

SQL TOOLKIT FOR G

Version 5.0

These release and installation notes contain information about the SQL Toolkit for G, which you can use to communicate with local or remote databases from both LabVIEW and BridgeVIEW. The SQL Toolkit works under Windows 3.1 and Windows 95/NT. With the SQL Toolkit, you can save, fetch, and update records of data in more than 30 different database formats using high-level data mapping functions.

Contents

About the SQL Toolkit for G	2
Installation Instructions	2
Technical Support	3
Distributing Applications that Use the SQL Toolkit	4
Licensing Issues	4
Bookkeeping	5
Create Your Application	5
Verify Communication	5
Install Drivers	6
Configure DSNs.....	6
Upgrading from Previous Versions.....	6
Upgrading from Version 2.0.....	6
Bug Fixes in Version 5.0	7
Upgrading from Version 1.14.....	7
Features Introduced in Version 2.0.....	7
Compatibility Issues when Upgrading from Version 1.14 or Earlier.....	9
Bug Fixes in Version 2.0	10
Upgrading from Version 1.13.....	11
Upgrading from Version 1.0.....	11

About the SQL Toolkit for G

The SQL Toolkit is designed to communicate with databases that have an Open Database Connectivity (ODBC) interface. ODBC is a standard interface for communicating with databases using Structured Query Language (SQL).

Previous versions of this toolkit were known as DatabaseVIEW, and were sold by Ellipsis Corporation. National Instruments has purchased this product fully and renamed it to the *SQL Toolkit for G*. Contact National Instruments Corporation for sales and support information.

Installation Instructions

The SQL Toolkit is distributed on CD. If you need to install on a system without a CD drive, either mount it as a volume over a network or create a set of floppies from the directory on the CD.

Version 5.0 of the SQL Toolkit is designed to work with LabVIEW 5.0 or later and BridgeVIEW 2.0 or later. The Windows 3.1 files are in `\win31`, and the Windows 95 and Windows NT files are in `\win95NT`.

The CD also includes a toolkit version that is compatible with LabVIEW 4.x or BridgeVIEW 1.x. You can find the installer for this version in the `\OLD` directory on the CD.

To install the SQL Toolkit, run `SETUP.EXE` from the appropriate directory.

Many ODBC drivers ship with databases. For example, Microsoft Access includes ODBC drivers with its installation. The SQL Toolkit includes a set of drivers created by Intersolv for communicating with a number of industry standard databases including dBASE, Oracle, INFORMIX, SQL Server, and others. During the installation you are prompted to designate which Intersolv ODBC drivers you want to install, if any. By default, the installer installs a dBase driver. You must install the driver to use the examples provided in the `examples\` directory. The examples use the dBase driver to read from and write to a sample database.

In addition to drivers, the installer places examples in the `examples\sql` directory, and VIs and menus in the `vi.lib\addons` directory. The SQL VIs that are installed in `vi.lib\addons` automatically appear in the function menu as **Functions»SQL**.

The installer deletes previous versions of the SQL Toolkit if you installed them in the same directory.

The installation process creates a program group that includes the ODBC Administrator, which is used to configure ODBC drivers. Use the ODBC Administrator to define the data sources you can access using the SQL Toolkit or other applications that support ODBC. For each data source, specify a name for the source and the database driver with which it is associated. In addition, you can choose to associate a source with a specific database, or leave it as a general source, in which case you must specify the database when you use the SQL Toolkit to connect to the data source.

The installer creates a DBV Demo Dbase data source that is associated with a Dbase driver and the `examples\SQL_exmpl1db` database. Additional data sources are created for each Intersolv driver that you installed. Also, if you have installed a database, such as Access, on your system, its installer created default data sources for that database driver.

See the help files for the ODBC Administrator and the various drivers for additional configuration information.

Technical Support

If you encounter problems communicating with a database, National Instruments suggests that you first check the help file for the driver to see if it describes any known issues with ODBC compliance. If you use an Intersolv driver, look at the help file in the SQL Toolkit program group. This file links to help files in the `Windows\System` directory for each of the ODBC drivers. If you are using a third-party driver, look to see if it installed a help file in `Windows\System` or in the application directory.

If the help file does not address the problem you encountered, check to see if a newer ODBC driver is available for that database. If you are using one of the supplied Intersolv drivers, contact National Instruments. If you are using a third party ODBC driver, check with the vendor who supplies that driver.

The following is a list of some known problems with specific drivers:

- **(Windows 3.1)** With Access 2.0, the Get Table Information VI does not work correctly. Instead of returning the list of tables available for the connection, it returns an empty array and no error code.
- **(Windows 3.1)** With Access 2.0, many of the dialogs in the Database Browser tool do not work because of the problem with the Get Table Information VI (see previous bullet). Many of these dialogs are designed to let you select a table from the tables associated with a database. Because this information cannot be retrieved from Access, many of the options in the Browser tool cannot work correctly.

- **(Windows 3.1)** With Oracle, your system might crash when you connect to a data source. If this happens, you should copy the `qelibfix.dll` from the `sqlfix` directory in your installation to the `Windows\System` directory. The problem occurs because the Oracle drivers are fairly complex—the drivers load a number of DLLs. These drivers require a fair amount of *stack space* to operate. (Stack space is a limited resource for an application). Without the `qelibfix.dll`, there might not be enough stack space to connect to the database.

While this DLL seems to correct the problem with Oracle, it prevents the Connect VI from showing a dialog even if you wire `TRUE` to show a dialog. Consequently, ensure that the connection string you specify includes the data source and any user identification and password used to establish a connection.

For additional support issues, contact National Instruments. Refer to Appendix B, *Customer Communication*, in the *SQL Toolkit for G Reference Manual*, for information about National Instruments technical support.

Distributing Applications that Use the SQL Toolkit

Because the SQL Toolkit uses driver DLLs that must be configured for each system on which the software runs, there is more to creating an executable than simply using the Application Builder. In the following sections, the computer on which you install your application is referred to as the target computer.

Licensing Issues

The SQL Toolkit includes the ODBC Administrator from Microsoft as well as ODBC drivers for a variety of databases including DBASE, Sybase, and INGRES among others.

If you plan to distribute an application, you might need to purchase a run-time license. Most database applications ship with their own ODBC drivers. If the application you plan to distribute communicates with a database that uses an ODBC driver that ships with it, you can distribute the application without a license for the driver.

If you use one of the ODBC drivers included with the SQL Toolkit, you must purchase a run-time license. There are two types of run-time licenses available:

- *Single Database Run-Time License*—Use this license if you are distributing one ODBC driver. This license is available for single or multiple users. To purchase this license, contact National Instruments.
- *Multiple Database Run-Time License*—Use this license if you are distributing two or more ODBC drivers. This license is available for single or multiple users. To purchase this license, contact National Instruments.

Bookkeeping

If your application has embedded data source names (DSN) in any block diagram, make note of the names and their associated databases. If your application communicates with more than one database, or you offer the choice of communicating with more than one, you should purchase the *Multiple Database Run-Time License*.

Create Your Application

Use the instructions included with the Application Builder. If you make an .EXE file, the VIs and subVIs that you used in your application are embedded in the .EXE file. If you make a run-time kernel only, open the top-level VI and **Save With Options**, being sure to save the VIs into the .LLB that you create. This encapsulates the LabVIEW VIs and CIN into your application. Be sure to include any necessary DLLs and drivers. Transfer your application to the target computer.

Verify Communication

If the target computer communicates with a database server or file over a network, verify that the communications work properly. With a database server, install the database vendor's middleware on the target computer, and use an application from the database vendor to verify the communication with the server. If you are simply sharing a database file over a network, verify that the shared volume on the target computer can be mounted. Once communication is verified, you can install the drivers.

Install Drivers

To install the ODBC Administrator and/or ODBC drivers, type `SETUP RUNTIME` at the command line to run the SQL Toolkit setup program. An Installation dialog box appears, and from it you can select the ODBC Administrator and any database drivers you need to install. You are not given the option to install VIs because they should be incorporated into your application.



Note: *If you install any of the ODBC drivers listed, you must purchase a run-time license. See the [Licensing Issues](#) section for more information about purchasing run-time licenses.*

Configure DSNs

Configure the data sources required by your application. Refer to the ODBC Administrator online help if you are unsure how to do this. For databases such as Microsoft Access, you might need to install a database file into a directory on the target computer so the ODBC Administrator can configure your data source. For server databases, networking middleware files might be necessary to configure a data source. Refer to the *Verify Communication* section for more information.

The database portion of your application should now function properly. If you encounter problems, contact National Instruments technical support.

Upgrading from Previous Versions

This section provides an overview of the how to upgrade from previous versions of the SQL Toolkit (or DatabaseVIEW).

Upgrading from Version 2.0

The new toolkit version number reflects the version of LabVIEW for which it is designed. The SQL Toolkit Version 5.0 is designed for LabVIEW Version 5.0 or later. This toolkit is also designed to work with BridgeVIEW 2.0 or later. SQL Toolkit Version 5.0 includes improvements and bug fixes for Version 2.0.

Bug Fixes in Version 5.0

- The G to ODBC Date function produced a date that included fractions of a second. While this should work with all ODBC interfaces, some, including Access, did not handle this properly. To improve compatibility, the fractional portion has been removed.
- The SQL Date to G Date function returns correct hours for hours greater than 9.
- Fetch Query Results now returns an empty array and no error if there is no data to return from an SQL operation. The Fetch Query Results VI previously returned error 4127 if the associated operation did not consist of a query. The Easy SQL VI uses Fetch Query Results regardless of the SQL command. Consequently, if you used Easy SQL to perform an SQL insert, for example, you received an error.
- The `DEMO-Query 1.vi` example in `examples\SQL_dbexsvi.llb` has the query results wired correctly.

Upgrading from Version 1.14

Features Introduced in Version 2.0

Version 2.0 introduced the following new features:

- **Dynamic SQL for higher performance**—Dynamic SQL provides faster performance than standard SQL, particularly when you are performing the same SQL command repeatedly and changing the data only. Dynamic SQL lets you pre-parse an SQL statement with question marks (?) in place of data fields that are filled in as needed. You access the Dynamic SQL VIs from **Functions»SQL»Advanced SQL»Dynamic SQL**.

For example, if you want to insert multiple rows into an SQL database, pass the following dynamic SQL statement to Prepare SQL VI:

```
INSERT INTO tableName (col1Name, col2Name, ...)
VALUES (?, ?, ...)
```

Then, when you are ready to fill in the values, use VIs such as Set SQL Long Parameter or Set SQL Character Parameter to fill in the values for values 1 through N, and then you call the Execute Prepared SQL VI to execute the statement. If you want to execute the statement again, you call the Clear SQL Parameter VI for each field, call the Set Parameter VIs to fill in the new values, and then call the Execute Prepared SQL VI to execute the statement with the new values.

- **Cluster to SQL templates simplify SQL programming**—The Cluster to SQL templates, accessible from **Functions»SQL»Cluster to SQL Templates**, are VIs that make it easy to create `SELECT` and `INSERT` statements, and also to parse data from the table of strings that the SQL VIs return into an array of clusters. Each of these VIs has a cluster on its front panel that you can modify to reflect the structure (names and data types) of the table upon which you want to operate. You can save modified versions of these VIs as part of your application. Together these VIs eliminate the complicated string parsing and construction required for the most commonly used SQL operations.
- **Retrieve query results directly into G data types**—The Template VIs make SQL simple, but they still involve conversion from G data types to SQL strings and vice-versa. The Fetch Data Parsing palette (**Functions»SQL»Advanced SQL»Fetch Data Parsing**) contains VIs that you can use to retrieve data directly from the SQL operation into G data types. This can result in faster retrieval and reduced memory usage.
- **Database Browser dialog for interactive testing**—The Database Browser dialog, accessible from the **Project** menu, makes it easy to examine databases. You can easily browse through the databases, viewing the tables that make up each database and the attributes of those tables. You can execute SQL against those databases, and an easy-to-use dialog helps you construct database queries.
- **Programmatically retrieve information about data sources, databases, and tables**—The Database Browser utility was written using the G SQL Toolkit VIs. All of the tools that it uses to mine information about databases are available to you as VIs you can use in your own application. This can be useful if you want to present a simple browser interface for selecting a database or table.
- **You can control access to databases when making critical modifications**—The VIs in the Transaction Locking palette (**Functions»SQL»Transactions»Transaction Locking**) allow you to control which kinds of operations can take place when you are performing SQL operations. In the lowest level, locks are obtained only when you modify data, and those locks are maintained until you complete the transaction. Higher levels let you also obtain locks when reading data. This is useful if you are making multiple reads as part of a transaction, and you want to avoid having the database changed by someone else in the middle of your transaction.

- **You can access the SQL Toolkit VIs more easily**—The palettes have been reorganized and are now directly accessible from the top level of the **Functions** palette. Also, a number of the VIs' icons have been changed to be consistent with LabVIEW and BridgeVIEW icons.
- **New examples show off dynamic SQL and SQL using the Cluster Templates**—The examples directory, now located in **Examples»SQL**, includes two new libraries: `insert.llb` and `query.llb`. These LLBs contain VIs that show how to use Dynamic SQL and SQL using the Cluster Templates. They also include benchmark VIs that time these operations so you can get a better idea about how the different options compare.

Compatibility Issues when Upgrading from Version 1.14 or Earlier

In addition to adding new functionality, there are a few changes that result in compatibility issues. In most cases, you either relink to a SubVI or popup and replace a call to a VI that has been renamed.

The following is a description of the changes:

- **Connector pane changes**—In addition to returning an error cluster, all of the SQL VIs previously returned a copy of the error number and the error text. The error number and error text outputs were removed to simplify the VIs, make them more consistent with other VIs that have error inputs and outputs, and to make room for additional inputs and outputs.

Also, when you start an SQL operation using the Execute SQL VI, you previously got back an SQL reference that was a numeric just like the connection reference that you passed into Execute SQL. Consequently, you might get the connection reference and the SQL reference mixed up when wiring subsequent VIs. Now, the Execute SQL VI returns a cluster that contains both the connection and the SQL reference. All VIs that previously took an SQL reference now take this cluster as an input. Also, previously most VIs took a reference number as an input, but did not duplicate this as an output. All SQL VIs are modified to produce a copy of the input reference number/cluster. This should simplify wiring, but it is a complication in converting your VIs to the new version.

When you load your VIs, any calls to the SQL VIs are broken because of the changed connector panes. To modify your VIs, popup on each call to an SQL VI and select **Relink to SubVI** from the popup menu. This replaces the old instance with the new instance and attempts to reroute wires to the correct inputs and outputs. In many instances if you didn't have the error text and error number wired up before, the correct connections are made automatically. In a few cases, you must delete a resulting bad wire and connect it to the correct input or output.

- **VI filename changes**—The VIs for converting dates to and from SQL formats were renamed to remove “LabVIEW” because this toolkit and all VIs in it apply both to LabVIEW and to BridgeVIEW. These VIs are located in **Functions»SQL»Advanced SQL»Date Conversion**. The following is a list of the old and new names:
 - LabVIEW Date to SQL Date Format VI changed to G Date to SQL Date Format VI.
 - SQL Date to LabVIEW Date Format VI changed to SQL Date to G Date Format VI.

Bug Fixes in Version 2.0

- The Version 1.14 installer did not do version checking correctly, so it could overwrite newer versions of the ODBC Administrator files that may have been installed on your system. If this happened, reinstall your database software to reinstall a new version of the ODBC drivers (for example Microsoft Access).
- Slightly newer versions of the Intersolv database drivers are included. This corrected a problem with Microsoft Access 97, and might correct other database communication problems.
- Windows 3.1 only—Version 1.14 introduced a bug correction for Oracle related to a crash that could occur when connecting to an Oracle data source. This modification involved a new DLL called `qelibfix.dll` in the `Windows\System` directory. While this DLL seems to correct the problem with Oracle, it has been discovered since then that this DLL introduced a problem with the Connect if you specify TRUE for Show Dialog. In Version 2.0, the `qelibfix.dll` is not installed by default, so the Show Dialog option for Connect should now work correctly. See the Appendix B, *Customer Communication*, in the *SQL Toolkit for G Reference Manual*, for more information about how to contact National Instruments.
- If the error cluster input to the Disconnect VI had an error, the specified connection was not disconnected. This has been corrected in Version 2.0.

Upgrading from Version 1.13

In addition to the new features and bug corrections that were made between versions 1.14 and 2.0, there were some additional changes made between versions 1.13 and 1.14.

In Version 1.14, two new VIs were added for Windows 3.1. Some drivers under Windows 3.1 are limited in terms of the size of the SQL statement you can execute (some cannot handle more than 20k of data). This can be a problem if you need to insert a large amount of data as part of an `INSERT` statement, for example. You can use the new `Set SQL VI` and `Append SQL VI` to work around this limitation. To use these VIs, break up your statement into smaller strings. Pass the initial part of your statement to the `Set SQL VI`. Make as many additional calls as are needed to the `Append SQL VI` to complete your statement. Then call `Execute SQL` to execute the statement.

In addition, Version 1.14 included new Intersolv drivers. Also, a problem with Oracle that caused a crash was corrected, although this correction was later modified in 2.0 as described in the previous section.

Upgrading from Version 1.0

Every effort has been made to make the process of upgrading as simple as possible. The names of all the VIs remains the same. However, some VIs created with Version 1.0 need modification when upgraded to The SQL Toolkit for G Version 1.1. VIs that use `connect.vi` and or `execute sql.vi` need modification because of the automatic reference number generation in the new version. Whereas these VIs require the user to assign a reference number to be input to them, the new VIs return a reference number as an output. You have to break reference number inputs, and wire the output reference number to other associated SQL Toolkit for G VIs.

The new error cluster input and output for each SQL Toolkit VI replaces the previous **Start** input and **Done** output Booleans. If you used these Booleans for dataflow programming, the upgraded VIs should replace the Boolean wire with a cluster wire and there is no need for rewiring. If, however, you wired a Boolean to the **Start** terminal, or wired the **Done** terminal to a Boolean object, you must rewire your diagram to correct the resultant broken wire.

To retain compatibility for existing Data Source Names, the `ODBC.ini` file retains the names and references of extant DSNs. Version 1.0 driver DLLs also are retained in your `Windows\System` directory. Other driver DLLs (the ODBC drivers) from Version 1.0 are replaced with newer versions. If you would like to use the new database drivers with existing applications,

you must use the ODBC Administrator to delete the existing Data Source Name definition and add an identically named data source that uses the new database driver.



321531C-01

Feb98