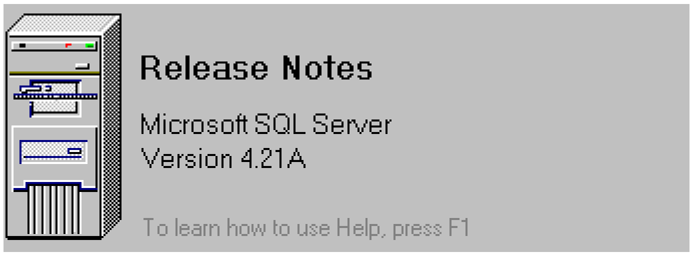

Release Notes

Microsoft SQL Server
Version 4.21A

To learn how to use Help, press F1

About These Release Notes

These Release Notes contain information pertaining to the installation and use of Microsoft(R) SQL Server(TM). This information supplements the manuals included in the SQL Server package.

Using SQL Server 4.21A with Windows NT Server Version 3.5

What's Changed in Microsoft SQL Server Version 4.21A

With this release, some SQL Server components have been updated for better compatibility with the Windows NT(TM) Server version 3.5 operating system.

Beginning with this release, Microsoft SQL Server version 4.21A is available in two package configurations: Microsoft SQL Server and Microsoft SQL Server Workstation System. Microsoft SQL Server and SQL Server users are now licensed separately from one another. You must acquire separate user licenses for each user who wants to access SQL Server; an administrative-use license to install and configure SQL Server is included. For Microsoft SQL Server Workstation System, no separate user license is needed.

Also provided with this release are some additional disks (this additional software is not provided on the Microsoft SQL Server CD-ROM):

- Supplemental Services. For Intel(R)-, MIPS(R)-, and Alpha AXP(TM)-based computers, this disk provides Net-Libraries for DECnet(TM) sockets and for AppleTalk(R) (ADSP). The DECnet sockets Net-Libraries allow VMS(TM) clients using DECnet to access SQL Server, and the AppleTalk Net-Libraries allow Apple(R) Macintosh(R)-based clients to connect to SQL Server via AppleTalk. For more information, see the README.TXT file included on the disk.
- ODBC 2.0. This disk provides ODBC client computer support only for Intel-based computers. It includes the Open Database Connectivity (ODBC) Driver Manager, SQL Server ODBC client drivers for Win32(R) and Win16, and online Help.

  To install ODBC drivers on Microsoft Windows(TM) version 3.1 clients, use either File Manager or Program Manager to run Setup from the root directory of the ODBC 2.0 disk. This installs the Win16 driver, plus a special 32-bit version of the ODBC Driver Manager that allows Win32S applications to access the 16-bit driver. To install ODBC drivers on Windows NT version 3.1 or 3.5 clients, run Setup from the \32i directory of the ODBC 2.0 disk. If you will be running 16-bit applications in the Windows on Win32 (WOW) subsystem of Windows NT, run both versions of Setup.

  For more information, see the ODBC online Help.

Microsoft SQL Server and Microsoft SQL Server Workstation System

Microsoft SQL Server version 4.21A is available in two package configurations:

- Microsoft SQL Server is a relational database management system that is capable (depending on your application and hardware environment) of supporting hundreds of simultaneous users.
- Microsoft SQL Server Workstation System has the same features and functionality as Microsoft SQL Server, except that it is a single-user licensed product that supports a maximum of 15 simultaneous database connections.

Install SQL Server on Windows NT Server Version 3.5

For Windows NT version 3.5, Microsoft SQL Server should be installed on a computer running the Microsoft Windows NT Server operating system. SQL Server is not supported for installation on a computer running Windows NT Workstation version 3.5. Windows NT Server is optimized for server functions and offers better performance, scalability, and capacity than Windows NT Workstation.

Microsoft SQL Server is not recommended for installation on a Primary Domain Controller or a Backup Domain Controller because those computers perform the resource-intensive tasks of maintaining and replicating the network accounts database and performing network login authentications.

For Windows NT version 3.5, Microsoft SQL Server Workstation System can be installed on a computer running either the Microsoft Windows NT Workstation or Microsoft Windows NT Server operating system.

Note that SQL Server and SQL Server Workstation System are both compatible with Windows NT version 3.1 and Windows NT Advanced Server version 3.1.

Setting Frame Types for NWLink IPX/SPX

In order to establish and maintain connections, SQL Servers and clients communicating via the NWLink IPX/SPX protocol must be configured to use the same frame type. The most commonly used frame types for Ethernet are 802.3 and 802.2. Windows NT version 3.5 offers an auto-detect option that in most cases properly configures a server or workstation for the appropriate frame type.

In some rare situations, especially on networks with mixed Novell(R) server versions or with clients running various frame types, you might encounter difficulty establishing connections between SQL Server clients and servers when the Windows NT version 3.5-based computers are configured for frame type auto-detect. If this occurs, you can configure the Windows NT version 3.5-based computers for specific frame types.

Note: Windows NT Servers can be configured for multiple frame types. Windows NT Workstations can be configured only for a single frame type.

First, determine what frame type(s) you will use. Check what frame type other network clients are using (you may want to set your Windows NT version 3.5-based computers to that frame type). For Ethernet, you will usually use frame type 802.2 for networks with servers running NetWare (R) version 3.12 or later, and frame type 802.3 for networks with other configurations.

To set frame types on each computer running Windows NT Server version 3.5:

1. From the Control Panel, choose the Network application. The Network Settings dialog box appears.
2. From the Installed Software list, select NWLink IPX/SPX Compatible Transport, and then choose Configure. The NWLink IPX/SPX Protocol Configuration dialog box appears.
3. Select the Manual Frame Type Detection option, select one or more frame types from the list, and then choose the Add button.
4. Choose OK. The Network Settings dialog box returns.
5. Choose OK. The Control Panel returns. Close the Control Panel.

To set a frame type on each computer running Windows NT Workstation version 3.5:

1. From the Control Panel, choose the Network application. The Network Settings dialog box appears.
2. From the Installed Software list, select NWLink IPX/SPX Compatible Transport, and then choose Configure. The NWLink IPX/SPX Protocol Configuration dialog box appears.
3. In the Frame Type box, select the appropriate frame type (you may choose only one), and then choose OK. The Network Settings dialog box returns.
4. Choose OK. The Control Panel returns. Close the Control Panel.

For additional information on frame types, choose the Help button from the NWLink IPX/SPX Protocol Configuration dialog box.

SQL Server and DHCP Servers

Windows NT version 3.5 provides easy administration of large TCP/IP networks by offering the Dynamic Host Configuration Protocol (DHCP) service for automatic TCP/IP configuration. Windows Internet Name Service (WINS) provides dynamic mapping of network names and addresses, enabling users to operate in large scale TCP/IP networking environments with little administrative support.

When using the DHCP service, computers are automatically assigned a "leased" (temporary) IP address by the DHCP server. Leases have an expiration date and time, but if the computer is available on the network when the lease expires, the leased IP address is automatically renewed. However, if a computer is not available on the network when the lease expires (for example, if it is turned off), the DHCP server can cancel the lease and that address becomes available to any computer needing a lease. When the original computer again becomes available on the network it is assigned an IP address that may be different from its previous IP address.

For SQL Server, this means that on rare occasions the IP address of a server might change (usually, only if the server was turned off at the time the IP address lease expired). If you occasionally encounter problems with dropped client-server connections when using the DHCP service on Windows NT version 3.5, you can assign reserved IP addresses to computers running SQL Server. (This is necessary only for servers, not clients.)

Instructions for setting a reserved IP address can be found in Chapter 4, "Installing and Configuring DHCP Servers," in *Microsoft Windows NT Server TCP/IP*, available with Windows NT Server.

Instructions for setting a reserved IP address are also available from the DHCP Manager online Help file. You can start DHCP Manager on the DHCP Server by choosing its icon from the Network Administration Tools group in Program Manager. View the online instructions by selecting the Help menu, choosing Contents, choosing "Administering DHCP Clients," and then choosing "Managing Client Reservations."

Using Dump Devices on a NetWare Server

The Client Service for NetWare allows a Windows NT Server version 3.5 to access files on a NetWare file server. If you want to use a file on a NetWare server as a disk dump device for SQL Server, you must:

1. Run SQL Server as a service under a Windows NT user account that matches the user ID and password of a valid NetWare user. You cannot run SQL Server as a service under the LocalSystem account.

   For information on how to set this up, see "Configuring Network Permissions for SQL Server," in Chapter 2 of the *Microsoft SQL Server Configuration Guide* (page 64).

2. Identify the network share point of the dump device using UNC naming rather than drive letters. For example, the following command would be valid:

   ```
   sp_addumpdevice 'disk','netware_dump','\\netware1\big_disk\dumpdir\dumpfile.dat', 2
   ```

   Note: In the above command, you cannot use a drive letter ("n:") in place of the UNC designation for the drive and dump share ("\\netware1\big_disk").

Running SQLMail on Windows NT Server Version 3.5

With Windows NT Server version 3.5, SQL Server can run as a service in a user account and use SQLMail functionality. If you run SQL Server in the LocalSystem account on Windows NT Server version 3.5, then in the Services application in the Windows NT Control Panel, in the Startup dialog box, you should select the option Allow Service To Interact With Desktop.

Installing or Upgrading

To install Microsoft SQL Server version 4.21A, run the Setup program and choose the Install SQL Server And Utilities option. To upgrade a previous version of Microsoft SQL Server to version 4.21A, run the Setup program and choose the Upgrade SQL Server option.

The CD-ROM disc contains three separate SETUP directories, one for Intel-based computers (the \I386 directory), one for MIPS(R)-based computers (the \MIPS directory), and one for Alpha AXP(TM)based computers (the \ALPHA directory). SQL Server must be installed from the directory containing the software compatible with your hardware platform. You cannot use a different processor architectures Setup program to install SQL Server.

For detailed installation instructions, use the Setup program's online Help or see the *Microsoft SQL Server Configuration Guide*. Note that the printed SQL Server documentation is the same for versions 4.2, 4.21, and 4.21A.

For more information about changes to Microsoft SQL Server, see the rest of the topics in this Windows Help file. You can print individual topics within this Help. If you want to print a copy of all the information at once, print the README.TXT file, found in your \SQL\INSTALL directory.

If You Have a Beta Version of SQL Server for the Alpha AXP Platform

If you have a Beta version of Microsoft SQL Server for an Alpha AXP(TM)-based computer, you must newly install this release.

This release includes a CD-ROM disc for installing SQL Server. Note that the CD-ROM disc contains three separate SETUP directories, one for MIPS(R)-based computers, one for Intel(R)-based computers, and one for Alpha AXP-based computers. Be sure you set up SQL Server using the directory for Alpha AXP-based computers (the \ALPHA directory). You cannot use a different processor architecture's Setup program to install SQL Server.

For more information about changes to Microsoft SQL Server, see the rest of the topics in this Windows Help file. You can print individual topics within this Help. If you want to print a copy of all the information at once, print the README.TXT file, found in your \SQL\INSTALL directory.

Migrating Data Between RISC and Intel Architectures

It is not possible to dump a database in one processor architecture and load it in another. Even if the operation appears to succeed, serious problems will subsequently result. You also should not run the Setup program to perform an upgrade from a database environment that existed on a different computer. For example, you cannot dump a database from an Alpha AXP-based computer and load it into an Intel-based computer. Nor should you copy your MASTER.DAT from an Intel-based computer to a MIPS- or Alpha AXP-based computer and run Setup to upgrade it. (However, you can use SQL Transfer Manager, provided with Microsoft SQL Server version 4.21, to transfer the data from one architecture to another.)

New SQLMail Functionality

- About SQLMail
- xp_startmail Extended Stored Procedure
- xp_stopmail Extended Stored Procedure
- xp_sendmail Extended Stored Procedure

About SQLMail

With version 4.21, SQL Server includes extended stored procedures that allow SQL Server to send messages through the built-in mail application programming interface (MAPI) client interface in Windows NT. These messages can consist of short text strings, the output from a query, or an attached file. You can send messages from within a trigger or a stored procedure. (For example, you can send an alert when changes occur in the database.) Mail functionality has also been integrated with SQL Monitor and SQL Administrator scheduled backups. You can use this capability with SQL Administrator's scheduled backup feature to enable SQL Monitor to notify a list of recipients when backups occur and if errors occur during a backup. In addition, you can use the SQLMail feature to notify you when long-running processes, such as data downloads, are complete.

A new DLL, SQLMAPI.DLL, is installed with SQL Server and includes the extended stored procedures necessary for mail enabling. You enable the SQLMail functionality through the Set Server Options dialog box in the Setup program. You can start and stop mail or send a message by using the new extended stored procedures (see the topics listed below). You can also automatically start mail when you start SQL Server by setting AutoStart Mail Client in the Set Server Options dialog box. For more information about enabling SQLMail, see the online Help included with the Setup program. Note that before you enable SQLMail you should ensure that the Mail client interface in Windows NT is set up and working and that you can send an ordinary mail message to the intended recipients of SQLMail messages. For more information about the Mail client interface, see your documentation for Microsoft Windows NT.

See Also

xp_startmail Extended Stored Procedure
xp_stopmail Extended Stored Procedure
xp_sendmail Extended Stored Procedure

xp_startmail Extended Stored Procedure

The **xp_startmail** extended stored procedure starts a SQL Server Microsoft Mail client session.

Syntax

xp_startmail ['@user'] [, '@password']

Parameters

@user
Is an optional parameter that specifies the mail user name.

@password
Is an optional parameter that specifies the mail password.

Remarks

SQL Server attempts to log in to Microsoft Mail using the name and password specified. If no name or password is supplied, SQL Server uses the name and password specified in the Mail Login dialog box under Set Server Options in Setup. If no name and password are specified in the Mail Login dialog box, or if they are incorrect, you will receive error 17903 "MAPI login failure" in the Windows NT Event Log and/or the SQL Server error log.

Even if you don't use the Mail Login dialog box to save your Mail user name and password, you must select the Copy SQLMail Configuration From Current User Account option in the Mail Login dialog box to set up SQLMail for the first time.

If Mail is already running on the workstation, SQL Server "piggybacks" on that instance of Mail instead of starting one of its own.

Example

```
xp_startmail 'sqluser', 'sqlpassword'
```

Messages

Msg 17952 "Failed to start Microsoft Mail session. Check the errorlog file in the SQL Server directory for details."
The name and/or password you have typed (either in the Mail Login dialog box, or as parameters to xp_startmail) could be incorrect. Check that you are using the correct name and/or password and that you have typed them correctly. Make sure that you can run a regular Microsoft Mail session using that user name and password. Also make sure that you have selected the option Copy SQLMail Configuration From Current User Account in the Mail Login dialog box after you have your Microsoft Mail client session working.

"Microsoft Mail session is already started."
A Microsoft Mail session is already started; SQL Server will not start one of its own.

Permissions

Execute permission defaults to the system administrator, who can grant permission to others.

See Also

xp_stopmail Extended Stored Procedure

xp_sendmail Extended Stored Procedure

xp_stopmail Extended Stored Procedure

The **xp_stopmail** extended stored procedure stops a SQL Server Microsoft Mail client session.

Syntax

xp_stopmail

Messages

Msg 17966 "Microsoft Mail session is not started."
There is no existing SQL Server mail session to stop.

Permissions

Execute permission defaults to the system administrator, who can grant permission to others.

See Also

xp_startmail Extended Stored Procedure
xp_sendmail Extended Stored Procedure

xp_sendmail Extended Stored Procedure

The **xp_sendmail** extended stored procedure sends a message, and/or a query result set, and/or an attachment to the specified recipients.

Syntax

xp_sendmail @recipients, [@message] [, @query] [, @attachments] [, @copy_recipients] [, @blind_copy_recipients] [, @subject] [, @type] [, @attach_results] [, @no_output] [, @no_header] [, @width]

Parameters

@recipients

Is a required parameter specifying the names of the people you are sending the mail to. If you specify more than one name, separate the names by semicolons.

@message

Is an optional parameter that specifies the message to be sent. You must specify @message, @query, or @attachments.

@query

Is an optional parameter that specifies a valid SQL Server query, the result of which will be sent in mail. You must specify @query, @message, or @attachments.

@attachments

Is an optional parameter that specifies a file to attach to the mail. You must specify @attachments, @query, or @message.

@copy_recipients

Is an optional parameter that identifies other recipients you are sending the mail to.

@blind_copy_recipients

Is an optional parameter that identifies other recipients you are sending a blind copy of the mail to.

@subject

Is an optional parameter that specifies the subject of the mail. If you do not specify a subject, "SQL Server Message" is used as the subject.

@type

Is an optional parameter that sets a custom message type of the mail message. Custom message types are of the form

IP<MIC>.VendorName.subclass

A message type beginning with IPM (interpersonal message) will show up in the recipients' Inbox; a message type beginning with IPC will not appear in the Inbox and must be read by a custom MAPI application. The default is Microsoft Mail's IPM type. See the *Windows NT Resource Guide* or the *Microsoft Mail Technical Reference* (available separately) for more information about using custom message types.

@attach_results

Is an optional parameter that specifies that the results set of a query should be sent in mail as an attached text file (.TXT) instead of appended to the mail. The default association for a .TXT file is Notepad, but a different association can be specified using File Manager. The default for this parameter is False, which means that the results set is appended to the message.

@no_output

Is an optional parameter that sends the mail but does not return any output to the client session of SQL Server that sent the mail. The default value is False, which means that the client session of SQL Server receives output.

@no_header

Is an optional parameter that sends the query results in mail but does not send column header information with the query results. The default is False, which means that column header information is sent with the query results.

@width

Is an optional parameter that sets the line width of the output text for an @query message. This parameter is identical to the **/w** parameter in ISQL. For queries that produce long output rows, use @width together with @attach_results to send the output without line breaks in the middle of output lines. The default width is 80 characters.

Remarks

The SQLMail session must be started prior to executing **xp_sendmail**. Sessions can be started either automatically (using the Auto Start Mail Client option in the Set Server Options dialog box of Setup) or with **xp_startmail**. One SQLMail session supports all users on the SQL Server, but only one user at a time can send a message. Other users will automatically wait their turn until the first user's message is sent.

If @query is specified, **xp_sendmail** logs in to SQL Server as a client and executes the specified query. SQLMail makes a separate connection to SQL Server; it does not share the same connection as the original client connection that issued **xp_sendmail**. Note that the @query can be blocked by a lock held by the client connection that issued **xp_sendmail**. For example, if you are updating a table within a transaction and you create a trigger for that update that attempts to select the updated row information as the @query parameter, the SQLMail connection will be blocked by the exclusive lock held on that row by the initial client connection.

Examples

A.

```
xp_sendmail 'user1', 'The master database is full.'
```

This example sends a message to user1 that the master database is full.

B.

```
xp_sendmail @recipients = 'user1;user2', @message = 'The master database is full.',
@copy_recipients = 'user3;user4', @subject = 'Master Database Status'
```

This example sends the message to user1 and user2, with copies sent to user3 and user4. It also specifies a subject line for the message.

C.

```
xp_sendmail 'user1', @query = 'sp_configure'
```

This example sends the results of the stored procedure **sp_configure** to user1.

D.

```
xp_sendmail @recipients = 'user1', @query = 'select * from sysobjects',
@subject = 'SQL Server Report', @message = 'The contents of sysobjects:',
@attach_results = 'True', @width = 250
```

This example sends the results of the query 'select * from sysobjects' as a text file attachment to user1. It includes a subject line for the mail and a message that will appear before the attachment. The @width option is used to prevent line breaks in the output lines.

E.

```
create table #texttab (c1 text)
insert #texttab values ('Put your long message here.')
declare @cmd varchar(56)
declare @tabname sysname(30)
select @tabname = name from tempdb.dbo.sysobjects
where name like '#texttab%' +
convert (varchar(5), @@spid) + '%'
select @cmd = 'select c1 from tempdb.dbo.' + @tabname
exec master.dbo.xp_sendmail
'user1', @query = @cmd, @no_header= 'True'
drop table #texttab
```

This example shows how to send a message longer than 255 characters. Because the @message parameter is limited to the length of a VARCHAR (as are all stored procedure parameters), this example writes the long message into a temporary table consisting of a single text column. The contents of this temporary table are then sent in mail by using the @query parameter. The @query parameter makes a separate connection, so the name of the temporary table must be retrieved from the sysobjects table in tempdb.

Messages

Msg 17914 "An invalid recipient was specified."
One or more of the recipients you specified could not be found.

If SQL Server detects an error when executing the **@query** statement, you will receive the full error message at the client, as if you had executed the query directly. Possible errors include Msg 229 "SELECT permission denied" and Msg 208 "Invalid object name." See the *Microsoft SQL Server Transact-SQL Reference* for information on errors pertaining to specific query statements.

Permissions

Execute permission defaults to the system administrator, who can grant permission to others.

See Also

xp_startmail Extended Stored Procedure

xp_stopmail Extended Stored Procedure

Network Support

Banyan VINES Support

Banyan(R) VINES(R) support for Windows NT-based clients and servers is available only for SQL Server on the Intel platform; it is not currently available on the MIPS or Alpha AXP platforms.

SQL Administrator Communication with SQL Monitor

SQL Administrator must communicate with SQL Monitor to complete tasks such as scheduled backups. SQL Administrator uses configuration entries to determine how to connect to SQL Monitor. If you use the default Net-Library, SQL Administrator will usually be able to connect to SQL Monitor using the defaults, and you will not need to adjust these configuration entries. If you are connecting to SQL Server using a logical server name (set up using the Advanced button of the SQL Client Configuration Utility), you will need to configure these entries using a text editor in Windows or the Registry Editor in Windows NT.

For clients running Windows, the SQL Server Setup program puts the following default lines in the [SQLMONITOR] section of WIN.INI:

```
dbnmp3=?1:\\?1\pipe\winsql\backup
dbmsspx3=?1:?1_mon
dbmsvin3=?1@?2:?1_mon?2
dbmssoc3=?1,?2:?1,1434
```

For clients running Windows NT, the SQL Server Setup program puts the following default REG_SZ values in the HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SQLServer\Client\Monitor key of the Windows NT Registry:

```
dbnmpntw     ?1:\\?1\pipe\winsql\backup
dbmsspxn     ?1:?1_mon
dbmsvinn     ?1@?2:?1_mon?2
dbmssocn     ?1,?2:?1,1434
```

These entries describe the default rules for each Net-Library that SQL Administrator uses to build a SQL Monitor connection string from the SQL Server name. You can override these rules by using a configuration entry that matches the SQL Server name. If SQL Administrator finds an entry that matches the SQL Server name, it will use that entry to attempt to connect to SQL Monitor. If no entries match the SQL Server name, SQL Monitor will use the default rules.

To override the default rules for a specific SQL Server, directly specify a SQL Monitor connection string using the following format:

servername=*netlib_name*,*monitor_connection_string*

where

servername
Is the SQL Server name.

netlib_name
Is the name of the Net-Library file (not including the .DLL extension) used to communicate with SQL Monitor.

monitor_connection_string
Specifies what value SQL Monitor is listening on.

For example, to use the default SQL Monitor named pipe when connecting to a SQL Server named GIZMO from Microsoft SQL Administrator:

```
gizmo=\\gizmo\pipe\winsql\backup
```

You can modify the default rules if they don't fit your environment. To specify a rule for building the SQL Monitor connection string given a specified SQL Server name, use the following format:

netlib_name=*server_source_template*:*monitor_subst_template*

where

netlib_name

Is the name of the default Net-Library file (not including the .DLL extension) used by the client.

server_source_template

Is a string template for the SQL Server name. This template can contain up to nine tagged expressions. Each tagged expression uses the format:

?*n*[*string*]

where

n

Is a single digit 1 through 9.

string

Is an optional case-sensitive string of zero or more characters to be searched for in the SQL Server name. This string terminates at the next '?' in the *server_source_template* or at the end of the SQL Server name.

The SQL Server name is parsed to build the tagged expressions. All characters up to, but not including, *string* are included in a tagged expression. Any subsequent portions of the SQL Server name include *string*.

For example, given the following *server_source_template*:

```
?1xxx?2
```

If the specified SQL Server name is AAAxxxBBB, then the following tags would be built:

```
?1 = AAA
?2 = xxxBBB
```

monitor_subst_template

Is the substitution template, used with the above tagged expressions, to build the SQL Monitor connection string. This string can contain tagged expressions using the format ?*n*. Each tagged expression is substituted with the corresponding tags built from the *server_source_template*. Any characters other than ?*n* are used directly to construct the SQL Monitor connection string.

Using the example above, and given the following *monitor_subst_template*:

```
?1_mon@test_?2
```

the resulting SQL Monitor connection string would be:

```
AAA_mon@test_xxxBBB
```

TCP/IP Windows Sockets Client Support

For clients running Windows for Workgroups or Windows NT, SQL Server supports client communication using standard Windows Sockets as the IPC method across the TCP/IP protocol.

The following client Net-Library files, supported for connecting only to Microsoft SQL Server, are provided with SQL Server:

- DBMSSOCN.DLL

  Windows NT-based Windows Sockets client. The Windows Sockets Net-Library for Windows NT is supported on Windows NT Server version 3.5, Windows NT Workstation version 3.5, Windows NT Advanced Server version 3.1, and Windows NT version 3.1.
- DBMSSOC3.DLL

  Windows-based Windows Sockets client. The Windows Sockets Net-Library for Windows is supported on Windows for Workgroups version 3.11 with Microsoft TCP/IP for Windows for Workgroups version 1.0 and on Windows for Workgroups version 3.11 with the new Microsoft TCP/IP-32 for Windows for Workgroups 3.11.

  The Windows Sockets Net-Library for Windows is also supported on the Windows 3.1 environment (WOW) of Windows NT Server version 3.5, Windows NT Workstation version 3.5, Windows NT Advanced Server version 3.1, and Windows NT version 3.1

  Note: Windows clients running the new Microsoft TCP/IP-32 for Windows for Workgroups 3.11 stack and attempting socket connections to the server must upgrade to the newest DBMSSOC3.DLL, which is included with SQL Server version 4.21A.
- DBMSSOC.EXE

  MS-DOS-based Microsoft TCP/IP socket client. The Microsoft TCP/IP socket Net-Library for MS-DOS is supported for MS-DOS-based applications running on Windows for Workgroups 3.11 with Microsoft TCP/IP for Windows for Workgroups version 1.0. It is not supported for the MS-DOS environment of Windows NT version 3.1, nor with the new Microsoft TCP/IP-32 for Windows for Workgroups 3.11 (because these environments support only the Windows Sockets specification).

The Windows Sockets Net-Libraries have been extensively tested on the supported platforms for connecting to Microsoft SQL Server for Windows NT. Support for other TCP/IP protocols that support Windows Sockets is planned for future releases. Using these Net-Libraries with other TCP/IP protocols or to connect to other than Microsoft SQL Server for Windows NT has not been extensively tested, and their use in these ways is not guaranteed. Any feedback you have on the Windows Sockets Net-Libraries is welcome on the Microsoft SQL Server (GO MSSQL) forum on CompuServe(R).

New TCP/IP-32 Protocol Stack for Windows for Workgroups Clients

A new client protocol stack, Microsoft TCP/IP-32 for Windows for Workgroups 3.11, is distributed with Windows NT Server version 3.5 (and will also be made available by Microsoft from other distribution sources). It allows Windows for Workgroups clients to upgrade to the improved 32-bit protected-mode stack and take advantage of new TCP/IP services provided by Windows NT Server version 3.5 (such as the new DHCP and WINS services).

Note: This protocol stack does not support the real-mode sockets as implemented in earlier versions of TCP/IP for Microsoft LAN Manager and Windows. All socket support is provided through the Windows Sockets specification. Because of this, DBMSSOC.EXE (the MS-DOS-based Microsoft TCP/IP socket client Net-Library) is not supported in this configuration.

Windows clients running the new Microsoft TCP/IP-32 for Windows for Workgroups 3.11 stack and attempting socket connections to the server must upgrade to the newest DBMSSOC3.DLL, which is included with SQL Server version 4.21A.

Simplified System Administration Using DHCP and WINS

Windows NT version 3.5 provides for easy administration of large TCP/IP networks by offering the Dynamic Host Configuration Protocol (DHCP) service for automatic TCP/IP configuration, and the Windows Internet Name Service (WINS) for dynamic mapping of network names and addresses. This enables users to operate in large scale TCP/IP networking environments with little administrative support. (For information on configuring clients, see the following section, "Client Configuration.")

Note that to take advantage of the DHCP and WINS services a Windows for Workgroups client must be using the new TCP/IP-32 protocol stack and must have these services enabled.

Client Configuration

To configure a client running Windows or Windows NT to use the Windows Sockets Net-Library, follow the steps under "Setting Up Server Connections" in Chapter 3 of the *Microsoft SQL Server Configuration Guide*. In the DLL Name box, type the appropriate Windows Socket Net-Library name (DBMSSOCN for Windows NT, DBMSSOC3 for Windows). The Connection String uses the format:

ip_address,[*socket_number*]

where *ip_address* is the IP address of the computer running SQL Server, and *socket_number* is the optional socket number that SQL Server is listening on.

It is also possible to configure a client running Windows or Windows NT to use the Windows Sockets Net-Library by default. From the SQL Client Configuration Utility, in the Default Network box select TCP/IP Sockets. Using this method, connections can be established by using the *ip_address*, [*socket_number*] directly as the SQL Server name, or by using a server name with the format:

host_name,[*socket_number*]

where *host_name* is a TCP/IP host name that has been defined in the client HOSTS file or on a Domain Name Service (DNS), or is the machine name of a Windows NT Server running SQL Server if WINS is enabled, and *socket_number* is the optional socket number that SQL Server is listening on. For a client running Windows for Workgroups, the HOSTS file is located in the \WINDOWS directory by default. For a client running Windows NT, the HOSTS file is located in the \WINDOWS\SYSTEM32\DRIVERS\ETC directory by default. For other TCP/IP protocols that support Windows Sockets, see your TCP/IP documentation.

To configure a client running MS-DOS to use the Microsoft TCP/IP sockets Net-Library, you use an environment variable before loading the Net-Library TSR (DBMSSOC.EXE). The environment variable *server* is a logical server name, and is set using this format:

set *server=ip_address*,[*socket_number*]

In all cases, if the *socket_number* is not specified, the Net-Library uses 1433, the official Internet Assigned Number Authority (IANA) socket number for Microsoft SQL Server. Because this official IANA socket number was not available at the time, SQL Server version 4.2 used a temporary TCP/IP socket number of 3180 by default.

Memory Considerations Using TCP/IP for Windows for Workgroups Version 1.0

The Microsoft TCP/IP for Windows for Workgroups WINSOCK.DLL requires 47K of low memory, 1.6K of low memory per socket defined, and 32K of low memory per socket opened. If available low memory becomes insufficient or fragmented, you might see the error "There is not enough DOS memory allocated for DOS buffers for all sockets" when attempting to open a socket connection. If this error occurs, try closing some applications, configuring your system to make more conventional memory available, or reducing the number of available sockets. Although Microsoft TCP/IP for Windows for Workgroups supports up to 21 sockets, it is recommended that you set this value to a realistic number of concurrent SQL Server connections.

Note that this applies to clients using Microsoft TCP/IP for Windows for Workgroups version 1.0. These considerations can be avoided by upgrading to the new Microsoft TCP/IP-32 for Windows for Workgroups 3.11.

Windows for Workgroups Clients on Novell NetWare Networks

Microsoft Windows for Workgroups version 3.11 enhances the network integration available with Novell(R) NetWare(R) by offering peer network functionality and connectivity to computers running Windows NT. SQL Server clients running Windows for Workgroups can be configured in several ways using the Network Setup application.

If only Windows support for Novell NetWare is installed (and Microsoft Windows Network is not installed), then you are using the Novell NetWare IPX protocol (IPXODI). This protocol supports SPX as the IPC method for connecting to SQL Server. This means that you should use the SQL Client Configuration Utility to set the default network to Novell IPX/SPX. This configuration will allow Windows for Workgroups to use the IPX/SPX Net-Library to connect to SQL Server for Windows NT running NWLink, and to connect to SQL Server for OS/2 running the Network Integration Kit for Novell NetWare Networks.

If only the Microsoft Windows Network is installed (and Windows support for Novell NetWare is not installed), then you can still use IPX to connect to SQL Server for Windows NT. Use the Drivers button to ensure that the IPX/SPX Compatible Transport With NetBIOS network protocol is installed. This IPX protocol is included with Windows for Workgroups 3.11 and is similar to Windows NT NWLink. This protocol supports named pipes as the IPC method for connecting to SQL Server. This means that you should use the SQL Client Configuration Utility to set the default network to Named Pipes. This configuration will allow Windows for Workgroups to use the named pipes Net-Library to connect to SQL Server for Windows NT running NWLink. However, it will not connect to SQL Server for OS/2, because Windows for Workgroups named pipes are not designed to interoperate with Novell OS/2 Requestor named pipes.

If both the Microsoft Windows Network and Windows support for Novell NetWare are installed, you can use either the named pipes or IPX/SPX Net-Library to connect to SQL Server for Windows NT running NWLink, or use the IPX/SPX Net-Library to connect to SQL Server for OS/2 running the Microsoft Network Integration Kit for Novell NetWare Networks.

It is recommended that the named pipes Net-Library be used, because features such as integrated security and adjustable packet size are available only with named pipes.

Updated Files Required for Windows for Workgroups 3.11-based Clients

Windows NT Server version 3.5 allows clients running Windows for Workgroups version 3.11 to connect to the server via direct host IPX. However, a client running Windows for Workgroups version 3.11 cannot establish named pipe connections over NWLink to SQL Server on Windows NT Server version 3.5 unless that client is provided with some updated Windows for Workgroups files. These files include a new version of the Windows for Workgroups redirector and other network support files that provide improved performance. They are distributed on the Windows NT Server version 3.5 CD, in the \CLIENTS\WFW\UPDATE directory. The updated files are:

NDIS.386
NET.EXE
NETAPI.DLL
NWNBLINK.386
VNETSUP.386
VREDIR.386

These files should be used only for Windows for Workgroups version 3.11. They will allow named pipe connections to work properly to Microsoft SQL Server over direct host IPX. To install these new files on a computer running Windows for Workgroups:

1. Copy all the files from the \CLIENTS\WFW\UPDATE directory on the Windows NT Server CD into the local C:\WINDOWS\SYSTEM directory of the Windows for Workgroups 3.11-based client.

   Note: Remember to use the Windows NT Server CD. The Microsoft SQL Server CD does not contain these files.

2. On the Windows for Workgroups 3.11-based client, move the NET.EXE file from C:\WINDOWS\SYSTEM to C:\WINDOWS.

Because these files provide a general maintenance upgrade along with a number of fixes and improvements, copying the updated files is the preferred solution. However, there is an alternative solution. On any Windows for Workgroups 3.11-based client that does not have the updated files, you can edit the SYSTEM.INI file of the client to set:

```
[network]
DirectHost=off
```

This allows that Windows for Workgroups 3.11-based client to establish named pipe connections over NWLink to SQL Server on Windows NT Server version 3.5.

Novell NetWare and Microsoft Windows 3.1 Software Patches

If you are experiencing occasional system hangs, black screens, frozen mouse pointers, or EMM386 errors on your client workstations while using SPX/IPX connections, several patches are available on CompuServe to help eliminate or reduce the frequency of these problems.

The majority of these types of problems are solved by upgrading the Novell client software. DOSUP9.EXE and WINUP9.EXE are distributed by Novell in the Novell Software Library (GO NOVLIB).

A small percentage of these failures can be caused by a problem in the Windows virtual timing device. Microsoft is distributing a patch (WW0863.EXE) for this problem in the Microsoft Software Library (GO MSL).

OS/2 2.1 Network Client Support

Installation of Microsoft LAN Manager version 2.2, IBM LAN Server version 2.0 (Entry System), or IBM(R) LAN Server version 3.0 on OS/2 2.1-based workstations is required to enable the use of named pipe client connections. With these versions, named pipe clients are supported in both native OS/2 2.1 and DOS-OS/2 environments.

Installation of the Novell NetWare Requester for OS/2 version 2.01 on OS/2 2.1-based workstations is required to enable the use of IPX/SPX client connections. With this version, IPX/SPX clients are supported in both native OS/2 2.1 and DOS-OS/2 environments.

Installation of the Banyan VINES version 5.52(5) client software on OS/2 2.1-based workstations is required to enable the use of VINES SPP client connections. With this version, VINES SPP clients are supported in both native OS/2 2.1 and DOS-OS/2 environments.

It is recommended that you run Windows-based SQL Server client applications in their native environment, on Microsoft Windows(TM) version 3.1. However, Windows-based applications that access SQL Server using named pipes will run on OS/2 version 2.1 as long as you are using Microsoft LAN Manager version 2.2 or IBM LAN Server version 3.0. Internal testing has found the WIN-OS/2 environment to be unstable with other networking software.

Tools Enhancements

- SQL Transfer Manager
- SQL Administrator Compatibility
- SQL Monitor Enhancements
- SQL Administrator Scheduled Backups
- SQL Object Manager and BCP
- Additional Information About Database Backups
- Additional Information About BCP and ISQL

SQL Transfer Manager

Microsoft SQL Server version 4.21 includes SQL Transfer Manager. SQL Transfer Manager provides an easy, graphical way to transfer objects and data from a Microsoft-based SQL Server or a SYBASE(R)-based SQL Server to Microsoft SQL Server. (For example, transferring data from Microsoft SQL Server running on an Intel-based server to a SQL Server on a different architecture, or transferring data from a SYBASE SQL Server on UNIX(R).)

You can also use this tool to transfer data from a server with one sort order to a server with a different sort order. Note, however, that this tool does not convert extended characters, so you cannot use it for conversion from one code page to another.

For detailed information about using SQL Transfer Manager, see the online Help provided with the utility.

SQL Administrator Compatibility

Although the version of SQL Administrator that runs against SQL Server for OS/2 4.2/4.2A will also work against SQL Server for Windows NT, it is recommended that you upgrade SQL Server client tools on any Windows 3.1-based clients that you plan to run against SQL Server. This enables you to use additional tools and features available with SQL Server (such as SQL Object Manager, ISQL/w, and the SQL Client Configuration Utility). SQL Server for OS/2 version 4.2B contains these tools (version 4.2/4.2A does not). You can upgrade your tools to the very latest version, 4.21, by using the Windows 3.1 Utilities disk provided with this release of SQL Server for Windows NT. If you use the Utilities disk, you must also upgrade the existing SQL Server for OS/2 installation if it is not version 4.2B (for example, it is version 4.2 or 4.2A) by running INSTMSTR.SQL, INSTCAT.SQL, ADMIN2.SQL, and OBJECT2.SQL, which are found in the \SQL\INSTALL directory of the SQL Server for Windows NT installation.

SQL Monitor Enhancements

SQL Monitor and Changing the SA Password

If you change the SA password in SQL Server and want to start SQL Monitor as a service (instead of starting SQL Monitor from the command line), you must change the SA password in SQL Monitor as well. To do this, use the /NEWPASSWORD = parameter in netsql start sqlmonitor. For example:

```
netsql start sqlmonitor /NEWPASSWORD=sqlsa
```

Note that starting SQL Monitor with the password parameter (/P=) overrides the set SA password.

SQL Monitor Is Now Mail Enabled

If you have configured SQL Server to be mail enabled, SQL Monitor can send mail messages to specified recipients about scheduled backup events. Use the Scheduled Backup Event Entry dialog box in SQL Administrator to specify a list of recipients. You can specify up to 60 characters; separate names by semicolons(;). After you have specified a list of recipients, those people will be sent mail whenever a backup occurs.

SQL Administrator Scheduled Backups

Transaction Log Backups

In previous versions of SQL Administrator, when SQL Monitor attempted to perform a scheduled backup for a transaction log and SQL Monitor terminated sometime during the process, SQL Monitor would dump the transaction log again when restarted, regardless of whether the previous backup was successful. This posed a problem because SQL Monitor would overwrite the previous transaction log dump, even though it might have completed, so the chain of transaction log backups would be incorrect. To solve this problem, SQL Monitor now handles failure during a transaction log dump in the following order:

1. SQL Monitor records a dump in progress while it is dumping the transaction log.

2. If SQL Monitor fails and then starts up again and sees that a dump was in progress, but it does not know whether that dump completed, SQL Monitor records an error to the Event Log and also sends a mail message about the failure to a list of recipients (if SQL Server is mail enabled and SQL Administrator is set up to send messages to a list of recipients).

3. Further dumps of the transaction log are suspended. After the user either dumps the full database or determines that the transaction log dump is complete, the user must reactivate the scheduled transaction log dumps. (This is done by selecting Yes under Enable in the Scheduled Backup Event Entry dialog box.)

Note that if the scheduled backup is a database backup, steps 1 and 2 are followed, and the schedule for database backups will continue as is and not be disabled.

Performing Long Scheduled Backups

If you are using SQL Administrator to perform scheduled backups of large databases (for example, those greater than 100 megabytes), you will likely need to increase the SQL Monitor /sqltimeout option to a length of time long enough to complete the backup.

SQL Object Manager and BCP

SQL Object Manager version 4.2 used bulk copy files based on the ANSI character set if AutoANSItoOEM was turned on, or based on the character set of the SQL Server if AutoANSItoOEM was turned off. The character mode BCP.EXE versions 4.20 and 4.21 utilities use bulk copy files based on the character set of the SQL Server. With version 4.21, SQL Object Manager now uses the OEM character set. If you used SQL Object Manager version 4.2 with AutoANSItoOEM turned on to bulk copy out character data that contained extended characters (character positions 128-255), the file is based on the ANSI character set; if you bulk copy the files back in using SQL Object Manager version 4.21, these extended characters will be converted incorrectly.

Additional Information About Database Backups

Database Names When Dumping and Loading

When you dump a database to tape, only the first 17 characters of the database name are stored. If the database name is longer than 17 characters and you try to restore the database, you will receive an error message indicating that the names are different. For this reason, it is recommended that you limit database names to 17 characters if you plan to back them up to tape. If you are using SQL Administrator to restore the database, you can choose to continue, even if the names are different, as long as you're sure that the actual databases are the same. If you are using ISQL, however, the restore process will be terminated.

Performing Multivolume Backups Using a QIC Tape Drive

If you are using a QIC tape drive to perform multivolume dump or load operations using ISQL, you must have the console program running. Do not use a batch operation to perform multivolume tape operations with this drive.

Additional Information About BCP and ISQL

Default Packet Size In ISQL and BCP

The documentation for the /a packet_size parameter of ISQL and BCP gives an incorrect value for default packet size. The correct default packet size for ISQL is 512. The correct default packet size for BCP on Windows NT is 4096; on MS-DOS and OS/2, the default packet size is 512.

BCP Out

In addition to the BCP support documented in the *Microsoft SQL Server Transact-SQL Reference*, the BCP utility now also supports copying data out of a view. This feature allows you to copy specific columns, add a WHERE clause, or perform special formatting such as changing data formats using the CONVERT function.

Cache Management Improvements

A new system process, Lazywriter, has been added to Microsoft SQL Server version 4.21. The Lazywriter's main task is to flush out batches of dirty, aged buffers and make them available to user processes.

Previously, individual processes would search the data cache for an available buffer. If a clean buffer was not found, the user process would select a dirty buffer and be required to flush the contents of the buffer to disk before claiming it. In the case where most buffers in the data cache were dirty (due to extensive update activity), the overhead of searching for free data buffers and having to individually flush dirty buffers to disk caused degradation in performance. One way of avoiding this problem previously was to reduce the recovery interval and have checkpoint flush the buffer cache in batches more often so that individual processes wouldn't have to do it one at a time.

With the introduction of Lazywriter, the need to checkpoint frequently for the purpose of creating available buffers has been eliminated. The batch I/O size used by Lazywriter can be set by the max async io parameter of **sp_configure**. This parameter controls both the checkpoint's and Lazywriter's batch I/O size, and it can be tuned to maximize the I/O throughput for specific hardware platforms and I/O subsystems.

The Lazywriter process automatically starts flushing buffers when the number of available free buffers falls below a certain threshold, and it stops flushing buffers when this number goes ~5-6% above the threshold. This threshold value is specified as a percentage of the total number of buffers in the buffer cache. The default threshold is set to 3% of the buffers in the data cache and should be sufficient in most situations. However, if necessary, the Lazywriter threshold can be modified using the BLDMASTR utility with the -y switch, as follows:

```
bldmastr -dc:\sql\data\master.dat -yLRUThreshold=x
```

where *x* is a percentage of the size of the data cache (values between 1 and 40 are valid).

CAUTION: Be careful to type the command exactly as shown. The command should complete nearly immediately and not prompt for any input whatsoever. If you are prompted for input, you have made a syntactic error. Use CTRL+C to terminate the utility immediately. INCORRECT USE OF THE BLDMASTR UTILITY CAN INADVERTENTLY CHANGE YOUR MASTER DATABASE. The LRUThreshold argument is case-sensitive.

Because cache management has been enhanced, some meanings of the SQL Server counters in the Performance Monitor have changed. For the most up-to-date definitions of the SQL Server performance counters, see the Explain text in the Performance Monitor.

Dirty buffers

Dirty buffers are buffers that contain changes that must be written back to disk before the buffer can be reused for a different page.

Clean buffers

Clean buffers are buffers that remained unchanged since the last read from disk

Enhanced NOCOUNT Option of the SET Statement

Previously, even if the NOCOUNT option of the SET statement was set (so that no messages about the number of rows affected by a SQL statement were displayed in a client session), the messages were still being sent across the network. For this release, the NOCOUNT option has been enhanced so that when it is set, no row count messages are sent across the network at all. This can reduce network traffic when executing a stored procedure containing a large number of SQL statements across a wide area network (WAN) or when using remote access or dial-up networks.

You can use this option from within a trigger or a stored procedure, or you can set it on a user basis. The system administrator can also set this option server-wide by using the trace flag 3640 when starting SQL Server. Note that when you use this trace flag, even if a client session wants to see the rows affected messages, the messages will not be sent.