SQL Server Driver

The SQL Server driver supports the SQL Server database system available from Microsoft and Sybase, Inc. It also supports the Sybase Net gateway to DB2.

The driver filename is IVSS*nn*.DLL.

ODBC SQL Server Driver Setup dialog box

Data Source Name
A string that identifies this SQL Server data source configuration in ODBC.INI. Examples include "Accounting" or "SQL Server-Serv1."

Description
An optional long description of a data source name. For example, "My Accounting Database" or "SQL Server on Server number 1."

Server Name
The name of the server that contains the desired database.

Database Name
The name of the database to which you want to connect by default. If you do not specify a value, the default database defined by SQL Server is used.

The following values are optional:

Server List
A comma-separated list of servers that will appear in the logon dialog box.

Database List
The databases that will be available in the SQL Server Logon Options dialog box. Separate the names with commas.

Default Logon ID
The default logon ID used to connect to your SQL Server database. This ID is case-sensitive. A logon ID is required only if security is enabled on your database. Your ODBC application may override this value or you may override this value in the logon dialog box or connection string.

Language
The national language to be used by the client. The default is English.

Application Name
The name SQL Server uses to identify your application.

Workstation ID
The workstation ID used by the client.

Cursor Cache Size
The number of cursors the cursor cache can hold. The driver creates a cache of statements; each statement represents an open connection to SQL Server. The cursor cache increases performance but uses database resources. The default is 1 (one cursor).

Yield Proc
A numeric value that determines whether you can work in other applications when SQL Server is busy. This attribute is useful to users of ODBC applications. Valid values are

- 0 (peek and dispatch), which causes the driver to check the Windows message queue and send any messages to the appropriate Windows application.
- 1 (no yielding, the default), which does not let you work in other applications.
- 3 (dispatch via Windows Yield function), which turns control over to the Windows kernel. The Windows kernel checks the message queue and sends any messages to the appropriate application window.

It is recommended that you use the value 1.

Character Conversion
This value controls the character set conversion between SQL Server (version 4.8 or later) and a client application.

If you omit this value, no character conversion takes place on your server.

Common values include iso-1 for ISO-8859-1, cp850 for Code Page 850, roman8 for Roman8 character set, and SJIS for a Japanese character set. See your SQL Server documentation for a complete list of values.

Cancel Behavior

A value that specifies how a previously executed statement should be canceled. Valid values are

- 0 fetches all of the remaining records if the statement was a Select.
- 1 cancels the statement by calling dbcancel. This is the default and should be used if dbcancel is supported in your client/server configuration.
- 2 closes the connection to the server for the statement. Use this value only if dbcancel is not supported for your configuration and the performance of fetching all remaining records is unacceptable.

Using Gateway

Select this check box if you are using Sybase Net-Gateway to access a DB2 database with this data source.

NETAPI.DLL Library Available

The driver uses NETAPI.DLL to get the name of your workstation. Most major PC networks support this feature. If your network supports this capability, select this option. If you supply a workstation ID, this field is ignored.

Two-Phase Commit

This check box, when selected, enables you to have two active statements within a transaction, using the SQL Server two-phase commit services. The active statements may deadlock if they reference the same SQL Server table.

{button ,AL(`H_SYB_CONFIGURING_DATA_SOURCES_STEPS;H_SYB_ISOLATION_AND_LOCK_LEVELS_SUPPORTED_REF;H_SYB_NUMBER_OF_CONNECTIONS_AND_STATEMENTS_SUPPORTED_REF;H_SYB_ODBC_CONFORMANCE_LEVELS_REF;H_SYB_SYSTEM_REQUIREMENTS_REF;',0)} See related topics

SQL Server Logon Options

Database Name
Type the name of the initial SQL Server database to connect to or select the name from the drop-down box. If not supplied, the default database defined by SQL Server is used.

Workstation ID
Type the name of your workstation.

Logon to SQL Server dialog box

Server Name

Type the name of the server containing the SQL Server database tables you want to access (case-sensitive) or select the name from Server Name box, which displays the server names you specified in the Setup dialog box.

Login ID

If required, type your Login ID (case-sensitive).

Password

If required, type your password for the system (case-sensitive).

{button ,AL(`H_CONNECTING_TO_SQL_SERVER_USING_A_CONNECTION_STRING_REF;H_CONNECTING_TO_SQL_SERVER_USING_A_LOGON_DIALOG_BOX_STEPS;H_SYB_NUMBER_OF_CONNECTIONS_AND_STATEMENTS_SUPPORTED_REF;',0)} See related topics

System requirements

You must have the appropriate DB-Library and Net-Library installed to gain access to Microsoft SQL Server or Sybase SQL Server databases.

Your database must support catalog stored procedures.

The DB-Library for Windows NT or Windows 95 is NTWDBLIB.DLL.

The Net-Library you need depends on the network protocol used to connect to SQL Server. For example, Named Pipes requires DBNMPNTW.DLL.

Contact your Microsoft SQL Server or Sybase SQL Server vendor to obtain the appropriate DB-Library and Net-Library.

If you attempt to configure a data source and you do not have NTWDBLIB.DLL on your path or in your Windows \SYSTEM32 or Windows 95 \SYSTEM directory, the following message appears:

"The setup routines for the INTERSOLV SQL Server ODBC driver could not be loaded. You may be low on memory and need to quit a few applications."

Configuring data sources

To configure a SQL Server data source, do the following:

1. Start the ODBC Administrator.

   A list of data sources appears.

2. If you are configuring a new data source, click Add.

   A list of installed drivers appears.

3. Select INTERSOLV 2.10 SQLServer and click OK.

4. If you are configuring an existing data source, select the data source name and click Setup.

   The setup dialog box appears.

5. Specify the values you want to use.

6. Click Translate to perform a translation of your data from one character set to another.

   The Select Translator dialog box appears, in which you select a translator. INTERSOLV provides a translator named INTERSOLV OEM ANSI that translates data from the IBM PC character set to the ANSI character set. The translators that are listed in this dialog box are determined by the values listed in the ODBC Translators section of ODBCINST.INI.

7. Click OK to close the Select Translator dialog box and perform the translation.

8. Click OK to write these values to ODBC.INI.

   These values are now the defaults when you connect to the data source. You can change the defaults by configuring your data source again. You can override the defaults by connecting to the data source using a connection string with alternate values.

Connecting to SQL Server using a Logon dialog box

Some ODBC applications display a logon dialog box when you are connecting to a data source. In these cases, the data source name has already been specified.

In this dialog box, do the following:

1. Type the name of the server containing the SQL Server database tables you want to access (case-sensitive), or select the name from the Server Name drop-down box, which displays the server names you specified in the setup dialog box.
2. If required, type your case-sensitive login ID.
3. If required, type your case-sensitive password for the system.
4. (Optional) Click Options to display the SQL Server Logon Options dialog box and specify the initial SQL Server database to connect to and the name of your workstation.
5. Click OK to log on to the SQL Server database installed on the server you specified and to update the values in ODBC.INI.

Connecting to SQL Server using a connection string

If your application requires a connection string to connect to a data source, you must specify the data source name that tells the driver which ODBC.INI section to use for the default connection information. Optionally, you may specify *attribute*=*value* pairs in the connection string to override the default values stored in ODBC.INI. These values are not written to ODBC.INI.

You can specify either long or short names in the connection string. The connection string has the form:

DSN=*data_source_name*[;*attribute*=*value*[;*attribute*=*value*]...]

An example of a connection string for SQL Server is

DSN=Accounting;DB=PAYROLL;UID=JOHN;PWD=XYZZY

The following table gives the long and short names for each attribute, as well as a description.

The defaults listed in the table are initial defaults that apply when no value is specified in either the connection string or in the data source definition in ODBC.INI. If you specified a value for the attribute when configuring the data source, that value is your default.

Attribute	Description
DataSourceName (DSN)	A string that identifies a SQL Server data source configuration in ODBC.INI. Examples include "Accounting" or "SQL Server-Serv1."
ServerName (SRVR)	The name of the server containing the SQL Server tables you want to access.
Database (DB)	The name of the database to which you want to connect.
LogonID (UID)	The case-sensitive logon ID used to connect to your SQL Server database. A logon ID is required only if security is enabled on your database. If so, contact your system administrator to get your logon ID.
Password (PWD)	A case-sensitive password.
Language (LANG)	The national language to be used by the client. The initial default is English.
ApplicationName (APP)	The name SQL Server uses to identify your application.
WorkstationID (WKID)	The workstation ID used by the client.
CursorCacheSize (CCS)	The number of cursors the cursor cache can hold. The driver creates a cache of statements; each statement represents an open connection to SQL Server. The cursor cache increases performance but uses database resources. The initial default is 1.
YieldProc (YLD)	A numeric value that determines if you can work in other applications when SQL Server is busy. This attribute is useful to users of ODBC applications. Valid values are • YieldProc=0 (peek and dispatch) causes the driver to check the Windows message queue and send any messages to the appropriate Windows application. • YieldProc=1 (no yielding, the initial default) does not let you work in other applications. • YieldProc=3 (dispatch via Windows Yield function) turns control over to the Windows kernel. The Windows kernel checks the message queue and sends any messages to the appropriate application window. It is recommended that you use YieldProc=1. YieldProc=0 and YieldProc=3 do not work with all Windows applications.
CharConv (CC)	A value that controls the character set conversion between SQL Server (version 4.8 or later) and a client application. Common values include iso-1 for ISO-8859-1, cp850 for Code Page 850, roman8 for the Roman8 character set, and SJIS for a Japanese character set. See your SQL

	Server documentation for a complete list of values.
Cancel (CAN)	A value that specifies how a previously executed statement should be canceled. Valid values are • Cancel=0 fetches all remaining records if the statement was a Select. • Cancel=1 cancels the statement by calling dbcancel. Set Cancel=1 if dbcancel is supported in your client/server configuration. This is the initial default. • Cancel=2 closes the connection to the server for the statement. Set Cancel=2 only if dbcancel is not supported for your configuration and the performance of fetching all remaining records is unacceptable.
Gateway (GW)	A value that specifies whether you are using the Sybase Net-Gateway to access a DB2 database with this data source. Set Gateway=1 if this is the case. Otherwise, set Gateway=0 (the initial default).
TwoPhaseCommit (TPC)	This attribute lets you have two or more active statements within a transaction, using the SQL Server two-phase commit services. Set TwoPhaseCommit=1 to use two-phase commit. The active statements may deadlock if they reference the same SQL Server table. Otherwise, set TwoPhaseCommit=0 (the initial default).
Netapi (NAPI)	A value that specifies whether NETAPI.DLL is available. Netapi=0, the initial default, indicates it is not available; Netapi=1 indicates it is available. If you supply a value for the WorkstationID attribute, this attribute is ignored.
ModifySQL (MS)	This attribute is provided for backward compatibility. It determines whether the driver modifies SQL statements to conform to ODBC specifications or passes the SQL statement directly to SQL Server. Specify ModifySQL=1 to have the driver modify the SQL statement to conform to ODBC specifications. Specify ModifySQL=0 to have the driver understand SQL dialects found in earlier drivers. The default is 1.

Data types

The SQL Server data types are mapped to the standard ODBC data types as follows:

SQL Server	ODBC
binary	SQL_BINARY
bit	SQL_BIT
char	SQL_CHAR
datetime	SQL_TIMESTAMP
float	SQL_FLOAT
image	SQL_LONGVARBINARY
int	SQL_INTEGER
money	SQL_DECIMAL
real	SQL_REAL
smalldatetime	SQL_TIMESTAMP
smallint	SQL_SMALLINT
smallmoney	SQL_DECIMAL
sysname	SQL_VARCHAR
text	SQL_LONGVARCHAR
timestamp	SQL_VARBINARY
tinyint	SQL_TINYINT
varbinary	SQL_VARBINARY
varchar	SQL_VARCHAR

Isolation and lock levels supported

SQL Server supports isolation level 1. SQL Server supports page-level locking.

ODBC conformance levels

The SQL Server driver supports the Core, Level 1, and Level 2 API functions listed in Supported ODBC Functions. In addition, the following Level 2 functions are supported:

- SQLBrowseConnect
- SQLColumnPrivileges
- SQLForeignKeys
- SQLPrimaryKeys
- SQLProcedureColumns
- SQLProcedures
- SQLTablePrivileges

The driver supports the minimum SQL grammar.

Number of connections and statements supported

The SQL Server database system supports multiple connections. With two-phased commit, SQL Server supports multiple statements per connection. Otherwise, SQL Server supports a single statement per connection if SQL_AUTOCOMMIT is 0 and multiple statements per connection if SQL_AUTOCOMMIT is 1.