end users to customize tools for specific uses or functions. DMF provides increased flexibility; server administration and management can be accomplished with either the supplied tools, or via user-defined applications that take advantage of the DMF framework.

One of the primary advantages of DMF is that it enables the automation many common administrative tasks such as database backup and alerting. The basis of almost all functionality within the SQL Enterprise Manager, the primary SQL Server administrative tool (see next section), is through the use of SQL-DMF.

SQL Enterprise Manager

The SQL Enterprise Manager provides the primary interface used to administer multiple servers within the enterprise. Almost every major management or administrative task can be accomplished through the use of this single application. In some cases, such as invoking queries to the SQL Server, SQL Enterprise Manager actually uses the same tool as would be used when invoking the respective application manually--in this case, ISQL/W.

Not only does Enterprise Manager centralize administration and management, it also allows centralized administration of multiple servers in the enterprise. This is accomplished by registering a SQL Server in the Enterprise Manager. After a server is registered, it's possible to administer just about any SQL Server function, such as the following:

- Starting and stopping SQL services
- Viewing the SQL error log
- Modifying the SQL Server configuration
- Administering SQL mail
- Creating databases and devices
- Adding/modifying logins

- Setting up scheduled tasks and alerts
- Administering SQL replication
- Setting up database maintenance

These tasks represent the most common SQL Server management and administrative requirements. Before this can be accomplished using Enterprise Manager, however, the application must know about each server that will be managed. This is accomplished by registering servers, which is explored next. From this point, you learn how each of the above administrative tasks is accomplished.

Registering Servers

Before a server can be administered using SQL Enterprise Manager, it must be registered. This allows the administrator to define multiple servers, quite possibly in multiple NT domains, to be administered from single location.

A server is registered by using the following steps:

1. Start the SQL Enterprise Manager by double-clicking the SQL Enterprise Manager icon in the Microsoft SQL Server program group. If no servers have been registered previously, the Register Server window will be displayed. Otherwise, the application will already indicate previously registered servers in the Server Display window. A server can also be registered by right-clicking on the default SQL 6.5 server group and selecting the Register Server option from the pop-up window.
2. In the Register Server window, enter the required server name in the Server field. This is usually the NT computer name for the SQL Server.
3. In the Login Information area, select the login method to be used. The Use Trusted Connection option is available only when SQL Server is configured for Integrated Security mode, relying on NT user account information to grant access to SQL Server. Standard Security is the default login method when using Standard or Mixed Security mode, where SQL Server logins are maintained separately from NT.

WARNING If a server has been registered using Standard Security, requiring the sa account password, this information is retained in the NT Registry. This means that anyone can obtain unauthorized access to the SQL Server if he is able to access SQL Enterprise Manager on the given computer.

4. Enter the login ID and password to be used by SQL Enterprise Manager. This must be the *sa* (*system administrator*) account for the given server. Trusted connections do not require login account information.
5. In the Server Group window, the default group of SQL6.5 is displayed. This is where the server will be added when displayed in the Explorer style tree in the Server window of Enterprise Manager.

6. Click the Register button to complete the server registration. The server should now be visible within the SQL 6.5 structure in the Server window. If the server isn't immediately visible, it may be necessary to expand the tree by clicking the + sign next to the SQL 6.5 group.

To manage and administer the server, click the + sign next to the server name to expand its available functions. Each aspect of the server's configuration is now displayed.

TIP Keep in mind that server registration information is maintained in the Windows NT Registry under the SQL Server software key. This means that the information is available when starting Enterprise Manager only on the workstation on which the server was registered. It might be helpful to designate certain machines that will be used for SQL Administration and register the appropriate servers from these machines. This would ensure that all servers could be accessed from a single location or multiple centralized locations. Additionally, registering each server on its own local console aids administration in that Enterprise Manager can easily be used when logging on to the local SQL Server.

Defining Server Groups

Server groups allow various SQL Servers to be presented in logical groupings within SQL Enterprise Manager to aid in administration (see fig. 3.2). For example, suppose you are responsible for administering SQL Servers in each region of the United States: Northeast, Southeast, Midwest, Northwest, and Southwest. Using Enterprise Manager, you could define corresponding server groups so that it's easy to identify the servers within each geographical region. Another method of grouping might be to identify servers for various departments within the enterprise.

Server groups are defined by selecting Server Groups from the Server menu in Enterprise Manager or by right-clicking on the default SQL 6.5 Server group and selecting New Server Group from the pop-up menu. Either of these two methods will display the Manage Server Groups window, as shown in figure 3.3. When adding a server group, you must determine whether the group will be a top-level group, or a sub-group within an already existing group. You determine the level by selecting the appropriate Level option. Enter the desired group name and click the Add button. This will add the new group to the server list in Enterprise Manager. Server groups can be renamed or removed by accessing the Manage Server Groups window and right-clicking on the required group name.

Figure 3.2 *Server groups can help organize SQL Servers to aid administration of multiple servers.*

Figure 3.3 *The Manage Server Groups window is used to organize SQL servers within the organization.*

After the appropriate server groups have been defined, servers can be assigned to various groups simply by highlighting the required server and dragging it to the group.

Starting and Stopping Servers

One of the most common administrative tasks is that of starting and stopping SQL servers, which is often required when invoking server configuration changes or, in some cases, to allow the SQL

Server database files to be backed up while SQL Server is off-line. This can be accomplished by using a number of methods, both from within SQL Enterprise Manager, as well as with other SQL Server tools.

From within Enterprise Manager, SQL Servers can be started and stopped by highlighting and right-clicking the desired server in the Server Manager window. This action displays the Server pop-up window that indicates the available control commands for the SQL Server service. The same command window can also be displayed by selecting SQL Server from the Server menu option in Enterprise Manager. The available commands to control the "state" of SQL Server are detailed in table 3.2. Table 3.2

SQL Server States

State	Description
Stop (or Stopped)	The service is stopped, making SQL Server non-functional. The server cannot be recognized by client workstations.
Start (Started)	Normal operation--the server is fully functional.
Pause	SQL Server is started and active, but no further user connections are possible after it has been paused. Existing users continue to operate and the SQL Server is visible on the network.
Continue	Deactivates the Paused state, resuming normal operation.

The current state of the server determines which actions can be taken to control it. For example, when right-clicking on an active server, the Start option will be grayed out. A server's state is indicated by its associated icon in the Server Display window. The icon uses a stoplight metaphor: a green light indicates a running server, a yellow light indicates a paused server, and a red light indicates a stopped server.

TIP Enterprise Manager makes extensive use of icons not only to indicate the status of a given server, but also to indicate various database objects and related services. To quickly view the meaning of each icon, display the legend by using the Toggle Legend option from the File menu. Just as with a map, the legend provides a floating window that can be used to quickly reference the meaning of various icons within Enterprise Manager. You can also use the display legend toolbar button located next to the Server field in the Server Display window.

Starting and Stopping Related Services

In addition to controlling the SQL Server itself through Enterprise Manager, you can start or stop supplemental SQL services by using the similar methods: either by right-clicking on the desired service in the Server Display window, or by selecting the appropriate option from the Server menu.

Supplemental services are those that provide additional functionality to SQL Server, but are not part of the core database engine (SQLSERVR.EXE). For example, the SQL Executive Service is used to process server alerts and scheduled tasks (see Chapter 7, "Server Alert Processing and Task Scheduling"). Without this service, alert and task processing is not possible; however, the database

engine is still functional and able to process user requests. In this case, the SQL Executive is a supplemental service.

These services follow:

- The SQL Executive
- SQL Distributed Transaction Coordinator (DTC)
- The SQL Mail Service

Figure 3.4 illustrates these services. Figure 3.4 *Supplemental SQL Services can be controlled by using Enterprise Manager.*

SQL supplemental services are started or stopped using the following steps:

1. Highlight the desired server in the Server Display window. (Note that this may require that a server group be expanded by clicking the + sign next to the group name.)
2. Verify that the desired SQL Server state is active by noting the green stoplight shown in the server's icon. The SQL Server must be active in order to control its supplemental services.
3. Click the + sign next to the server to expand the server's properties. The first three items shown are the supplemental services.
4. Highlight the required service--SQL Mail, for example--and right-click it. The server pop-up window will be displayed along with the available states for that service.
5. Select the desired state (Start or Stop) to control the service.

WARNING Some SQL Services such as SQL Mail and SQL DTC require specific configuration settings before they can be started successfully. Other than SQL Executive, which is required for alerting and scheduled database maintenance, don't start services such as SQL Mail or DTC unless you require the specific functionality provided by these services.

Supplemental SQL services are dependent on SQL Server, meaning that they can be started only if SQL Server is already started.

SQL Service Manager

Another tool that can be used to start and stop SQL Services is the SQL Service Manager. The Service Manager tool provides an interface to control SQL Server services on any connected server in the enterprise. This capability is required in order to easily administer multiple servers in the enterprise.

The Service Manager makes it possible to easily control any SQL server in the network, without the need to invoke SQL Enterprise Manager. This is especially useful in situations where administrative personnel must start and stop servers, but are not responsible for additional configuration items that

are performed by using Enterprise Manager.

TIP The SQL Server and SQL Executive services can also be directly controlled using the Windows NT Control Panel's Services applet. Note that SQL Server is listed as the MSSQLServer service.

The SQL Service Manager controls the SQL Server and the SQL Executive services of local or remote servers (see fig. 3.5). SQL Mail and SQL DTC services can be controlled only by using Enterprise Manager.

Figure 3.5 *The SQL Service Manager controls the SQL Server and SQL Executive services.*

Stopping the SQL Server service on a can be accomplished by following these steps:

1. Start the SQL Service Manager by double-clicking the SQL Service Manager icon in the SQL Server program group. In the Server field, enter the name of the local SQL server. When connecting to remote servers, enter the server name and click the Connect button.
2. In the Service field, verify that either the SQL Server or SQL Executive service is indicated. Use the pull-down menu to select the appropriate choice for the service you are administering.
3. Double-click the stoplight icon for the action being performed. For example, double-clicking the red light will stop the selected service. Double-clicking the green light will start the service.

The bottom portion of the SQL Service Manager window indicates the current state of the selected service. Possible states follow:

- The Service is Starting
- The Service is Running
- The Service is Stopping
- The Service is Stopped
- The Service is Paused

Noting the server state is useful to verify when an operation has been completed. Servers maintaining large databases or with high user activity can sometimes take several minutes to complete a shutdown or startup request.

TIP Clicking the system icon in the upper-left corner of the SQL Service Manager window displays the System Menu. By using options on the system menu, you can modify behavior of the application. By selecting the option Remote Control, the Server field will not be displayed; this prevents SQL Service Manager from administering remote servers. The Action Verify option will prompt for verification when starting or stopping services. Use the Polling Interval option to control the frequency at which SQL Service Manager will poll the service to determine its current state. Foreground Polling

Interval is used when the application is active on the desktop. Background Polling
Interval determines the frequency of updates when SQL Service Manager is minimized.

Modifying the SQL Server Configuration

Another common SQL management task is to modify the server's configuration. For example, suppose you want to increase the number of concurrent user connections supported by a given SQL server. Modify the User Connections system parameter, which is part of its overall configuration. SQL system parameters are modified by using the SQL server, Configure option from the Server menu in Enterprise Manager (see fig. 3.6).

Figure 3.6 *SQL Server configuration parameters.*

Because SQL Server allocates memory from Windows NT when it is started, it's essential that the server's "working set" portion of memory is allocated properly. Many configuration options allocate additional memory from the operating system based on how they are configured. Although many SQL operational settings almost never require modification, many common settings come into play often. Following are some examples of these settings:

- **User Connections.** Controls the maximum number of user connections the server will accept.
- **Memory.** Determines the total amount of memory SQL Server will allocate for its use upon startup.
- **Open Databases.** Determines the maximum number of simultaneous open databases.
- **Open Objects.** Determines the maximum number of open database objects for the given server.

To modify the SQL server configuration, use the following steps:

1. Start the SQL Enterprise Manager. Select the required local or remote server in the Server Display window.
2. From the Server menu, select SQL Server and then select Configure from the pop-up window. The Server Configuration Options window appears. (Note that many of the Server options on the first tab can also be controlled using the SQL Setup program.)
3. Click the Configuration tab to access the Server configuration information. This window indicates each configuration option and its minimum, maximum, running, and current values.
4. Highlight the required server option and enter a new value. Note that the value must be within the minimum and maximum range for that setting.
5. Click the Apply Now button to put the change into effect. Depending on which option was changed, this action affects only the current value and not the running value for the configuration option; most configuration options require a restart in order to be put into effect.

6. Click the OK button to return to the Server Display window.

TIP By highlighting a configuration option, you can view the amount of memory allocated to it, as well as additional details about its use, in the Description field at the bottom of the Configuration window.

WARNING Use caution when changing SQL configuration options; incorrectly setting items like Memory can prevent the SQL Server from starting. If this happens, use the SQL Setup program to add the -f startup option to the server, which causes SQL Server to start in a minimal configuration, essentially ignoring the offending configuration option settings. Correct the required settings and restart SQL Server without the -f startup option, to again return to normal operation.

Administering SQL Mail

The SQL Mail service allows SQL Server to send query results as e-mail messages, and accept query requests from user-submitted e-mail messages. SQL Mail also plays a critical role in the capability of SQL Server to forward messages based on events and alerts. Suppose you are responsible for ensuring that when errors are encountered within a given database, notification is sent to the application developer. SQL Mail interacts with SQL Server to forward e-mail messages when data errors are encountered in the database.

To be fair, SQL Mail relies heavily on the ability of the messaging software (Microsoft Exchange or an NT post office) to obtain its configuration information. The configuration requirements for SQL Mail basically consist of providing the proper mail login ID and password, or an Exchange user profile name.

Understanding how to enable and configure SQL Mail plays a vital role in Database Server management. The SQL Mail service can really be considered part of the server's overall configuration. Briefly explore how to configure this capability in the following section.

Configuring SQL Mail

To use SQL Mail, it is essential that Microsoft Exchange Server software first be installed and functional on the SQL Server. The Exchange installation will modify NT Registry parameters, which affects the information displayed when configuring SQL Mail from the SQL Setup program or SQL Enterprise Manager. Follow these steps to configure SQL Mail:

1. In Enterprise Manager, highlight the SQL Mail service on the desired server and right-click it to display the Service pop-up window. Select the Configure option to configure the service. (Alternatively, select the SQL Mail service from the Server menu in Enterprise Manager.)
2. The Exchange Profile box is displayed. Enter the name of the Exchange user profile to be used with SQL Server. Click OK to return to the Server Display window.

WARNING The Exchange user profile for SQL Mail must be set up using the same account that the MSSQLServer is started with in the Windows NT Control Panel. A

good practice is to start MSSQLServer with a valid NT domain account, rather than with the NT LocalSystem unique account. This will not only facilitate SQL Mail, but is vital for additional SQL domain operations, such as replication and database backup/restore to network locations.

Starting SQL Mail

To start SQL Mail, highlight the service, right-click it, and select the Start option. If SQL Mail is properly configured, the icon will change from red to green. If an error message appears when you start the service, consult the SQL Server error log for additional details (use the Error Log menu option from the Enterprise Manager Server menu).

TIP You can also start SQL Mail by executing the XP_STARTMAIL command from ISQL/W or from the SQL Query Tool in the Tools menu of Enterprise Manager.

Stopping SQL Mail

Stopping SQL Mail is accomplished in much the same way as starting the service: right-click the icon in the Server Display window and select Stop from the Service pop-up window.

TIP You can also execute the XP_STOPMAIL command from ISQL/W or from the SQL Query Tool in the Tools menu of Enterprise Manager, to stop the SQL Mail Service.

SQL Client Configuration Utility

One of SQL Server's strengths lies in its capability to be integrated into almost any environment, regardless of the network protocols that are in place, such as TCP/IP or IPX/SPX. In order to provide this functionality, it is necessary to understand how to configure client workstations for the required connectivity method or transport that the SQL Server is supporting. This is accomplished by using the SQL Client Configuration Utility.

The SQL Client Configuration Utility determines the Network Library, or *Net Library* or *Net Lib*, that the NT Workstation or Server will use to communicate with SQL Server. The designated Net Library is dependent on the installed network protocol to provide connectivity to SQL Server. In this way, SQL Server can be integrated in many different network environments.

SQL Server Network Library Types

Prior to understanding how to change a client's Network Library, it is beneficial to review some background information about the available Network Libraries and the best situations for their use.

SQL Server Net Libraries determine which network transport clients are used in connection with the SQL Server. Depending on the Net Library used, certain SQL functionality may or may not be available. Table 3.3 describes each Net Library and its characteristics. Table 3.3

Defining SQL Server Network Library Types

Net Library	Description
Named Pipes	The default communication method for SQL Server, Named Pipes requires no additional configuration changes to the client or server. Named Pipes must be used to take advantage of integrated NT and SQL Security. This Net Library is common in pure Microsoft networking environments, and is always available on the server, regardless of additional Net Libraries that may be in use.
Multi-Protocol	This Net Library uses <i>RPCs (Remote Procedure Calls)</i> to communicate with SQL Server and does not require additional configuration parameters. Multi-Protocol clients can take advantage of encryption and integrated security with any installed protocol, such as Novell clients using IPXODI or SPX.
NWLink IPX/SPX	The NWLink Net Library utilizes the SPX network protocol to communicate with SQL Server. It is most commonly used in Novell Netware environments so that workstations can communicate with SQL Server without additional network transports being installed. Be sure to install the IPX/NWLink transport on the NT Server first.
TCP/IP Sockets	Clients using this Net Lib communicate with SQL by using native TCP/IP socket connections. By default, SQL Server listens for TCP/IP connections on Port 1433. This Net Lib is ideal for mixed environments or those requiring SQL connectivity with the Internet. Note that integrated NT/SQL security cannot be used with TCP/IP Sockets clients.
Banyan VINES	The VINES Net Lib enables clients to connect to SQL Server by using the <i>VINES SPP (Sequenced Packet Protocol)</i> . To use this Net Library, you must install the VINES protocol on both the SQL server and the client workstation.

Advanced Client Options

To use Net Library communication methods other than Named Pipes, support for the appropriate communication method must be installed on the server using the SQL Setup program. Depending on the Net Library used, both SQL Server and the client workstation might require that specific network protocols be installed. After the required protocols are installed, it may be necessary to set up an alias for the client to use in connecting to SQL Server using the alternative communication method.

For a client to use TCP/IP Sockets, for example, the SQL server's DNS name would be referenced in the alias configuration. The following steps illustrate how to set up a client to use TCP/IP Sockets:

1. Start the SQL Client Configuration Utility. The DB-Library tab appears.
2. Select the Advanced tab to access the advanced client options (see fig. 3.7).
3. In the Client Configuration section, enter a server name in the Server field. (The pull-down option is used for modifying existing entries only.) This is the server name that is used to connect to the server using the alternative Net Library.
4. In the DLL Name field, select the appropriate Net Library. For the purposes of this example, the Net Library would be TCP/IP Sockets.

5. In the Connection String field, enter the required connection information. This information will vary depending on the chosen Net Library. For TCP/IP connections, the format is the IP address (or DNS Name) for the SQL Server followed by the port number:
My_DNSServername, 1433.

6. Click the Done button to complete the configuration and add the entry to the Current Entries section.

To modify existing entries, highlight the appropriate entry in the Current Entries section and change the Net Library information as required. When the edits are complete, click the Add/Modify button to modify the existing entry.

NOTE The name entered in the Server Name field is a logical definition used to reference the SQL server. This value can be anything you want; however, you should avoid using the NT computer name, if possible, to eliminate confusion in situations in which NT clients can connect by using either Named Pipes or TCP/IP. A good practice is to make the name unique to the given Net Library, such as Servername_IP or Servername_IPX.

Figure 3.7 *The SQL Client Configuration Utility shows an alias for a TCP/IP Sockets Connection.*

Changing the Default Net Library

By using the Client Configuration Utility, you can choose which Net Library method the workstation will use by default. The default automatically substitutes the correct *.DLL file for the required communication method. This automatic substitution is useful in environments in which the default method of Named Pipes is not possible. For connections to SQL Server in a Novell Netware environment, for example, NwLink IPX/SPX will most likely be the method of choice. To change the default Net Library, follow these steps:

1. Start the SQL Client Configuration Utility.
2. Select the Net Library tab to display the Net Library configuration. Use the pull-down box in the Default Network field to highlight the desired Net Library.
3. Press the Done button to complete the Default Network change.

SQL Security Manager

The SQL Security Manager is used to administer the selected SQL Server security mode (see fig. 3.8). In short, SQL Server security can be integrated with already-established security in NT domain user accounts, or it can be maintained separately, providing greater flexibility for application development and security integration. It is important to understand the implications that each method brings to the table. More importantly, when managing multiple servers in the enterprise, each SQL server can be configured to support different security configurations.

This section details the available security methods and how the SQL Security Manager utility can be used to modify the server security configuration.

SQL Server provides the capability to support multiple methods for login and security. Different security methods enable SQL Server to integrate tightly with existing Windows NT security, or to maintain its own separate security mechanisms for database login. Table 3.4 outlines each of the SQL Security modes and situations in which each might be used. Table 3.4

SQL Server Security Modes

Mode	Description
Standard	SQL Server user accounts are maintained separately from existing Windows NT domain accounts. This is the default mode, which is desirable for applications that require specific security settings or environments in which access to Windows NT resources is not desired.
Integrated	In Integrated mode, SQL Server dynamically maps SQL user IDs to NT domain accounts. After an NT account is created, no additional action is required on the SQL Server for the user to obtain access. This mode is useful for minimizing NT and SQL account administration.
Mixed	Mixed security mode is a combination of Standard and Integrated modes. It is useful in mixed environments where some client workstations are unable to take advantage of Windows NT security.

Figure 3.8 *The SQL Security Manager.*

Although SQL security administration is accomplished by using the SQL Security Manager, selecting the supported security mode is accomplished through the use of the SQL Setup Program, which is discussed later in this chapter.

Using SQL Security Manager, NT user accounts can be defined as either SQL user accounts or SQL *sa* (*system administrator*) accounts. The Security Manager then creates an account that maps back to the specified NT user account. Understanding how to grant account privileges is crucial, especially when the server is configured for Integrated security mode. For example, support personnel utilizing an NT domain account are members of the Administrators group. Although these personnel require administrative access to the NT Server for system maintenance and support, it may not be desirable that these personnel gain access to SQL server databases with SQL *sa* authority. By controlling how account privileges are granted, these situations can be better managed and controlled.

Figure 3.9 indicates three NT user accounts and their corresponding SQL server logins.

NT accounts can be defined as SQL logins by using the following steps:

1. Start the SQL Security Manager and log in to the desired server. The Security Manager's User Privilege screen appears, showing NT accounts that currently have user-level logins in SQL Server.

Figure 3.9 *The SQL Security Manager integrates NT domain accounts with SQL logins.*

2. If user accounts are currently defined, the Users group button appears. Click the Users button to expand its detail.

3. From the Security menu, select the Grant New option. This will display the Grant User Privilege window. Select a local or domain user group and click the Grant button. Make sure the Add Login IDS for Group Members option is selected. This will create a SQL login for each NT user account within the selected group.

sa level accounts are added in a similar manner. After starting SQL Security Manager, select the SA Privilege option from the View menu.

TIP Double-clicking an account in SQL Security Manager displays the account details. Using the account details, user privileges can be controlled to access specific databases, as well as change the default database that users are connected to, after login to SQL Server.

SQL Setup

Within NT domains, many server administrative changes can be made centrally, and applied to all NT Servers within a domain. One example is adding user accounts. Because NT centralizes account information into a single domain security database, centralized changes can be accomplished.

With SQL Server, however, each server within an enterprise must be managed separately, even when the servers are located within a single NT domain. For this reason, it is necessary that certain configuration and management tasks be performed on each SQL server. One tool that is used for this purpose is the SQL Setup Program. The SQL Setup program is one administrative tool that really serves multiple purposes. Unlike many other setup programs, which are used solely for installation of the application, the SQL Setup program is vital for ongoing server administrative needs. The SQL Setup program is used to perform the following administrative tasks:

- Upgrade the SQL Server software
- Change the installed network support
- Add optional foreign language support to the server
- Rebuild the Master database in the event of a catastrophic software failure
- Modify various SQL Server options
- Change the security mode of the SQL Server

Explore each of these activities further in the following sections.

Upgrading SQL Server

Subsequent SQL Server software upgrades are performed using the Setup program (SETUP.EXE). Upgrades can be performed with the installed SQL Setup program, or with the SQL Setup program that comes with the new SQL installation software in the I386 directory for Intel-based machines or in the ALPHA directory for Digital Alpha based servers. Using the latter Setup program is the more

common of the two methods. Prior to starting an upgrade to SQL 6.5 from a previous version, it is essential to run the CHKUPG65.EXE utility located in the MSSQL\BINN installation directory. This ensures that no keyword conflict or other conflicts exist in the installed databases prior to the upgrade.

To perform an upgrade, simply start the SQL Setup program. The program will detect the fact that a prior installation of SQL Server has been installed and will prompt to verify that an upgrade is desired. The Setup program will then prompt to verify the Master database configuration, server options, and network support. In short, the upgrade process is similar the installation process, with the exception that Setup will upgrade the existing user database as required, depending on the version of SQL Server that is being upgraded.

For additional information on installing/upgrading SQL Server, refer to Chapter 1, "Server Installation and Upgrade."

Changing Installed Network Support

By using the Setup program, SQL Server can be configured to integrate with various network types. This determines the transport and Net Library that client workstations must use when connecting to SQL Server. By default, SQL Server will accept client connections via the Named Pipes Net Library. To accept additional Net Library connection types such as TCP/IP Sockets (see fig. 3.10), support must be enabled using the setup program.

Figure 3.10 *Additional network support is added using the SQL Setup program.*

To enable additional network support, the following steps can be used:

- 1.** Start the SQL Setup program from the icon in the Microsoft SQL 6.5 program group to launch the installed setup program located in the MSSQL\BINN installation directory.
- 2.** The Setup program will verify that SQL Server is already installed. Click the Continue button to display the setup options screen.
- 3.** Select the Change Network Support option and press the Continue button. The Select Network Protocols screen appears.
- 4.** As mentioned earlier, the Named Pipes option is enabled by default. To enable additional network support, click the check box next to the desired option and then click the OK button.
- 5.** The Setup program will now prompt for specific configuration information for each of the installed network support options. The default ListenOn Pipe for Named Pipe connections will be displayed, for example. If TCP/IP support was selected, the setup program will then prompt to verify or change the TCP/IP port number that SQL Server uses to listen for TCP/IP connections (the default is 1433).
- 6.** Click Continue until all installed options have been verified. The Setup program will now prompt to Exit to Windows NT. Note that any network support options that were changed will not be effected until SQL Server is restarted.

WARNING Modifying the default Named Pipe definition \\.\pipe\sql\query in the SQL Setup program is not usually recommended because the modification can adversely affect NT client connectivity, and may interfere with scheduled database maintenance tasks.

Adding Foreign Language Support

SQL Server supports both the U.S. English language and various localized languages such as German and French. This enables SQL Server to display system messages and present date/time formats not only in English, but also in the installed local language. Additional language support is installed using the Setup program, without the need to reinstall the SQL Server.

Foreign language support is installed by starting the Setup program and selecting Add Language from the SQL 6.5 Setup Options screen. The Select Language window appears, in which the required language can be entered along with the SQL Server sa account password (see fig. 3.11). Similar to the addition of network support options, SQL Server requires a restart before the changes are put into effect.

Figure 3.11 Use the Setup program to add support for foreign languages.

Rebuilding the Master Database

In certain failure situations, it is necessary to rebuild the Master database. SQL Server stores its own configuration information, as well as information on each user-installed database in the Master database. Because rebuilding this database essentially resets all information back to the system default, user-added information about databases, logins, permissions, and so on, is lost. Rebuilding the Master database, therefore, is somewhat less desirable than restoring the required database from a system backup.

WARNING Rebuilding the Master database will destroy all previously stored user configuration data. Ensure that all user databases and the existing Master database are properly backed up prior to the rebuild.

Certain aspects of the SQL Server configuration require that the database be rebuilt; this is true when changing the installed sort order or code page. The *sort order* is the algorithm SQL server uses to perform sorting activity of data. A number of methods are available at installation time, each carrying its own performance implications. A *code page* determines the character sets that SQL Server will support.

Normally the sort order and code page are selected at installation time and cannot be changed. In the event that it is necessary to change these options via a database rebuild, however, the follow these steps:

1. Start the SQL Setup program. Select Rebuild Master Database from the Server Options window. A verification notice appears, indicating that all previously stored data in the Master database will be lost. Click Resume to continue the rebuild process.
2. The Rebuild Options window appears. Change the sort order or code page options here as

required and click Continue.

3. The SQL Server Installation Path window now appears. Verify that the correct directory location for the existing SQL Server installation is displayed and click Continue.

4. The Rebuild Master Device window appears, which enables a different directory location to be specified for the newly rebuilt Master database. Additionally, the size for the newly rebuilt Master database device (or file) can be modified. Select the directory location and size as required and click Continue.

SQL Server now begins the rebuild process. Depending on the size specified for the Master database and the server hardware configuration, this process may take a few minutes. After the process is complete, you must restart the SQL Server. Note that when the server is restarted, all previously stored user information will not be present.

Modifying SQL Server Options

Static configuration options for SQL Server are modified by using the Setup program. Each of these options require that SQL be restarted before they are put into effect. To change static options, start the SQL Setup program and select Set Server Options from the SQL 6.5 Options window (see fig. 3.12).

Figure 3.12 *Static configuration options modified by using the SQL Setup Program.*

Table 3.5 describes the available options and typical uses for modifying each. Table 3.5

Static SQL Server Configuration Options

Option	Description/Typical Use
Root Directory	Determines the location of SQL Server System Files. This option is useful for specifying an alternative file location for troubleshooting or debugging purposes. Note that NT Registry changes might also be required to complete the location change configuration.
Master Database	Determines location of SQL Server installed Path Master database. This option is helpful in situations where an alternative Master database file may need to be temporarily refer-enced without disturbing the original SQL database configuration.
Auto Start Server	Controls whether the SQL Server service will At Boot Time automatically start when the NT Server is started or rebooted. This option is useful in situations in which SQL Server availability is required in a short timeframe after the NT Server restart.
Parameters Button	Controls SQL Server command line startup parameters. This option is most commonly used to specify troubleshooting or debugging parameters that are required when SQL Server is started.
Auto Start SQL	Similar to the Auto Start Server option-- Executive at it allows the SQL Executive service to be Boot Time started automatically when the NT Server is restarted.
Tape Support	This option controls how SQL Server handles Button database backups directly

	to tape--for example, how long SQL Server will retry attempts to back up to a tape that hasn't been loaded before it aborts the backup. This option is only a factor in environments backing up databases directly to tape.
Auto Start Mail	Controls whether SQL Mail capability is Client available automatically upon startup of SQL Server. This option might be useful when strict control over user's SQL Mail capability is required.
Mail Login	Determines the mail (or MS Exchange) account that the SQL Mail service will use to process SQL Mail messages.
Windows NT Event	Allows SQL Server system messages to be Logging displayed in the NT Event Log, in addition to the SQL Server Error Log. This option is enabled by default in SQL 6.5. Using NT Event Logging allows server application messages to be consolidated into a single logging mechanism, easing SQL Server administration.
SQL PerfMon	This option makes SQL Server performance Integration data available to the Windows NT Performance Monitor. When enabled, the Integration mode must also be determined. Direct Response mode provides better response time in the NT Performance Monitor; however, the actual data displayed is actually one refresh interval behind. On Demand mode provides a slower response time, although the information displayed is the most current SQL performance data.
XP_CMDSHELL	XP_CMDSHELL is an extended stored Impersonates procedure that enables SQL Server to directly Client interact with the NT operating system. Because of the powerful nature of this procedure, enabling this option restricts XP_CMDSHELL use to only system administrators (SQL sa account), or members of the local NT Administrators group on Servers supporting Integrated security.
Enable SNMP	This option is available only on NT Servers Support with SNMP (System Network Management Protocol) support installed. Selecting this option enables SQL system messages to be available as SNMP alerts. This is useful in environments in which SNMP management tools are in use.

SQL Trace

The SQL Trace Utility, new in SQL 6.5, provides the primary means by which specific user connections and server processes can be monitored. This utility enables the administrator to obtain detailed information on the actual SQL database queries that a user executes from a given application. SQL Trace becomes an important tool in managing server performance as well as in troubleshooting. For example, suppose that users complain of poor SQL performance every day at around the same time. SQL Trace could help determine what condition or user request is contributing to the poor performance.

By using SQL Trace, filters that monitor connection information for a given user, or for all users on the SQL Server, can be defined. Additionally, a filter can be built to monitor connections on a single server or multiple SQL Servers on the network. SQL Trace filters can redirect output in one of several ways (see table 3.6). Table 3.6

SQL Trace Output Options

Output Type	Description/Use
Screen File	The Screen File option redirects trace output to the filter display screen in SQL Trace. This method is best suited for online monitoring and ad-hoc query debugging (see fig. 3.13).
Script File	This method is helpful because trace commands are formatted for explicit Transact-SQL execution. Script output also allows performance data to be incorporated (see fig. 3.14).
Log File	Log File output provides good audit trail information of subsequent connection activity (see fig. 3.15).

Figure 3.13 *SQL Trace Screen File Output.*

Figure 3.14 *SQL Trace Script File Output.* **Figure 3.15** *SQL Trace Log File Output.*

TIP SQL Trace functionality can also be initiated from the command line by using the XP_SQLTRACE extended stored procedure. In fact, the SQL Trace application simply provides a graphical interface to use much of this procedure's existing functionality.

SQL Trace also provides the option of including performance information when filtering results of a given trace. This option is helpful for identifying potential performance problems caused by poor query design or database structure. SQL Trace performance data incorporates the following statistics:

- The duration of the SQL event
- CPU utilization for the SQL event
- Number of disk reads
- Number of synchronous writes for the SQL event

SQL Performance Monitor

While the SQL Trace tool provides information about SQL connections, the SQL Performance Monitor provides information about all metrics of SQL Server's operation (see fig. 3.16). When SQL Server is installed, additional performance counters are configured in the NT Registry that are specific to SQL Server Operation. This makes SQL Server performance data available to the Windows NT Performance Monitor.

Figure 3.16 *The SQL Performance Monitor.*

SQL Performance Monitor utilizes the capability of the Windows NT Performance Monitor (PERFMON.EXE) to save custom chart settings in a *.PMC file. This chart settings file (SQLCTRS.PMC) is located in the MSSQL\BINN installation directory, and contains the most common SQL Server Performance Monitor Counters.

SQL Performance Monitor provides a centralized means to monitor multiple SQL servers in the

network. Multiple servers can be monitored by using the following steps:

1. Start SQL Performance Monitor using the Performance Monitor icon in the SQL 6.5 Program Group. This action will launch the Performance Monitor with the common counters for the local SQL Server displayed.
2. Click the + tool button to add additional performance counters. The Add to Chart window appears.
3. In the Computer field, enter a remote SQL Server name or click the Browse button (located next to the field) to select the required server.
4. Click the Object field to select the SQL Server object on the given server. After the SQL Server object is selected, the available counters are displayed in the Counters field.
5. Choose the counters that are to be monitored and click the Add button. After all the required counters have been added, click the Done button to return to the Performance Monitor display. This makes it possible to view performance data for each server in a single chart display.

TIP Keep in mind that SQL Performance Monitor is nothing more than presaved settings for the Windows NT Performance Monitor. This means that the original SQLCTRS.PMC file can be modified to suit your given environment by using the File, Save menu option.

ISQL/W

The ISQL/W Tool is the interface to the SQL Server query engine. Its primary use is to execute explicit Transact-SQL commands on the SQL Server. Although it is not used extensively for managing servers within the enterprise, many system-related SQL commands can be invoked using ISQL/W. This provides direct access to many of the same procedures that are used by other SQL tools for server management activity. Table 3.7 illustrates some common ISQL/W alternatives for server administration. Table 3.7

Common ISQL/W Alternatives for Server Management

Command/ Procedure	Description
SHUTDOWN	This command can be invoked to shut down a SQL Server. After the server has been stopped, however, the connection to ISQL/W is broken and the server will have to be restarted using Enterprise Manager or the SQL Service Manager.
XP_SQLTRACE	This is the extended stored procedure that provides identical functionality to the SQL Trace tool. It provides a command-line method to trace user connection output to a log or *.SQL file.
sp_configure	This is the stored procedure used to configure SQL Server operation parameters. It provides the same functionality as the Configure Servers menu option in SQL Enterprise Manager.
sp_who	This command displays a list of user connection activity. Similar in function to

	the Current Activity display in Enterprise Manager.
xp_startmail	This command starts the SQL Mail Service.
xp_Stopmail	This command stops the SQL Mail Service.

Exercises

Exercise 3.1: Registering Servers By Using SQL Enterprise Manager

In this exercise, you will register a SQL Server so that it can be administered using the Enterprise Manager. This allows all aspects of the server's configuration to be remotely modified and controlled. This exercise can be used to register a SQL server by using Standard SQL security. This is a rather simple exercise that should take no more than 10 minutes.

1. Start the SQL Enterprise Manager from the Microsoft SQL Server program group.
2. If no servers were previously registered, the Register Server window is displayed. Enter the server name in the Server field or click the Servers button to display a list of active SQL servers.
3. In the Login Information section, highlight the option for Standard Security. This will require a valid SQL login ID and password.
4. Enter **sa** for the account name and the required password.
5. Click the Register button to complete the server registration.
6. Click the Close button to return to the Server Display window. The newly added server should now be visible in the default SQL 6.5 Server group.

For more information, consult the "Registering Servers" section, earlier in this chapter.

Exercise 3.2: Creating Server Groups in Enterprise Manager

This exercise demonstrates how to create server groups in the SQL Enterprise Manager. Server groups are used to logically arrange multiple servers within the enterprise to ease administration. This exercise should take approximately five minutes.

1. Start the SQL Enterprise Manager. If no servers have been registered, register at least a single server now. (Refer to Exercise 3.1.)
2. In the Server Display window, highlight the default SQL 6.5 global group.
3. Click the right mouse button and select the New Server Group option. This will display the Manage Server Groups window.
4. Enter a group name and select the Top Level option. This will create a top-level group as a peer to the default SQL 6.5 group.

5. Click the Add button and then click Close to complete the group definition and return to the Server Display window.

Subgroups can be created in the same manner by selecting the Subgroup option from the Manage Server Groups window.

For more information, refer to the earlier "Defining Server Groups" section.

Exercise 3.3: Starting and Stopping Servers

In this exercise you'll start and stop servers by using the SQL Service Manager. This represents just one method that can be used to start and stop servers. The required time to complete this exercise is approximately five minutes.

1. Start the SQL Service Manager from the Microsoft SQL Server Program group.
2. Ensure that the desired SQL Server name appears in the Server field. If not, highlight the server and enter the required server name.
3. From the Services pull-down menu, select the MSSQL Server service.
4. Double-click the red light on the stoplight icon to stop the SQL service. The status line at the bottom of the window will indicate when the service has been completely stopped.
5. Double-click the green light on the stoplight icon to start the SQL Server service. The status line will indicate that the service is running after startup is complete.

The SQL Executive service and SQL DTC services can be controlled in the same manner simply by selecting the appropriate service from the pull-down menu in the SQL Service Manager.

For more information on stopping and starting servers, see the sections titled "SQL Service Manager" and "Starting and Stopping Servers" earlier in chapter.

Exercise 3.4: Modifying the SQL Server Configuration

In this exercise, you'll modify the server configuration to increase the number of simultaneous user connections that the server will accept. This is a common administrative function that can be completed by using the SQL Enterprise Manager. This exercise will take approximately 10 minutes to complete.

1. Start the SQL Enterprise Manager. Click the + sign next to the Server icon to expand its object list and ensure connectivity to the correct server for configuration.
2. From the Server menu, select SQL Server and Configure. This will display the SQL Configuration/Options window.
3. Click the Configuration tab and scroll down the list of options until the User Connections option is displayed.

4. In the Current column, increase the number of simultaneous user connections by ten (10). The current configuration value can be obtained by viewing the value in the Running column. For example, if the running value is 10, enter 20 in the Current column.

4. Click the Apply Now button to save the configuration option change. Click the OK button to return to the Server Display window.

5. In order to put the configuration change into effect, it is necessary to restart the SQL Server. Can you remember one of the methods that can be used to accomplish this?

Refer to the earlier section "Modifying the SQL Server Configuration" for additional information.

Exercise 3.5: Using the SQL Server Setup Program

The SQL Server Setup program is used to modify various aspects of the SQL Server installation. In this exercise, you'll use the setup program to disable Windows NT event logging and configure SQL Server to start automatically when the NT Server is started. This exercise should take approximately five minutes to complete.

1. **1.** From the Microsoft SQL Server program group, start the SQL Setup program.
2. **2.** Click the Continue button twice to display the SQL Server Options window.
3. **3.** Select Set Server Options and click the Continue button.
4. **4.** Select the Auto Start Server at Boot Time.
5. **5.** Deselect Windows NT Event Logging.
6. **6.** Click the Change Options button to save the configuration changes to the NT Registry and put them into effect for SQL Server.
7. **7.** Click Exit to quit the setup program and return to Windows NT.
8. **8.** Be sure to restart SQL Server to put the configuration changes into effect.

For more information on using the SQL Setup program, see the section titled "SQL Setup."

Exercise 3.6: Using SQL Trace

The SQL Trace tool is useful for monitoring specific information on a given user session or all sessions with the SQL Server. In this exercise, you set up an onscreen filter to capture all information for the sa user account when accessing SQL Server with the ISQL/W query tool. Allow approximately 10 to 15 minutes to complete this exercise.

- 1.** From the SQL Server Program group, start the SQL Trace application.
- 2.** When prompted for login, log in to the server with the sa user account and password.
- 3.** From the File menu, select the New Filter option.

4. Enter a name for the new filter, such as SA Active Trace.
5. Press the Browse button next to the Login Name field to display available SQL Server logins.
6. Highlight the sa account and press OK.
7. Click the Browse button next to the Application field and select Microsoft ISQL/W from the list of applications. Click the OK button.
8. Make sure <All> is displayed in the Host Name field to ensure that sa user activity from ISQL/W on any client workstation is filtered.
9. In the SQL Scripts section, ensure that the View on Screen option is selected.
10. Click OK to complete the filter configuration. You will be prompted to start the filter if it is not currently running.
11. Click Yes to start the filter. The SQL Trace Filter output screen will appear, verifying that the filter has been successfully defined and is running.

Refer to the section titled "SQL Trace" for additional information.

Review Questions

The following questions will test your knowledge of the information in this chapter.

1. Name three tools that can be used to stop SQL Server.
 - A. SQL Enterprise Manager, SQL Setup Program, SQL Service Manager
 - B. SQL Enterprise Manager, SQL Service Manager, Windows NT Control Panel
 - C. SQL Service Manager, SQL Performance Monitor, SQL Trace
 - D. SQL Enterprise Manager, ISQL/W, SQL Performance Monitor
2. The foundation for SQL Server administration is through the use of SQL-DMF. What does DMF stand for?
 - A. Distributed Monitoring Facility
 - B. Distributed Monitoring Foundation
 - C. Dynamic Management Facility
 - D. Distributed Management Framework
3. Which single utility is used to stop SQL Server and all of its related services?
 - A. SQL Service Manager

- B.** SQL Setup Program
 - C.** SQL Security Manager
 - D.** None of the above

- 4.** You want to configure the SQL Server so that it must be started manually after the NT Server is started, rather than starting automatically upon boot. Which administration tool is used to perform this change?
 - A.** SQL Setup Program
 - B.** SQL Enterprise Manager
 - C.** SQL Service Manager
 - D.** ISQL/W

- 5.** Name the ISQL command that can be used to stop the SQL Mail service using the ISQL/W tool.
 - A.** SHUTDOWN SQL_MAIL
 - B.** XP_STOPMAIL
 - C.** EXEC SQL_MAIL(STOP)
 - D.** NET STOP SQLMAIL

- 6.** What two (2) methods can be used to configure SQL Server to stop accepting new user connections, while still maintaining existing connections?
 - A.** Using SQL Service Manager, click the yellow light in the stoplight icon.
 - B.** Start the SQL Setup program. Select the option Disallow Future Logins from the Server Options window.
 - C.** Using the SQL Security Manager, select Paused Connections from the View menu.
 - D.** Right-click the SQL Server icon in the SQL Enterprise Manager Server Display window. Select Pause from the pop-up menu.

- 7.** What is the default Net Library for SQL Server?
 - A.** Named Pipes
 - B.** RPC
 - C.** IPX/SPX
 - D.** TCP/IP

- 8.** Which Security mode enables SQL Server to integrate SQL user logins with NT domain user accounts?
 - A.** Standard
 - B.** Mixed
 - C.** Integrated
 - D.** B and C

9. A server's configuration can be modified by using the SQL Enterprise Manager, and selecting the SQL Server, Configure option from the Server menu. What Transact-SQL command can be used in ISQL/W to accomplish the same function?

- A.** EXEC sys.configs
- B.** SELECT options FROM sysoptions
- C.** XP_CONFIG
- D.** sp_configure

10. You are implementing SQL Server in a Netware 4.1 environment. Which of the following configurations must be made in order for Novell client workstations to access SQL Server? (Choose one answer.)

- A.** Enable GSNW (Gateway Services for Netware) on the NT/SQL Server. Enable the IPX/SPX Net Lib in the SQL Server Setup program.
- B.** No special configuration is required.
- C.** Install the IPX/NWLink protocol on the NT Server.
- D.** Install the IPX/NWLink protocol on the NT Server. Enable the IPX/SPX Net Lib in the SQL Setup program.

11. Which of the following is true of the SQL Performance Monitor?

- A.** It is really a Chart Settings file of common SQL counters.
- B.** It cannot be used if SQL Enterprise Manager is stopped.
- C.** It is a special version of Performance Monitor that can only be used with SQL Server.
- D.** It cannot monitor multiple SQL Servers concurrently.

12. How can NT Event logging for SQL Server messages be disabled?

- A.** This feature cannot be disabled.
- B.** Choose the Error Log option from the Server menu in SQL Enterprise Manager and select the Disable option.
- C.** Start the SQL Setup program and deselect the NT Event Logging option from the Server Options window.
- D.** Using the SQL Setup program, reinstall SQL Server and select the NT Event Logging option when the Server Options window is displayed.

13. Which two (2) SQL Server Security modes enable SQL user logins to be integrated with

NT domain security?

- A. Integrated mode
- B. Standard (default) mode
- C. RPC Security mode
- D. Mixed Security mode

14. Which of the following tools is used to collect performance information about SQL user connections?

- A. SQL Performance Monitor
- B. SQL Enterprise Manager
- C. SQL Trace
- D. SQL Connection Manager

15. The SQL Setup program is used to modify which of the following?

- A. Static SQL configuration parameters
- B. SQL Security mode
- C. Server registration status
- D. A and B

16. SQL Mail is controlled by using which of the following tools?

- A. The SQL Enterprise Manager
- B. SQL Trace
- C. ISQL/W
- D. SQL Service Manager

17. Before a SQL Server can be administered by using Enterprise Manager, what must first be done to the server?

- A. SQL Server must be added to the SQL Remote Server list in the Enterprise Manager Server Display window.
- B. SQL Server must be registered.
- C. No special actions are required prior to using Enterprise Manager.
- D. The SA must first attach to a network driver on the server before SQL Server can be administered.

18. Which Net Library does not support integrated NT/SQL Security?

- A. RPC
- B. Named Pipes
- C. IPX/SPX
- D. TCP/IP

19. Which two (2) of the following files are components of the SQL Distributed Management Framework?

- A. SQLDMF.DLL
- B. SQLOLE65.SQL
- C. SQLOLE.REG
- D. SQLOLE65.TLB

20. The SQL Security Manager can be used to perform which of the following tasks?

- A. Grant SQL sa authority to NT Domain Administrator accounts.
- B. Prevent unauthorized users from accessing a given database.
- C. Add NT domain accounts using SQL Server.
- D. None of the above.

21. What is the name of the extended stored procedure that can be executed via ISQL/W to provide similar functionality to the SQL Trace tool?

- A. XP_SQLTRACE
- B. DBCC(SQLTRACE)
- C. sp_configure trace, 'on'
- D. sp_SQL_TRACE

22. Which tool is used to enable SNMP alerting for SQL Server?

- A. SQL Client Configuration Utility
- B. SQL Enterprise Manager
- C. SQL Trace
- D. SQL Setup Program

23. Which of the following is true using the L Setup program?

- A. The server must be completely reinstalled in order to install foreign language support.
- B. Foreign language support can be installed using the SQL Setup program without reinstalling the server.
- C. Foreign language support cannot be managed using the SQL Setup program.
- D. Foreign language support is automatically installed when running the SQL Setup program.

24. What are Server groups in SQL Enterprise Manager?

- A. A logical display of servers presented in groupings to aid in administration.

B. SQL Servers within a domain that are configured to use the same security mode.

C. NT domain user accounts that have the required privileges to access SQL Server.

D. SQL Server user logins that are classified by their respective privilege levels.

25. Which of the following statements is true regarding SQL memory allocation?

A. SQL Server allocates its memory "working set" from Windows NT when SQL Server is started.

B. SQL Server memory allocation can only be modified by reinstalling the server.

C. SQL Server allocates its memory "working set" dynamically, based on load.

D. None of the above.

Review Answers

1. B

2. D

3. D

4. A

5. B

6. A, D

7. A

8. D

9. D

10. D

11. A

12. C

13. A, D

14. C

15. D

16. A

17. B

18. D

19. B, C, D

20. A

21. A

22. D

23. B

24. A

25. A

Answers to Test Yourself Questions at Beginning of the Chapter

1. SQL Server can be stopped using the SQL Service Manager tool. Select the SQL Server service and double-click the red light on the stoplight icon. Another method is to use the Windows NT Control Panel to access the MSSQLServer service. Click the Stop button to stop the server. SQL Server can also be stopped by right-clicking the SQL Server in the Server display window of the SQL Enterprise Manager and then selecting Stop from the Service pop-up window. Refer to the section titled "Starting and Stopping Servers."

2. To increase SQL memory allocation, use the Configure option from the Server, SQL Server menu in the Enterprise Manager. Access the Configuration tab from the Server Option window and modify the Memory setting. Configuration parameters are also modified using the sp_configure Transact-SQL command in ISQL/W. Refer to the section titled "Modifying the SQL Server Configuration" for details on modifying the server's configuration.

3. The SQL Server Security mode is modified using the SQL Security Manager utility. For more information on this utility, refer to the "SQL Security Manager" section.

4. Windows NT Security is integrated with SQL Server by using the SQL Setup program to set the server to use either Mixed or Integrated Security mode. Windows NT domain accounts can be customized with various SQL Server account privileges by using the

SQL Security Manager application. This is discussed in greater detail in the "SQL Security Manager" section.
