Trafodion Client Installation Guide

Contents

About This Document

This manual describes how to install and configure client applications that enable you to connect to and use a Trafodion database.

Intended Audience

This manual is intended for users who want to connect to and use a Trafodion database.

New and Changed Information in This Edition

This manual shows updated versions for Trafodion Release 1.3.0. It also provides instructions on how to download and install Trafodion odb, a new multi-threaded, ODBC-based command-line tool for parallel data loading and extracting. See "Installing odb" (page 36).

Notation Conventions

General Syntax Notation

This list summarizes the notation conventions for syntax presentation in this manual.

UPPERCASE LETTERS

Uppercase letters indicate keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

SELECT

Italic Letters

Italic letters, regardless of font, indicate variable items that you supply. Items not enclosed in brackets are required. For example:

file-name

Computer Type

Computer type letters within text indicate case-sensitive keywords and reserved words. Type these items exactly as shown. Items not enclosed in brackets are required. For example:

myfile.sh

Bold Text

Bold text in an example indicates user input typed at the terminal. For example:

ENTER RUN CODE

?**123**
CODE RECEIVED: 123.00

The user must press the Return key after typing the input.

[ ] Brackets

Brackets enclose optional syntax items. For example:

DATETIME [*start-field* TO] *end-field*

A group of items enclosed in brackets is a list from which you can choose one item or none. The items in the list can be arranged either vertically, with aligned brackets on each side of the list, or horizontally, enclosed in a pair of brackets and separated by vertical lines. For example:

DROP SCHEMA *schema* [CASCADE]
                    [RESTRICT]

```
DROP SCHEMA schema [ CASCADE | RESTRICT ]
```

{ } Braces

Braces enclose required syntax items. For example:

```
FROM { grantee[, grantee]...}
```

A group of items enclosed in braces is a list from which you are required to choose one item. The items in the list can be arranged either vertically, with aligned braces on each side of the list, or horizontally, enclosed in a pair of braces and separated by vertical lines. For example:

```
INTERVAL { start-field TO end-field }
        { single-field }
```

```
INTERVAL { start-field TO end-field  | single-field }
```

| Vertical Line

A vertical line separates alternatives in a horizontal list that is enclosed in brackets or braces. For example:

```
{expression | NULL}
```

… Ellipsis

An ellipsis immediately following a pair of brackets or braces indicates that you can repeat the enclosed sequence of syntax items any number of times. For example:

```
ATTRIBUTE[S] attribute [, attribute]...
```

```
{, sql-expression}...
```

An ellipsis immediately following a single syntax item indicates that you can repeat that syntax item any number of times. For example:

```
expression-n…
```

Punctuation

Parentheses, commas, semicolons, and other symbols not previously described must be typed as shown. For example:

```
DAY (datetime-expression)
```

```
@script-file
```

Quotation marks around a symbol such as a bracket or brace indicate the symbol is a required character that you must type as shown. For example:

```
"{" module-name [, module-name]... "}"
```

Item Spacing

Spaces shown between items are required unless one of the items is a punctuation symbol such as a parenthesis or a comma. For example:

```
DAY (datetime-expression)
```

```
DAY(datetime-expression)
```

If there is no space between two items, spaces are not permitted. In this example, no spaces are permitted between the period and any other items:

```
myfile.sh
```

Line Spacing

If the syntax of a command is too long to fit on a single line, each continuation line is indented three spaces and is separated from the preceding line by a blank line. This spacing distinguishes items in a continuation line from items in a vertical list of selections. For example:

```
match-value [NOT] LIKE pattern

   [ESCAPE esc-char-expression]
```

Publishing History

Product Version	Publication Date
Trafodion Release 1.3.0	November 2015

We Encourage Your Comments

The Trafodion community encourages your comments concerning this document. We are committed to providing documentation that meets your needs. Send any errors found, suggestions for improvement, or compliments to:

issues@trafodion.incubator.apache.org

Include the document title and any comment, error found, or suggestion for improvement you have concerning this document.

1 Introduction to the Client Products

This manual describes how to install and configure the following client applications, which enable you to connect to and use a Trafodion database.

Client Name	Description
Trafodion ODBC Driver for Windows	A driver that enables applications, which were developed for the Microsoft Open Database Connectivity (ODBC) application programming interface (API) and which run on a Windows workstation, to connect to a Trafodion database.
Trafodion ODBC Driver for Linux	A driver that enables applications, which were developed for the Microsoft ODBC API and run on a Linux workstation, to connect to a Trafodion database.
Trafodion JDBC Type 4 Driver	A driver that enables Java applications that run on a client workstation to connect to a Trafodion database. NOTE: The Trafodion Command Interface (TrafCI), DbVisualizer, and SQuirreL SQL Client require this driver to be installed on the client workstation.
Trafodion Command Interface (TrafCI)	A command-line interface that allows you to connect to a Trafodion database and run SQL statements and other commands interactively or from script files. For more information, see the *Trafodion Command Interface Guide*.
Trafodion odb tool	A multi-threaded, ODBC-based command-line tool for parallel data loading and extracting. For more information, see the *Trafodion odb User Guide*.
DbVisualizer	A third-party database tool that allows you to connect to a Trafodion database. For more information, see the DbVisualizer website.
SQuirreL SQL Client	A third-party database tool that allows you to connect to a Trafodion database. For more information, see the SQuirreL SQL Client website.

2 Installing the Trafodion ODBC Driver for Windows

These installation instructions apply to Trafodion Release 1.1.n and subsequent updates of the product until otherwise indicated.

- Installation Package
- Installation Requirements
- Installation Instructions

Installation Package

The installer file for the Trafodion ODBC Driver for Windows is packaged within the Trafodion clients tar file, `clients-1.1.`n`.tar.gz`, on the Trafodion downloads site (http://downloads.trafodion.org/). Use this installer file to install the Trafodion ODBC Driver for Windows:

Installer File	Description	Client Operating System
`TFODBC64-1.1.`n`.msi`	Windows MSI installer file	Windows 64-bit

By default, a new version of the ODBC driver will be installed in this directory and its folders unless you specify a different directory during installation:

Default Installation Directory	Client Operating System
`C:\Program Files\Trafodion\TRAF ODBC 1.0`	Windows 64-bit

The ODBC client installation file installs or links to multiple client components:

This client component...	Does this...
Microsoft ODBC Driver Manager	Manages access to ODBC drivers for applications. The driver manager loads and unloads drivers and passes calls for ODBC functions to the correct driver.
Trafodion ODBC driver	Implements ODBC function calls to enable an ODBC client application to access the Trafodion database.
Microsoft ODBC Administrator	Adds, configures, and removes ODBC data sources on client workstations.

Installation Requirements

Computer	Windows compatible PC workstation
Memory	Recommended minimum 32 MB
Disk space	Minimum 30 MB additional free space
Operating system	x64 Edition of Microsoft Windows 7, Windows 8, or Windows Server 2008
Network software	TCP/IP

Installation Instructions

Installing the Trafodion ODBC Driver for Windows

NOTE: To install the driver on your PC, you must be logged on with a user ID that has administrator privileges.

Task 1: Download the ODBC client installation file

1. Create a download folder on the client workstation.
2. Open a Web browser and navigate to the Trafodion downloads site (http:// downloads.trafodion.org/).
3. Click on the clients-1.1.*n*.tar.gz link to start downloading the Trafodion clients tar file to your workstation.
4. Place the `clients-1.1.n.tar.gz` file into the download folder.
5. Extract the contents of the `clients-1.1.n.tar.gz` file by using an extract tool like WinZip.

   The package file contains the `TFODBC64-1.1.n.msi` installer file, which is extracted to the `clients` subdirectory.

Task 2: Start the InstallShield wizard

The InstallShield wizard walks you through the steps to install the client components (Trafodion ODBC 1.0) on your workstation. You can perform the installation in *interactive mode*, in which you provide input or accept defaults when prompted as ODBC is installed. See "Interactive Mode Installation" (page 10).

Interactive Mode Installation

1. Double-click the `.msi` distribution file to start the InstallShield wizard.
2. On the Welcome page, click Next.
3. On the Destination Folder page, click Next to select the default location:

   `C:\Program Files\Trafodion\TRAF ODBC 1.0\`

   This location is the installation directory for ODBC header and help files. All other ODBC files are installed in %SYSTEMROOT%\system32.

4. On the Set Certificate Directory page, click Next to accept the default location for the security certificate (SYSTEM_DEFAULT).
5. On the Ready to Install the Program page, click Install. The installation begins.

   The Installing Trafodion ODBC64 1.0 page displays a progress bar.

6. On the InstallShield Wizard Completed page, click Finish to exit the wizard. Do not add a new Trafodion ODBC data source to the MS ODBC Administrator until you have restarted your system.

Task 3: Add a client data source

1. Start the Microsoft ODBC Administrator:

   - On Windows 7: Start > All Programs > Trafodion ODBC 1.0 > MS ODBC Administrator
   - On Windows 8: Right-click the Trafodion ODBC 1.0 icon on the desktop and select MS ODBC Administrator.

2. In the ODBC Data Source Administrator dialog box, click Add.
3. Select TRAF ODBC 1.0, and then click Finish to start the Create a New Trafodion ODBC Data Source wizard.
4. Enter the data source name (for example, Default_DataSource_Schema1) and an optional description, and click Next.
5. Enter the IP address or host name for the database platform. Enter the port number as 37800. Leave the defaults as is, and click Next.
6. Enter the schema name. The default schema name is SEABASE. Click Next.

7. Enter the translate DLL name and its option, if you have one. If not, leave it blank. Leave the localization defaults as is.

   The Replacement Character replaces any character that is incompatible for translation when retrieving data. It is one character (one or two bytes long). The Replacement Character is assumed to be in the character set specified in the Client/Server Character Set Interaction. If it is not specified, '?' is used as the default.

   Click Next.

8. Select the trace options, if desired. Change the Log File Path to the location you want, or leave the default as is, and click Finish.

9. The wizard gives you an opportunity to test the connection. Click Test Connection and click OK.

10. The server ID and schema are filled in for you. Enter a valid user name and password, and click OK.

    The wizard attempts to connect to the data source and displays a message stating whether it was successful or not.

11. Click OK to save the data source, or click Cancel twice to quit the Create Data Source wizard.

Reinstalling the Trafodion ODBC Driver for Windows

To reinstall the driver, we recommend that you fully remove your ODBC driver and then install the new version. See "Uninstalling the Trafodion ODBC Driver for Windows" (page 11) and then "Installing the Trafodion ODBC Driver for Windows" (page 9).

Uninstalling the Trafodion ODBC Driver for Windows

1. Start to remove the ODBC driver:
   - On Windows 7: Start > All Programs > Trafodion ODBC 1.0 > Remove TRAF ODBC 1.0
   - On Windows 8: Right-click the Trafodion ODBC 1.0 icon on the desktop and select Remove TRAF ODBC 1.0.

2. When the Windows Installer dialog box asks you if you want to uninstall this product, click Yes.

3. The Trafodion ODBC 1.0 dialog box displays the status and asks you to wait while Windows configures Trafodion ODBC 1.0 (that is, removes the Trafodion ODBC Driver from your Windows workstation).

   After this dialog box disappears, Trafodion ODBC 1.0 is no longer on your workstation.

NOTE: Uninstalling the ODBC driver does not remove pre-existing data source definitions from the Windows registry.

Setting Up the Client Environment

All client data sources connect to the preconfigured server data source on the database platform, which is Default_DataSource. You can configure only one data source, Default_DataSource, on the database platform, but you can create other data source definitions on the workstation. For example, if you have more than one schema on the database platform and you want to connect to each of those schemas on the database platform, you can create a client data source for each of those schemas. Instead of changing the schema definition in the data source definition on the workstation, you can create multiple data source definitions with different schemas on the workstation. The client data source will use the specified schema but will connect to Default_DataSource on the database platform.

To create a data source on the client workstation, follow these steps:

1. Launch the MS ODBC Administrator. For example, on Windows 7, select Start→All Programs→Trafodion ODBC 1.0→MS ODBC Administrator.
2. In the ODBC Data Source Administrator dialog box, select the User DSN tab, and click Add.
3. Select the TRAF ODBC 1.0 driver, and then click Finish.

   A new dialog box appears, prompting you to create a new data source.
4. Enter the name of the data source, `Default_DataSource`, and click Next to continue.
5. Enter the IP address and port number of the Trafodion system to which will be connecting. The port number must be 37800. Click Next to continue.
6. Select the default schema. If you do not select a schema, the default is USR. Click Next to continue.
7. If desired, configure the translate dll, which translates data from one character set to another, and configure the localization. By default, the client error message language is English, and the client's local character set is used. Click Next to continue.
8. If desired, set the trace options and the file path of the trace log. Click Finish.

   The Test Trafodion ODBC Connection dialog box appears, allowing you to test the connection using the data source that you created.
9. Click Test Connection.
10. When prompted, enter your username and password, and click OK.

    If the connection is successful, you will see `Connected Successfully` in the Test Trafodion ODBC Connection dialog box.

Enabling Compression

When compression is enabled in the ODBC driver, the ODBC driver can send and receive large volumes of data quickly and efficiently to and from the Trafodion Database Connectivity Services (DCS) server over a TCP/IP network. By default, compression is disabled.

To enable compression in the ODBC driver or to change the compression setting, follow these steps:

1. Launch the MS ODBC Administrator. For example, on Windows 7, select Start→All Programs→Trafodion ODBC 1.0→MS ODBC Administrator.
2. In the ODBC Data Source Administrator dialog box, select the User DSN tab, select the name of your data source under User Data Sources, and click Configure. If you did not create a data source, see "Setting Up the Client Environment" (page 11).

   A new dialog box appears, showing the configuration of your data source.
3. Select the Network tab, and then select one of these values for Compression:

   - SYSTEM_DEFAULT, which is the same as no compression
   - no compression
   - best speed
   - best compression
   - balance
   - An integer from 0 to 9, with 0 being no compression and 9 being the maximum available compression
4. Click OK to accept the change.
5. Click OK to exit the ODBC Data Source Administrator dialog box.

Running the Sample Program (basicsql)

To build and run the executable file, follow these steps:

1. Open a Visual Studio x64 Win64 Command Prompt. Make sure to select the x64 version of the command prompt. For example, on Windows 7, select Start→All Programs→Microsoft Visual Studio 2010→Visual Studio Tools→Visual Studio x64 Win64 Command Prompt.

2. At the command prompt, move to the directory where you put the `basicsql.cpp` and build files.

3. Run `build` at the command prompt. You will see `basicsql.exe` created in the same directory as the source file.

4. Before running the sample program, create a Trafodion data source named `Default_DataSource` on the client workstation using MS ODBC Administrator. For instructions, see "Setting Up the Client Environment" (page 11).

5. From the command prompt, run the sample program by entering either `run` or this command:

    ```
    basicsql Default_DataSource username password
    ```

    If the sample program executes successfully, you should see this output:

    ```
    Using Connect String: DSN=Default_DataSource;UID=user1;PWD=pwd1;
    Successfully connected using SQLDriverConnect.
    Drop sample table if it exists...
    Creating sample table TASKS...
    Table TASKS created using SQLExecDirect.
    Inserting data using SQLBindParameter, SQLPrepare, SQLExecute
    Data inserted.
    Fetching data using SQLExecDirect, SQLFetch, SQLGetData
    Data selected: 1000 CREATE REPORTS      2014-3-22
    Basic SQL ODBC Test Passed!
    ```

3 Installing the Trafodion ODBC Driver for Linux

These installation instructions apply to Trafodion Release 1.1.*n* and subsequent updates of the product until otherwise indicated.

- Installation Package
- Installation Requirements
- Installation Instructions

Installation Package

The distribution file for the Trafodion ODBC Driver for Linux is packaged within the Trafodion clients tar file, `clients-1.1.n.tar.gz`, on the Trafodion downloads site (http:// downloads.trafodion.org/). Use this distribution file to install the Trafodion ODBC Driver for Linux:

Distribution File	File Contents	Client Operating System
TRAF_ODBC_Linux_Driver_64.tar.gz	connect_test.cpp install.sh libicudataNv44.so.44 libicuucNv44.so.44 libtrafodbc_l64.so libtrafodbc_l64_drvr.so LICENSE license.txt MD5SUM TRAFDSN	x86_64 Linux

By default, a new version of the Trafodion ODBC driver will be installed in the following directories unless you specify a different directory during installation:

- `/usr/lib64`
- `/etc/odbc`

NOTE: The following header files are not packaged with the Trafodion ODBC driver:

- `sql.h`
- `sqlext.h`
- `sqltypes.h`
- `sqlucode.h`

To install those header files, see "Setting Up the Client Environment" (page 15).

Installation Requirements

The driver for Linux requires libgcc 3.4.3 and libstd++ 6.0.

If you are building ODBC applications, please use the preferred build platform, RedHat 6.4 or CentOS 6.4.

Installation Instructions

Downloading the Trafodion ODBC Driver for Linux

1. Create a download folder on the client workstation.

2. Open a Web browser and navigate to the Trafodion downloads site (http://downloads.trafodion.org/).

3. Click on the clients-1.1.*n*.tar.gz link to start downloading the Trafodion clients tar file to your workstation.

4. Place the `clients-1.1.n.tar.gz` file into the download folder.

5. Unpack the `clients-1.1.n.tar.gz` file. For example:

   `tar -xzf clients-1.1.0.tar.gz`

   The package file contains the Trafodion ODBC distribution file, `TRAF_ODBC_Linux_Driver_64.tar.gz`, which is extracted to the `clients` subdirectory.

Installing or Reinstalling the Trafodion ODBC Driver for Linux

NOTE: You must have root access to install the Trafodion ODBC Driver for Linux at the default system location.

1. Change the directory to the `clients` subdirectory, and decompress the `.tar.gz` distribution file:

   `gunzip TRAF_ODBC_Linux_Driver_64.tar.gz`

2. Extract the contents of the `.tar` file. A directory called `PkgTmp` is created.

   `tar -xvf TRAF_ODBC_Linux_Driver_64.tar`

3. Install the product by entering these commands:

   ```
   cd PkgTmp
   ./install.sh
   ```

   Except for the sample file, the `install.sh` script saves a copy (`.SAV`) of your previous installation files if they exist.

4. Accept the terms of the license agreement by entering **yes**.

5. Enter a directory for the library files, or press Enter to use the default directory (`/usr/lib64`).

6. Enter a directory for the datasource template file, or press Enter to use the default directory (`/etc/odbc`).

7. Enter a directory for the sample program, or press Enter to use the default directory (`/etc/odbc`).

Setting Up the Client Environment

If you selected default options during installation, ensure that:

- The libraries are located in the `/usr/lib64` directory.
- A TRAFDSN file is in the `/etc/odbc` directory.

If you select non-default locations during installation, ensure that the files are installed in the directories that you specified during installation.

The driver expects the TRAFDSN file to be present in either the default location (`/etc/odbc`) or the current working directory (CWD) of the application.

If you are building ODBC applications, you will need to install these header files in your build environment:

- `sql.h`
- `sqlext.h`
- `sqltypes.h`
- `sqlucode.h`

To install those header files from the latest packages, run this yum command:

```
yum install libiodbc libiodbc-devel
```

The yum command automatically installs the header files in the `/usr/include` and `/usr/include/libiodbc` directories.

Enabling Compression

When compression is enabled in the ODBC driver, the ODBC driver can send and receive large volumes of data quickly and efficiently to and from the Trafodion Database Connectivity Services (DCS) server over a TCP/IP network. By default, compression is disabled.

To enable compression in the ODBC driver or to change the compression setting, follow these steps:

- If you are using the Trafodion ODBC driver manager, add `Compression = compression-level` to the DSN section of TRAFDSN file.

- If you are using a third-party driver manager, such as unixODBC, add `Compression = compression-level` to the DSN section of the `odbc.ini` file.

The `compression-level` is one of these values:

- `SYSTEM_DEFAULT`, which is the same as no compression

- `no compression`

- `best speed`

- `best compression`

- `balance`

- An integer from 0 to 9, with 0 being no compression and 9 being the maximum available compression

Using a Third-Party Driver Manager

NOTE: For better performance, we recommend that you use at least version 2.3.x of unixODBC.

- If you are using an external driver manager, you must point to `libtrafodbc_drvr64.so` and not to `libtrafodbc64.so`.

- The driver, `libtrafodbc_l64_drvr.so`, has been verified with iODBC and unixODBC driver managers.

- These driver managers, as well as documentation, can be found at these Web sites:

  http://www.iodbc.org/

  http://www.unixodbc.org/

- For information on the necessary data-source configuration options, you will need to add to the respective configuration files (for example, to `odbc.ini`).

Running the Sample Program (connect_test)

NOTE: The examples after each step assume that you have default installation directories.

If you have a previous version of the Trafodion ODBC driver installed, you need to re-link your existing application to ensure that you pick up the correct version of the driver. If you are unsure of the version, check the version of your application with this command:

```
ldd object-file
```

1. Move to the directory where you installed the sample program:

   ```
   cd /etc/odbc
   ```

2. Set the environment variable LD_LIBRARY_PATH:

```
export LD_LIBRARY_PATH=<path-to-odbc-library-files or /usr/lib64>
```

3. In the `/etc/odbc/TRAFDSN` file, add the correct IP address to the `Server` parameter for the `Default_DataSource`. For example:

```
[Default_DataSource]
Description                  = Default Data Source
Catalog                      = TRAFODION
Schema                       = SEABASE
DataLang                     = 0
FetchBufferSize              = SYSTEM_DEFAULT
Server                       = TCP:1.2.3.4:37800
SQL_ATTR_CONNECTION_TIMEOUT  = SYSTEM_DEFAULT
SQL_LOGIN_TIMEOUT            = SYSTEM_DEFAULT
SQL_QUERY_TIMEOUT            = NO_TIMEOUT
```

4. Compile the sample program.

```
g++ -g connect_test.cpp -L/usr/lib64 -I/usr/include/odbc -ltrafodbc64 -o connect_test
```

5. Run the sample program:

```
./connect_test -d Default_DataSource -u username -p password
```

If the sample program runs successfully, you should see output similar to the following:

```
Using Connect String: DSN=Default_DataSource;UID=username;PWD=****;

Connect Test Passed...
```

Running the Sample Program (basicsql)

NOTE: The Basic SQL sample program is not currently bundled with the ODBC Linux driver. To obtain the source code for this program, see the "Source Code for the Sample Program— basicsql.cpp" (page 43).

If you have a previous version of the Trafodion ODBC driver installed, you need to re-link your existing application to ensure that you pick up the correct version of the driver. If you are unsure of the version, check the version of your application with this command:

```
ldd object-file
```

1. Move to the directory where you put the `basicsql.cpp` file.
2. Set the environment variable LD_LIBRARY_PATH:

```
export LD_LIBRARY_PATH=<path-to-odbc-driver-dlls>
```

3. In the `/etc/odbc/TRAFDSN` file, add the correct IP address to the `Server` parameter for the `Default_DataSource`. For example:

```
[Default_DataSource]
Description                  = Default Data Source
Catalog                      = TRAFODION
Schema                       = SEABASE
DataLang                     = 0
FetchBufferSize              = SYSTEM_DEFAULT
Server                       = TCP:1.2.3.4:37800
SQL_ATTR_CONNECTION_TIMEOUT  = SYSTEM_DEFAULT
SQL_LOGIN_TIMEOUT            = SYSTEM_DEFAULT
SQL_QUERY_TIMEOUT            = NO_TIMEOUT
```

4. Compile the sample program.

```
g++ -g basicsql.cpp -L. -I. -ltrafodbc64 -o basicsql
```

5. Run the sample program:

```
basicsql Default_DataSource username password
```

If the sample program runs successfully, you should see output similar to the following:

```
Using Connect String: DSN=Default_DataSource;UID=user1;PWD=pwd1;
Successfully connected using SQLDriverConnect.
Drop sample table if it exists...
Creating sample table TASKS...
Table TASKS created using SQLExecDirect.
Inserting data using SQLBindParameter, SQLPrepare, SQLExecute
Data inserted.
Fetching data using SQLExecDirect, SQLFetch, SQLGetData
Data selected: 1000 CREATE REPORTS      2014-3-22
Basic SQL ODBC Test Passed!
```

4 Installing the Trafodion JDBC Type 4 Driver

These installation instructions apply to Trafodion Release 1.1.n and subsequent updates of the product until otherwise indicated.

- Installation Package
- Installation Requirements
- Installation Instructions

Installation Package

The distribution file for the Trafodion JDBC Type 4 Driver is packaged within the Trafodion clients tar file, `clients-1.1.`n`.tar.gz`, on the Trafodion downloads site (http://downloads.trafodion.org/). Use this distribution file to install the Trafodion JDBC Type 4 Driver:

Distribution File	Description	Client Operating System
JDBCT4.zip	Zip file	Windows or Linux

By default, a new version of the JDBC Type 4 driver will be installed in this directory and its folders unless you specify a different directory during installation:

Default Installation Directory	Client Operating System
<user-specified directory> (For example: `/usr/local/jdbc`)	Windows or Linux

Installation Folder	Files	Description
`/lib`	`jdbcT4.jar`	Product JAR file
`/samples`	`t4jdbc.properties`	Properties file that you can configure for your application environment
	`README`	Readme file that explains how to use the common sample set
`/samples/common`	`sampleUtils.java`	Sample source code for creating, populating, and dropping sample tables
`/samples/DBMetaSample`	`DBMetaSample.java`	Sample source code for getting metadata about the sample tables
	`README`	Readme file that explains how to use this sample set
`/samples/PreparedStatementSample`	`PreparedStatementSample.java`	Sample code for simple or parameterized SELECT statements that are prepared
	`README`	Readme file that explains how to use this sample set
`/samples/ResultSetSample`	`README`	Readme file that explains how to use this sample set
	`ResultSetSample.java`	Sample source code for fetching rows from a result set

Installation Folder	Files	Description
/samples/StatementSample	README	Readme file that explains how to use this sample set
	StatementSample.java	Sample source code for fetching rows from a simple SELECT statement

Installation Requirements

The Trafodion JDBC Type 4 Driver requires a Java-enabled platform that supports the Java Development Kit (JDK) 1.7 or higher.

To determine the Java version that you have installed on your client workstation and to install a supported version, see "Installing and Verifying the Java Runtime Environment (JRE)" (page 20).

Installing and Verifying the Java Runtime Environment (JRE)

The Trafodion JDBC Type 4 Driver requires that a compatible Java version be installed on the client workstation and that the Java path be set to the correct location. The supported Java version is 1.7 or higher.

Verifying the Java Version

To display the Java version of the client workstation on the screen, enter:

```
java -version
```

For example:

```
C:\>java -version
java version "1.7.0_45"
Java(TM) SE Runtime Environment (build 1.7.0_45-b18)
Java HotSpot(TM) Client VM (build 24.45-b08, mixed mode, sharing)

C:\>
```

The Java version should be 1.7 or higher. If the returned version is not supported, see "Installing a Supported Java Version" (page 20).

If you see this message:

```
'java' is not recognized as an internal or external command, operable program or batch file.
```

It indicates that the Java path is not set. Follow one of these sets of instructions, depending on the operating system of your client workstation:

- "Setting the PATH to a Supported Java Version on Windows" (page 21)
- "Setting the PATH to a Supported Java Version on Linux" (page 22)

Installing a Supported Java Version

The supported Java version is 1.7 or higher. To install one of the supported Java versions on the client workstation, go to this link:

http://www.java.com/en/download

After installing the Java version, proceed with setting the Java path. Follow one of these sets of instructions, depending on the operating system of your client workstation:
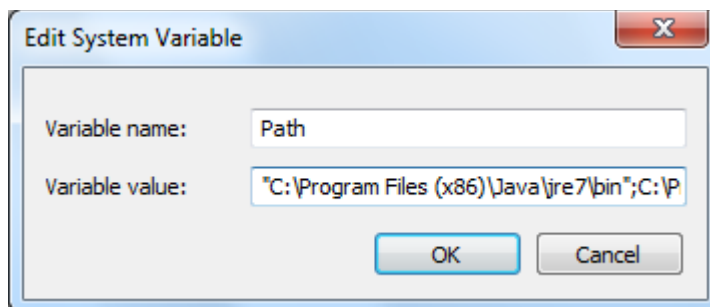
- "Setting the PATH to a Supported Java Version on Windows" (page 21)
- "Setting the PATH to a Supported Java Version on Linux" (page 22)

Setting the PATH to a Supported Java Version on Windows

1. Right-click the Computer icon on your desktop, and then select Properties.

   The Control Panel > System and Security > System window appears.

2. In the left navigation bar, click the Advanced system settings link.

3. In the System Properties dialog box, click the Environment Variables button.

4. Under System variables, select the variable named Path, and then click Edit:
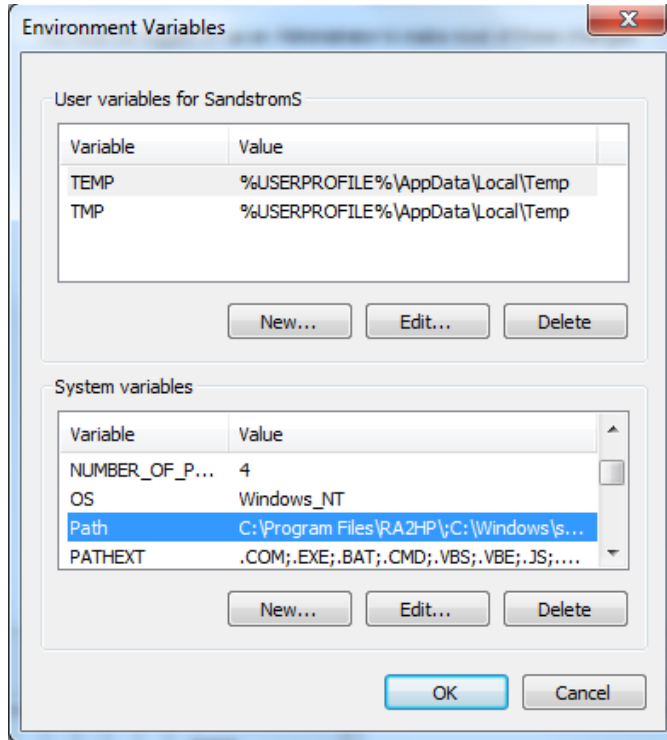
5. Place the cursor at the beginning of the Variable value field and enter the path of the Java `bin` directory, ending with a semicolon (;):

For example:

```
"C:\Program Files (x86)\Java\jre7\bin";
```

NOTE: Check that no space exists after the semicolon (;) in the path. If there are spaces in the directory name, delimit the entire directory path in double quotes (") before the semicolon.

6. Click OK.

7. Verify that the updated Path appears under System variables, and click OK.

8. In the System Properties dialog box, click OK to accept the changes.

Setting the PATH to a Supported Java Version on Linux

1. Open the user profile (`.profile` or `.bash_profile` for the Bash shell) in the `/home` directory. For example:

   ```
   vi .profile
   ```

2. In the user profile, set the PATH environment variable to include the path of the Java `bin` directory. For example:

   ```
   export PATH=/opt/java1.7/jre/bin:$PATH
   ```

   NOTE: Place the path of the Java `bin` directory before $PATH, and check that no space exists after the colon (:) in the path. In the C shell, use the `setenv` command instead of `export`.

3. To activate the changes, either log out and log in again or execute the user profile. For example:

   ```
   . .profile
   ```

Installation Instructions

Downloading the Trafodion JDBC Type 4 Driver

1. Create a download folder on the client workstation.
2. Open a Web browser and navigate to the Trafodion downloads site (http:// downloads.trafodion.org/).
3. Click on the clients-1.1.*n*.tar.gz link to start downloading the Trafodion clients tar file to your workstation.
4. Place the `clients-1.1.`*n*`.tar.gz` file into the download folder.
5. Unpack the `clients-1.1.`*n*`.tar.gz` file. For example:

   ```
   tar -xzf clients-1.1.0.tar.gz
   ```

   The package file contains the `JDBCT4.zip` distribution file, which is extracted to the `clients` subdirectory.

6. Proceed with "Installing the Trafodion JDBC Type 4 Driver From the Zip File" (page 22).

Installing the Trafodion JDBC Type 4 Driver From the Zip File

1. Change the directory to the `clients` subdirectory.
2. Extract the contents of the `JDBCT4.zip` file by using the unzip command (or the WinZip extract tool):

   ```
   unzip JDBCT4.zip
   ```

   The command creates these subdirectories:

   - `lib`, which contains the `jdbcT4.jar file`
   - `samples`, which contains sample programs

3. Set up the client environment. See Setting Up the Client Environment.

Uninstalling the Trafodion JDBC Type 4 Driver

Run one of these sets of commands to remove the Trafodion JDBC Type 4 Driver:

- On Linux:

  ```
  rm -rf jdbc-installation-directory
  ```

  For example:

```
rm -rf /usr/local/jdbc
```

- On Windows:

```
del jdbc-installation-directory
rmdir jdbc-installation-directory
```

For example:

```
C:\>del /s JDBC
C:\JDBC\*, Are you sure (Y/N)? Y
C:\JDBC\install\*, Are you sure (Y/N)? Y
Deleted file - C:\JDBC\install\t4jdbcSanityCheck.class
Deleted file - C:\JDBC\install\t4jdbcUninstall.class
Deleted file - C:\JDBC\install\product.contents
C:\JDBC\lib\*, Are you sure (Y/N)? Y
Deleted file - C:\JDBC\lib\jdbcT4.jar
C:\JDBC\samples\*, Are you sure (Y/N)? Y
Deleted file - C:\JDBC\samples\t4jdbc.properties
Deleted file - C:\JDBC\samples\README
C:\JDBC\samples\common\*, Are you sure (Y/N)? Y
Deleted file - C:\JDBC\samples\common\sampleUtils.java
C:\JDBC\samples\DBMetaSample\*, Are you sure (Y/N)? Y
Deleted file - C:\JDBC\samples\DBMetaSample\DBMetaSample.java
Deleted file - C:\JDBC\samples\DBMetaSample\README
C:\JDBC\samples\PreparedStatementSample\*, Are you sure (Y/N)? Y
Deleted file - C:\JDBC\samples\PreparedStatementSample\PreparedStatementSample.java
Deleted file - C:\JDBC\samples\PreparedStatementSample\README
C:\JDBC\samples\ResultSetSample\*, Are you sure (Y/N)? Y
Deleted file - C:\JDBC\samples\ResultSetSample\README
Deleted file - C:\JDBC\samples\ResultSetSample\ResultSetSample.java
C:\JDBC\samples\StatementSample\*, Are you sure (Y/N)? Y
Deleted file - C:\JDBC\samples\StatementSample\README
Deleted file - C:\JDBC\samples\StatementSample\StatementSample.java

C:\>rmdir /s JDBC
JDBC, Are you sure (Y/N)? Y

C:\>
```

Reinstalling the Trafodion JDBC Type 4 Driver

1. Close all applications running on the workstation, except the Web browser.
2. In the browser, navigate to the download site and download the client package file. For more information, see "Downloading the Trafodion JDBC Type 4 Driver" (page 22).
3. Extract the contents of the zip file by following these instructions: "Installing the Trafodion JDBC Type 4 Driver From the Zip File" (page 22).
4. Set up the client environment. See Setting Up the Client Environment.

Setting Up the Client Environment

Check that you have a supported Java version installed on the client workstation. The supported Java version is 1.7 or higher. For more information, see "Installing and Verifying the Java Runtime Environment (JRE)" (page 25).

If you plan to write and run Java applications that use the Trafodion JDBC Type 4 Driver to connect to a Trafodion database, set these environment variables on the client workstation, replacing *jdk-directory* with the location of your Java Development Kit and replacing

jdbc-installation-directory with the name of the directory where you downloaded the JDBC Type 4 driver:

Environment Variable	On Windows	On Linux
JAVA_HOME	`set JAVA_HOME=`*jdk-directory*	`export JAVA_HOME=`*jdk-directory*
PATH	`set PATH=%PATH%;%JAVA_HOME%\bin`	`export PATH=$PATH:$JAVA_HOME/bin`
CLASSPATH	`set CLASSPATH=%CLASSPATH%;`*jdbc-installation-directory*`\lib\jdbcT4.jar;`	`export CLASSPATH=$CLASSPATH:`*jdbc-installation-directory*`/lib/jdbcT4.jar:`

Configuring Applications

Edit the `t4jdbc.properties` file. Set these values for your environment:

- catalog: Specify a catalog that exists in the database.
- schema: Specify a schema that exists in the database.
- user: Specify the name of a user who will be accessing the database.
- password: Specify the password of a user who will be accessing the database.
- url: Specify this string:

    `jdbc:t4jdbc://`*host-name*`:`*port-number*`/:`

    host-name is the IP address or host name of the database platform, and *port-number* is the location where the Trafodion Database Connectivity Service (DCS) is running, which is 37800 by default.

Example of an `t4jdbc.properties` file:

```
catalog = your-catalog
schema = your-schema
user = user-name
password = password
url = jdbc:t4jdbc://host-name:port-number/:
```

NOTE: The driver's class name is `org.trafodion.jdbc.t4.T4Driver`.

5 Installing the Trafodion Command Interface (TrafCI)

These installation instructions apply to Trafodion Release 1.1.n and subsequent updates of the product until otherwise indicated.

- "Installation Package" (page 25)
- "Installation Requirements" (page 25)
- "Installation Instructions" (page 26)
- "Post-Installation Instructions" (page 33)

Installation Package

The distribution file for the Trafodion Command Interface (TrafCI) is packaged within the Trafodion clients tar file, `clients-1.1.n.tar.gz`, on the Trafodion downloads site (http://downloads.trafodion.org/). Use this distribution file to install TrafCI:

Distribution File	Description	Client Operating System
trafci.zip	Zip file	Windows or Linux

The distribution zip file contains this installer file: `trafciInstaller.jar`, an executable JAR file.

Installation Requirements

The Trafodion Command Interface (TrafCI) runs on Windows and Linux. Before installing TrafCI on the client workstation, follow these instructions:

- "Installing and Verifying the Java Runtime Environment (JRE)" (page 25)
- "Installing and Verifying the Trafodion JDBC Type 4 Driver" (page 25)
- "Installing Perl or Python" (page 26)

Installing and Verifying the Java Runtime Environment (JRE)

TrafCI and the Trafodion JDBC Type 4 Driver require that a compatible Java version be installed on the client workstation and that the Java path be set to the correct location. The supported Java version is 1.7 or higher.

To determine the Java version that you have installed on your client workstation and to install a supported version, see "Installing and Verifying the Java Runtime Environment (JRE)" (page 20).

Installing and Verifying the Trafodion JDBC Type 4 Driver

TrafCI requires a compatible version of the Trafodion JDBC Type 4 Driver to be installed on the client workstation before you install TrafCI.

Installing the JDBC Driver

To install and use the JDBC driver, see Installing the Trafodion JDBC Type 4 Driver (page 19).

Verifying the JDBC Driver

To display the version of the JDBC driver that is already installed on the client workstation:

1. Change the directory to the `lib` directory, which contains the JDBC driver JAR file:

    - On Windows, enter this command:
      ```
      cd jdbc-installation-directory\lib
      ```
      `jdbc-installation-directory` is the directory where you installed the JDBC driver.

    - On Linux, enter this command:
      ```
      cd jdbc-installation-directory/lib
      ```
      `jdbc-installation-directory` is the directory where you installed the JDBC driver.

2. Enter this command to return version information:
    ```
    java -jar jdbcT4.jar
    ```
    For example:
    ```
    C:\>cd "\<user-specified directory>\lib"

    C:\<user-specified directory>\lib>java -jar jdbcT4.jar
    Traf_JDBC_Type4_Build_40894

    C:\<user-specified directory>\lib>
    ```
    If the JAR file is inaccessible or the returned version is not supported, see "Installing the JDBC Driver" (page 25).

Installing Perl or Python

If you plan to use Perl or Python scripts with TrafCI, verify that you have Perl or Python installed on the client workstation. TrafCI supports these versions of Perl and Python:

- Perl version 5.8.8
- Python version 2.3.4

If you do not have Perl or Python, download it from any open source software provider. You can perform this installation procedure anytime before or after installing TrafCI.

If you plan to run the sample scripts provided with TrafCI, verify that you have the Perl JavaServer and Jython (Java implementation of Python) installed on your client workstation. Use the TrafCI Installation Wizard to automatically download and install both the Perl JavaServer and Jython open source extensions. To download them manually, see the README in the samples directory.

Installation Instructions

NOTE: Before following the installation procedures, you must install the Trafodion JDBC Type 4 Driver on the client workstation. For more information, see the "Installing the JDBC Driver" (page 25).

Follow these instructions to download and install TrafCI on the client workstation.

- "Downloading the Installer" (page 26)
- "Running the Executable JAR Installer" (page 27)

To uninstall TrafCI, see "Uninstalling TrafCI" (page 33).

Downloading the Installer

1. Create a download folder on the client workstation.
2. Open a Web browser and navigate to the Trafodion downloads site (http://downloads.trafodion.org/).
3. Click on the clients-1.1.*n*.tar.gz link to start downloading the Trafodion clients tar file to your workstation.
4. Place the `clients-1.1.n.tar.gz` file into the download folder.

5. Unpack the `clients-1.1.`*n*`.tar.gz` file. For example:

   ```
   tar -xzf clients-1.1.0.tar.gz
   ```

   The package file contains the `trafci.zip` distribution file, which is extracted to the `clients` subdirectory.

6. Change the directory to the `clients` subdirectory.

7. Extract the contents of the `trafci.zip` file to a location on your client workstation by using the unzip command (or the WinZip extract tool):

   ```
   unzip trafci.zip
   ```

   The command extracts these files:

   - `README`
   - `trafciInstaller.jar`

8. Proceed with "Running the Executable JAR Installer" (page 27).

Running the Executable JAR Installer

When using the executable JAR file, `trafciInstaller.jar`, to install TrafCI, you have a choice of running the installer from the Installer Wizard Graphical User Interface (GUI) or from the command line:

- "Installer Wizard Steps" (page 27)
- "Command-Line Installation Steps" (page 31)

Installer Wizard Steps

NOTE: On Linux, to run the Installer Wizard, you must have the X Window system installed on the client workstation. If the client workstation does not have the X Window system, see the "Command-Line Installation Steps" (page 31).

Launching the Installer Wizard

1. Locate the `trafciInstaller.jar` file in the folder where you extracted the contents of the distribution (`.zip`) file.

2. Verify that the `trafciInstaller.jar` file appears as an executable JAR File.

   If not, skip the next two steps and go to Step 5.

3. Double-click the `trafciInstaller.jar` file to launch the Installer Wizard.

4. Proceed to "Using the Installer Wizard" (page 27).

5. At a command prompt, change to the directory where you extracted the installer files:
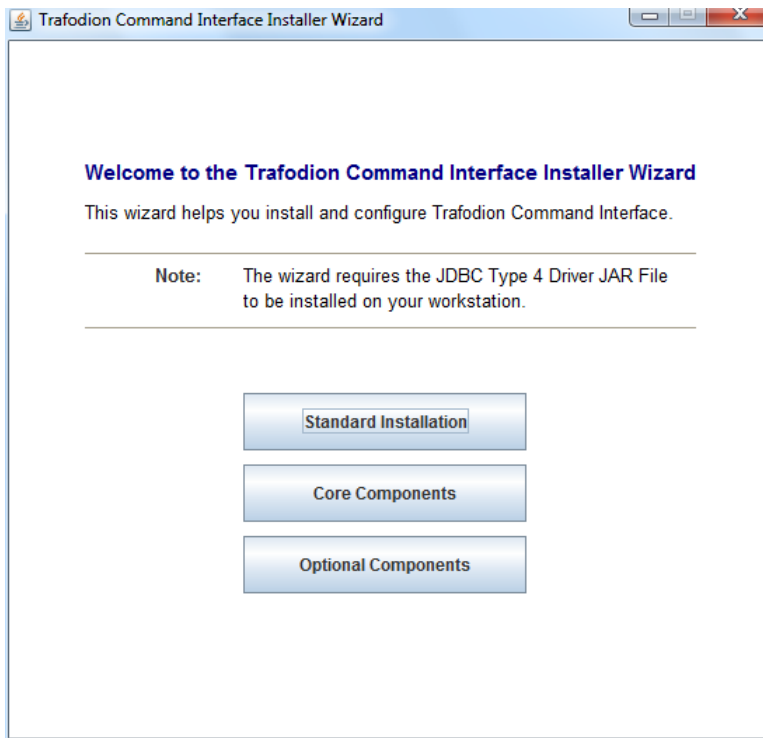
   ```
   cd installer-directory
   ```

   installer-directory is the directory where you extracted the installer file, `trafciInstaller.jar`.

6. Launch the Installer Wizard by entering:

   ```
   java -jar trafciInstaller.jar
   ```

7. Proceed to "Using the Installer Wizard" (page 27).

Using the Installer Wizard

When you execute `trafciInstaller.jar`, the Installer Wizard appears:

1. Click one of these buttons for the type of installation that you would like to perform:

   - Standard Installation to start the Installer Wizard, which guides you through installing both the core TrafCI components and the optional open source extensions

   - Core Components for a quick installation of the core TrafCI files

   - Optional Components if you have already installed the core TrafCI files but want to install only the optional open source extensions

2. After you have selected the components you wish to install, browse and select the JDBC JAR file and then specify an installation directory where you will install TrafCI.
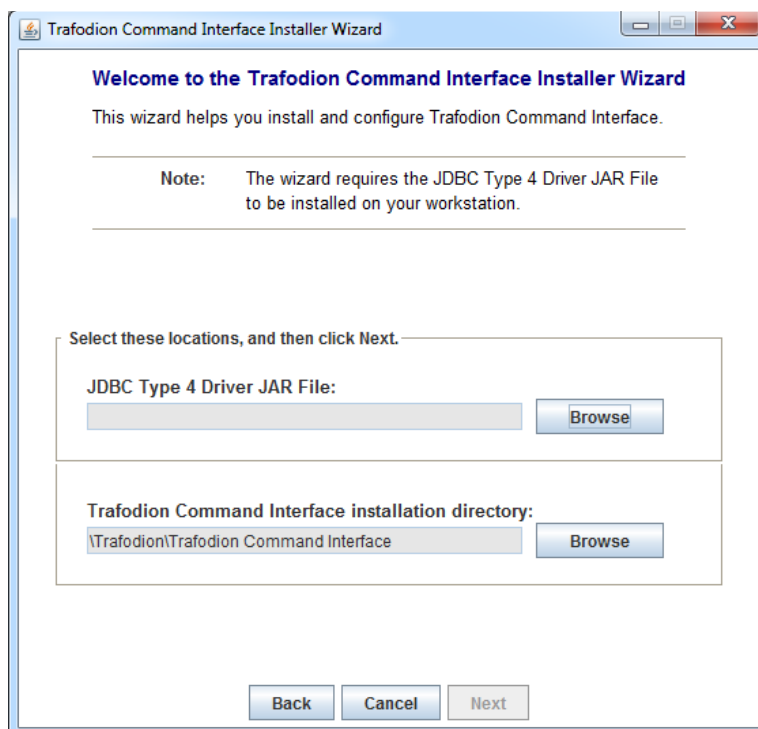
3. To locate the JDBC driver JAR file, click Browse next to the JDBC Type 4 Driver JAR File.
4. In the Select JDBC Type 4 Driver JAR File dialog box, navigate to and select the lib folder of the Trafodion JDBC driver, and then click Open.
5. Select jdbcT4.jar so that it appears in the File Name box, and then click Select.

   The Installer Wizard now displays the path of the JDBC driver JAR file.

6. To install in the default location, proceed to Step 9. To install in your own preferred location, proceed to Step 7.
7. To find an installation location for TrafCI, click Browse next to the Trafodion Command Interface installation directory.
8. In the Select Trafodion Command Interface Installation Directory dialog box, select the folder where you want to install TrafCI so that the directory path appears in the File Name box, and then click Select.

   The Installer Wizard displays the directory where TrafCI is installed.

9. Click Next to review the open source legal disclaimer.
10. If you agree to the terms and conditions, select the check box, and click Next.

    The Installer Wizard dialog box shows which components are available for you to download and install.
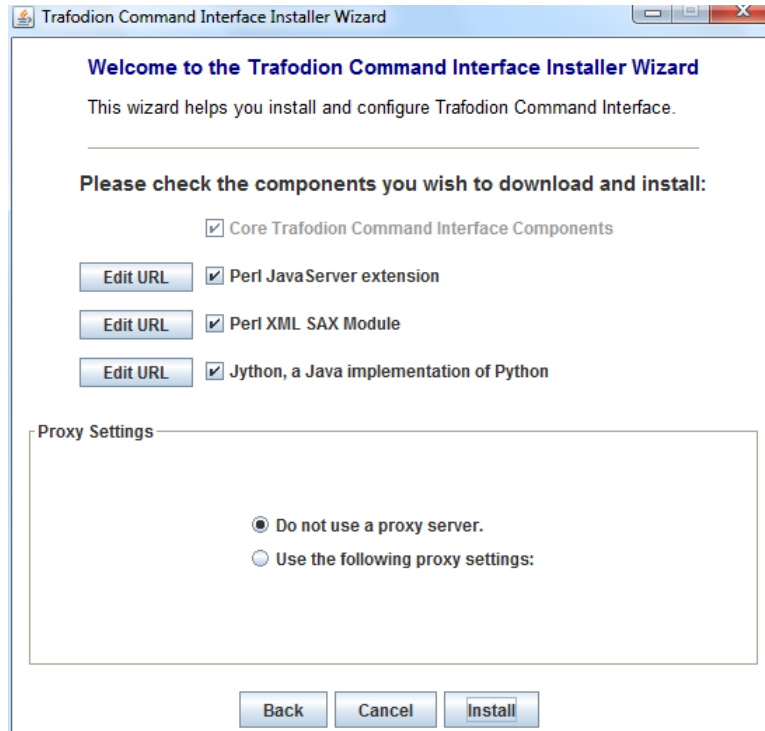
11. Select the optional components to be downloaded and installed. Each optional component is installed if the component box is checked.

    If you want to change the download URL for the extensions, click Edit URL, and this dialog box appears:
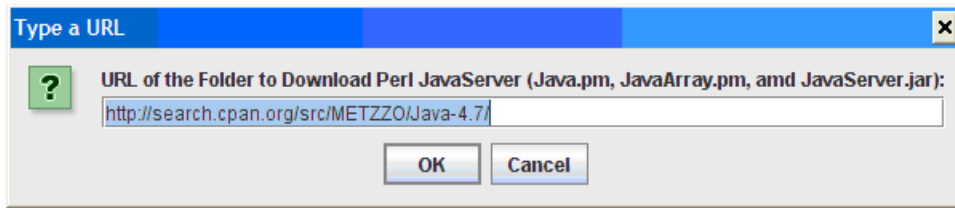
Type a new path, and click OK.

NOTE: Perl and Python must be installed for the respective extensions to work.

---

12. If you do not require a proxy server, proceed to Step 15.

13. If you require a proxy server, select Use the following proxy settings and enter the proxy server and port for downloading the open source extensions.
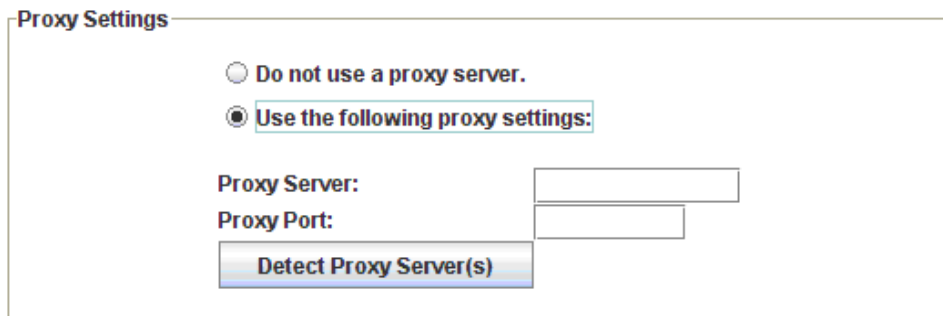

14. Click Detect Proxy Server(s) for TrafCI to try to auto-detect your proxy settings. If TrafCI detects one or more proxy servers, it displays them in a drop-down menu next to the Detect Proxy Server(s) button.

15. Click Install to start the installation.

16. After the core TrafCI files are installed, the Installation Status dialog box appears indicating how many files were extracted to the installation directory:
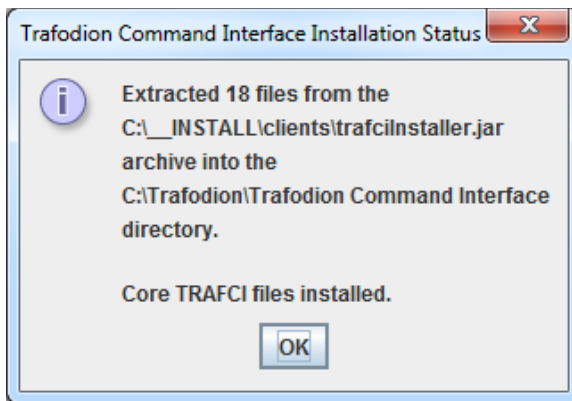
Click OK to continue the installation.

17. If you chose to install the optional components, the installer attempts to download and install them. The progress bar indicates the download progress of each file. In addition, an installation log provides details about the status of the download and installation of the components.
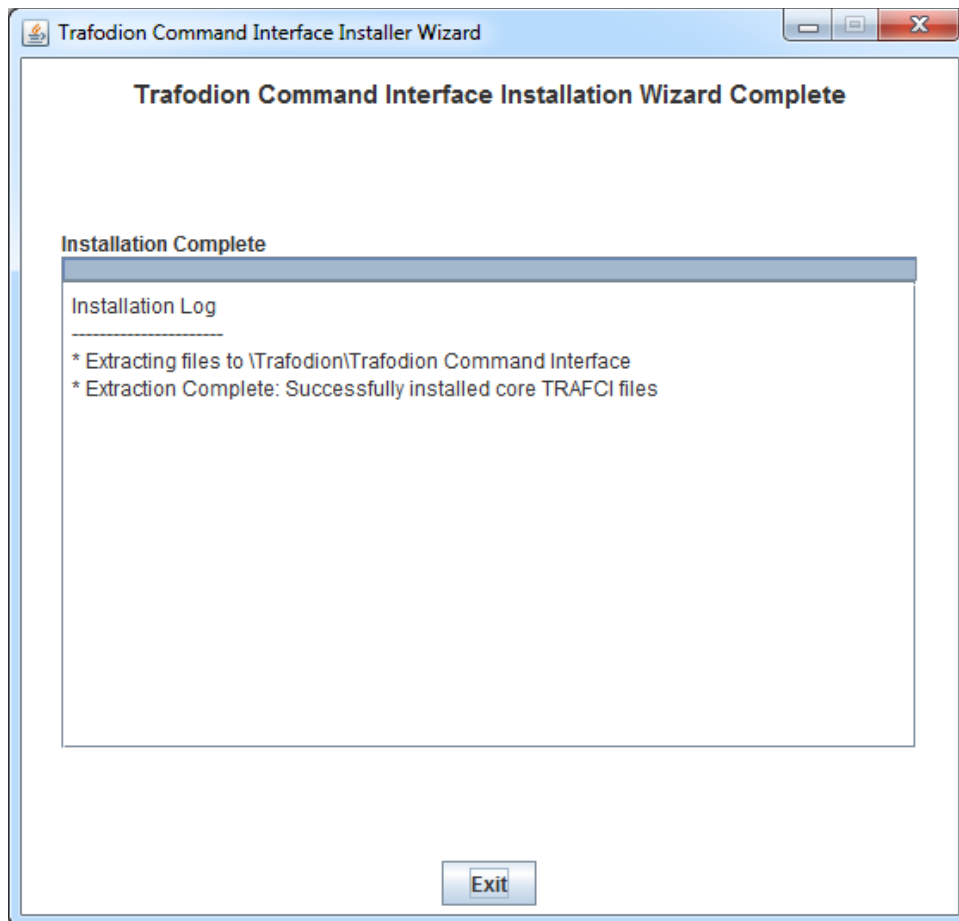
18. After all TrafCI files are installed, the Installer Wizard completes.
19. Click Exit.

Command-Line Installation Steps

1. At a command prompt, change to the directory where you extracted the contents of the distribution (`.zip`) file:

    ```
    cd installer-directory
    ```

    `installer-directory` is the directory where you extracted the installer files.

2. Before launching the command-line installer, see the command options below:

    ```
    java -jar trafciInstaller.jar -help

    Usage: java -jar <installer jar> [ -help] | <-cm|-silent> [-jdbcFile <jdbc filename>] [-installDir <install
     dir>] ]
    ```

    The `-silent` option installs the client without prompting you for options. For example:

    ```
    java -jar trafciInstaller.jar -silent -jdbcFile "C:\JDBC\lib\jdbcT4.jar" -installDir C:\TRAFCI
    ```

    `-jdbcFile` and `-installDir` are optional parameters. If you do not specify those parameters, you will be prompted to enter them during installation.

3. Launch the command-line installer by entering this command:

    ```
    java -jar trafciInstaller.jar cm
    ```

    The command-line installer starts and prompts you to enter the type of installation:

    ```
    /home/myname/trafcitemp>java -jar trafciInstaller.jar cm

    ****************************************************************
    ***                                                        ***
    *** Welcome to Trafodion Command Interface Installer        ***
    ***                                                        ***
    ```

```
***                                              ***
***    NOTE: The installer requires the JDBC Type 4    ***
***          Driver to be installed on your workstation.   ***
***                                              ***
*****************************************************************

Type Y for a standard installation, or N for optional components only.

Standard Installation [Y]:
```

- For a standard installation, type Y and press Enter.

- To install the optional components only, type N, press Enter, and proceed to Step 7.

NOTE: All items in square brackets are default values. Press Enter to accept the default value.

4. Enter the full directory path and file name of the JDBC driver JAR file, `jdbcT4.jar`, which is located in the JDBC driver `lib` directory:

```
JDBC Type 4 Driver JAR File
---------------------------
Enter the location and file name:
```

5. Enter an existing directory where you would like to install TrafCI:

```
Trafodion Command Interface
-------------------------------
Enter the installation directory:
```

The installation status appears, indicating how many files are installed in the installation directory:

```
Extracted 18 files from the
/home/myname/trafcitemp/trafciInstaller.jar
archive into the
/usr/local/trafci
directory.

Core TRAFCI files installed.

Do you want to install the optional components? [Y]:
```

6. If you do not wish to download and install the optional components, type N at the prompt and press Enter, and your installation is complete. Otherwise, type Y, press Enter, and proceed through the remainder of the installation.

7. Type Y and press Enter if you agree to the terms. If you are doing an optional install only, you are prompted to enter a valid TrafCI installation directory:

```
Do you agree to these terms? (Y or N): Y
Enter your installation directory:
```

8. If you do not require a proxy server, type N, press Enter, and proceed to Step 10. Otherwise, type Y, press Enter, and proceed to Step 9.

```
Use a proxy server? [N]:
```

9. When prompted to auto-detect proxy servers, type Y and press Enter to direct TrafCI to detect your proxy settings. If TrafCI finds proxy servers, it displays them. If you type N and press Enter, TrafCI prompts you to enter the proxy server and port:

```
Use a proxy server? [Y]: Y
Attempt to auto-detect proxy server(s)? [Y]: N
Enter the proxy server (do not include the port): myproxyserver.com
Enter the proxy port: 8080
```

10. You are prompted to select which optional components you wish to download and install. You can also change the download URL.

```
Install Perl JavaServer extensions? [Y]:
Perl JavaServer requires 3 files: Java.pm, JavaArray.pm, and JavaServer.jar
URL of the folder which contains these files [http://search.cpan.org/src/METZZO/Java-4.7/]:
```

```
Install Perl XML SAX Module? [Y]:
Perl SAX XML Module URL (PerlSAX.pm)
[http://search.cpan.org/src/KMACLEOD/libxml-perl-0.08/lib/XML/Parser/PerlSAX.pm]:

Install Jython, a Java implementation of Python? [Y]:
Jython URL (jython_installer-2.2.jar)
[http://fr.sourceforge.jp/frs/g_redir.php?m=jaist&f=%2Fjython%2Fjython%2F2.2%2Fjython_installer-2.2.jar]
```

11. The setup proceeds to download and install the optional open-source components. As each component is retrieved, dots (.) are printed to indicate the progress of the download.

```
Downloading Perl JavaServer [1 of 3] - Java.pm
......................... 100%
Downloading Perl JavaServer [2 of 3] - JavaArray.pm
......................... 100%
Downloading Perl JavaServer [3 of 3] - JavaServer.jar
......................... 100%
Successfully added settings.pl

Downloading Perl XML SAX Module [1 of 1] - PerlSAX.pm
......................... 100%

Downloading Jython [1 of 1] - jython_installer-2.2.jar
......................... 100%
Sucessfully Installed Jython.
Successfully added settings.py

Trafodion Command Interface Installation Complete.


/home/myname/trafcitemp>
```

Uninstalling TrafCI

If you used the executable JAR file, `trafciInstaller.jar`, to install TrafCI, delete the entire `Trafodion Command Interface` folder to uninstall TrafCI.

Post-Installation Instructions

Verifying the Installed Software Files

After downloading and running the installer file, verify that the TrafCI software files are installed in the correct locations. See Table 1.

Table 1 TrafCI Software Files

Folder	Files	Description
bin	trafci	
	trafci.cmd	Windows launch file
	trafci.pl	Perl wrapper script. `trafci-perl.pl` is renamed `trafci.pl`. To run this script, see the *Trafodion Command Interface Guide*.
	trafci.py	Python wrapper script. `trafci-python.py` is renamed as `trafci.py`. To run this script, see the *Trafodion Command Interface Guide*.
	trafci.sh	Linux launch file
	trafci-perl.pl	Perl wrapper script. This script has been modified to invoke `trafci.pl`. This script is retained for backward compatibility.

Table 1 TrafCI Software Files *(continued)*

Folder	Files	Description
	`trafci-python.py`	Python wrapper script. This script has been modified to invoke `trafci.py`. This script is retained for backward compatibility.
`lib`	`trafci.jar`	Product JAR file
`lib>perl`	`Session.pm`	Product file
`lib>python`	`Session.py`	Product file
`samples`	`README`	Readme file that describes how to use the sample scripts
	`arrayDML.pl`	Sample Perl program that executes DML statements and returns results in an array format
	`sample.pl`	Sample Perl program that supports multiple sessions in one script. `sample-beta.pl` has been renamed as `sample.pl`.
	`sample.sql`	Sample SQL script
	`sampleDDL.py`	Sample Python file that uses Jython to execute DDL statements
	`sampleDML.py`	Sample Python file that uses Jython to execute DML statements
	`sampleTables.pl`	Sample Perl file that lists all tables and respective rowcounts. The file accepts a wildcard argument on the command line.
	`sampleTables.py`	Sample Python file that lists all tables and respective rowcounts. The file accepts a wildcard argument on the command line.

Testing the Launch of TrafCI

Before launching TrafCI, make sure that you have set the Java path to the correct location. For more information, see:

- "Setting the PATH to a Supported Java Version on Windows" (page 21)
- "Setting the PATH to a Supported Java Version on Linux" (page 22)

If you did not set the Java path on your client workstation and you try to launch TrafCI, you might see the following error message appear momentarily in the TrafCI window before the TrafCI window disappears:

```
'java' is not recognized as an internal or external command, operable program or batch file.
```

For information about setting up the client, such as choosing the look and feel of the interface or presetting launch parameters, see the *Trafodion Command Interface Guide*.

6 Installing the Trafodion odb Tool

These installation instructions apply to Trafodion Release 1.1.n and subsequent updates of the product until otherwise indicated.

- "Installation Package" (page 35)
- "Installation Requirements" (page 35)
- "Installation Instructions" (page 35)

Installation Package

The distribution file for odb is packaged within the Trafodion clients tar file, `clients-1.1.n.tar.gz`, on the Trafodion downloads site (http://downloads.trafodion.org/). Use this distribution file to install odb:

Distribution File	File Contents	Client Operating System
odb64_linux.tar.gz	`README` `/bin/odb64luo`	64-bit Linux

Installation Requirements

The odb tool runs on 64-bit Linux. On a Linux workstation, odb requires pthread libraries, which are usually installed by default. It also requires the unixODBC Driver Manager to be installed and configured on the client workstation. For more information, see the *Trafodion odb User Guide*.

Installation Instructions

NOTE: Before following these installation instructions, please make sure to install and configure unixODBC on the client workstation. For more information, see the *Trafodion odb User Guide*.

Follow these instructions to download and install odb on the client workstation.

- "Downloading odb" (page 35)
- "Installing odb" (page 36)

To uninstall odb, see "Uninstalling odb" (page 36).

Downloading odb

1. Create a download folder on the client workstation.
2. Open a Web browser and navigate to the Trafodion downloads site (http:// downloads.trafodion.org/).
3. Click on the `clients-1.1.n.tar.gz` link to start downloading the Trafodion clients tar file to your workstation.
4. Place the `clients-1.1.n.tar.gz` file into the download folder.
5. Unpack the `clients-1.1.n.tar.gz` file. For example:

   ```
   tar -xzf clients-1.1.0.tar.gz
   ```

   The package file contains the `odb64_linux.tar.gz` distribution file, which is extracted to the `clients` subdirectory.
6. Proceed with "Installing odb" (page 36).

Installing odb

1. Change the directory to the `clients` subdirectory.
2. Unpack the contents of the `odb64_linux.tar.gz` file to a location on your client workstation:

   `tar -xzf odb64_linux.tar.gz`

   The command extracts these files:

   - `README`
   - `/bin/odb64luo`

   `odb64luo` is the odb executable.

3. You are now ready to run the odb executable. For more information, see the *Trafodion odb User Guide*.

Uninstalling odb

To uninstall odb, delete the `README` and `/bin/odb64luo` files from their installed location.

`rm -rf odb-installation-directory`

7 Configuring DbVisualizer

These configuration instructions apply to Trafodion Release 1.1.n and subsequent updates of Trafodion until otherwise indicated.

- "Prerequisite Software" (page 37)
- "Configuration Instructions" (page 37)

Prerequisite Software

Make sure that you have this software installed on your workstation:

- Java Runtime Environment (JRE) 1.7 or higher. See "Installing and Verifying the Java Runtime Environment (JRE)" (page 20). In addition, see DbVisualizer's FAQ "How to" page: How do I change the Java version that DbVisualizer uses?
- Trafodion JDBC Type 4 Driver. See "Installing the Trafodion JDBC Type 4 Driver" (page 19).
- DbVisualizer 9.x.x. See the DbVisualizer website.

Configuration Instructions

- "Disabling the Connection Validation Select Option" (page 37)
- "Registering the Trafodion JDBC Type 4 Driver in DbVisualizer" (page 38)
- "Connecting to Trafodion Through DbVisualizer" (page 39)

Disabling the Connection Validation Select Option

If you are using DbVisualizer 9.1 (or an earlier version), edit the `DbVisualizer-Install-Dir\resources\dbvis-custom.prefs` file and disable the `ConnectionValidationSelect` option as shown below:

`dbvis.generic.-ConnectionValidationSelect=disabled`

If you are using DbVisualizer 9.2 (or a later version), you will need to set the `Physical Connection` property to keep your connections alive. Follow these steps:

1. For the database connection, select the `Properties` tab.
2. In the left navigation tree, expand the `Generic` connection properties, and select `Physical Connection`.
3. Under Validation SQL, enter `values(current_timestamp)`.
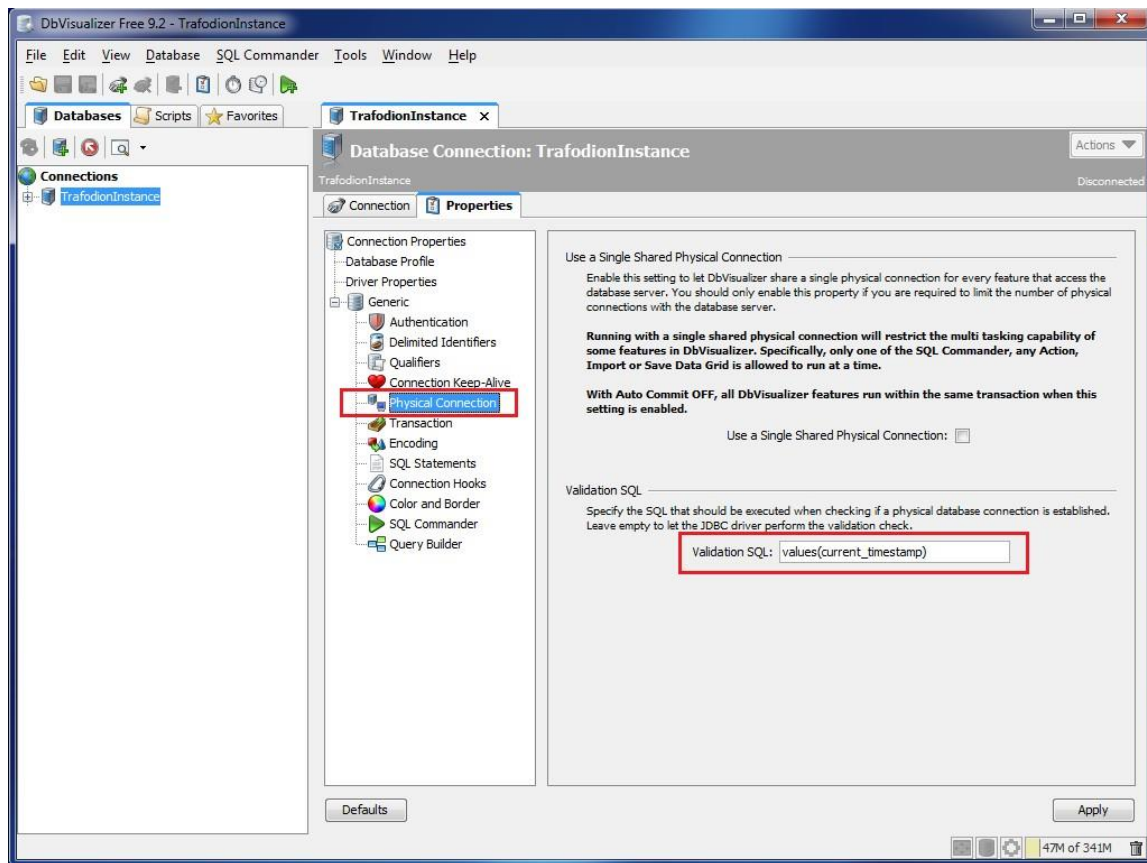
Registering the Trafodion JDBC Type 4 Driver in DbVisualizer

Use the DbVisualizer Driver Manager and register the Trafodion JDBC Type 4 Driver.
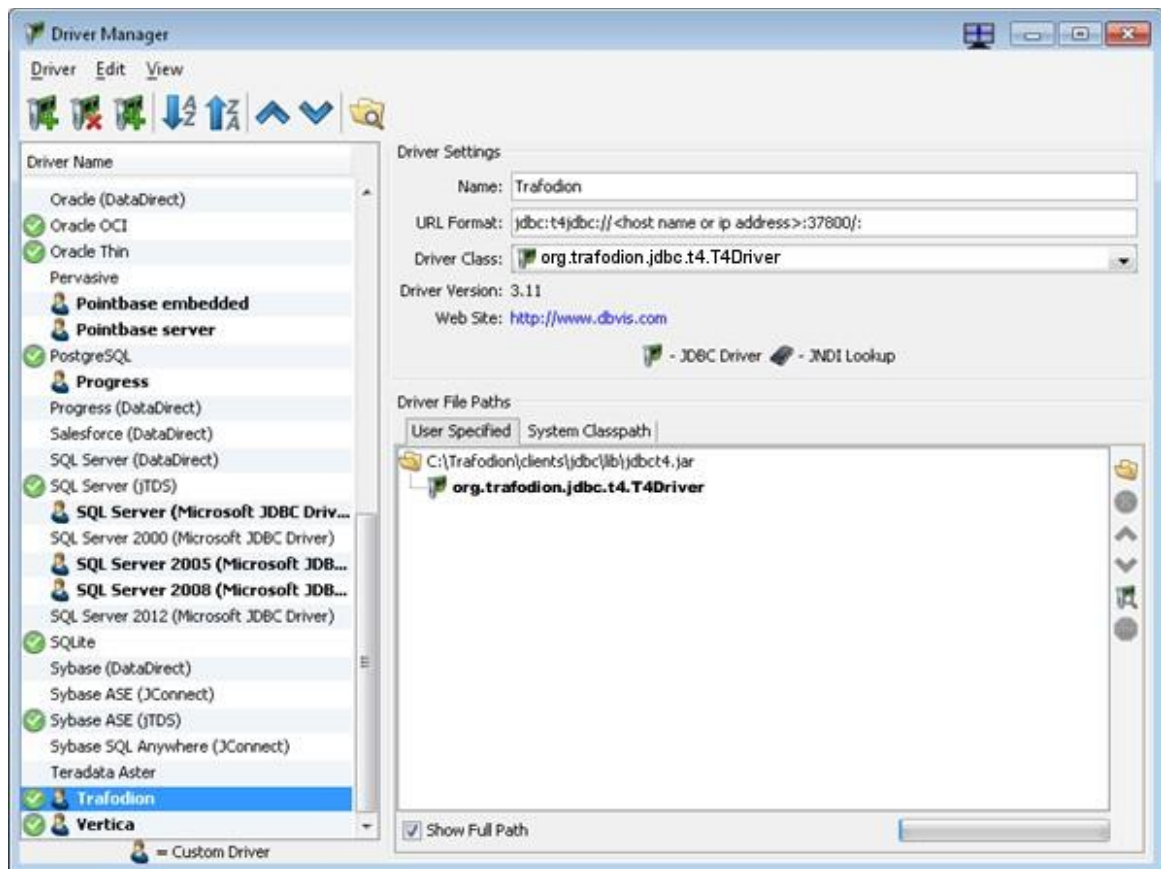
- Use the Open File icon and locate the Trafodion JDBC Type 4 Driver.
- Use the JDBC URL format:

  ```
  jdbc:t4jdbc://host-name or ip-address:37800/:
  ```

Connecting to Trafodion Through DbVisualizer

Create a new connection by selecting the Trafodion JDBC Type 4 Driver and filling in the connection parameters. Edit the database URL to match your Trafodion system's host name and port number:

```
jdbc:t4jdbc://host-name or ip-address:37800/:
```

Once you have connected successfully, execute a query using SQL Commander to confirm the connection.

8 Configuring SQuirreL SQL Client

These configuration instructions apply to Trafodion Release 1.1.n and subsequent updates of Trafodion until otherwise indicated.

- "Prerequisite Software" (page 41)
- "Configuration Instructions" (page 41)

Prerequisite Software

Make sure that you have this software installed on your workstation:

- Java Runtime Environment (JRE) 1.7 or higher. See "Installing and Verifying the Java Runtime Environment (JRE)" (page 20).
- Trafodion JDBC Type 4 Driver. See "Installing the Trafodion JDBC Type 4 Driver" (page 19).
- SQuirreL SQL Client 3.5.0. See the SQuirreL SQL Client website.

Configuration Instructions

- "Registering the Trafodion JDBC Type 4 Driver in the SQuirreL SQL Client" (page 41)
- "Connecting to Trafodion Through the SQuirreL SQL Client" (page 42)

Registering the Trafodion JDBC Type 4 Driver in the SQuirreL SQL Client

Use the "Add Driver" function and register the Trafodion JDBC Type 4 Driver:

- Click on the Add button and locate the Trafodion JDBC Type 4 Driver.
- Click on the List Drivers button to find the JDBC Driver Class Name.
- Set the properties as shown below:
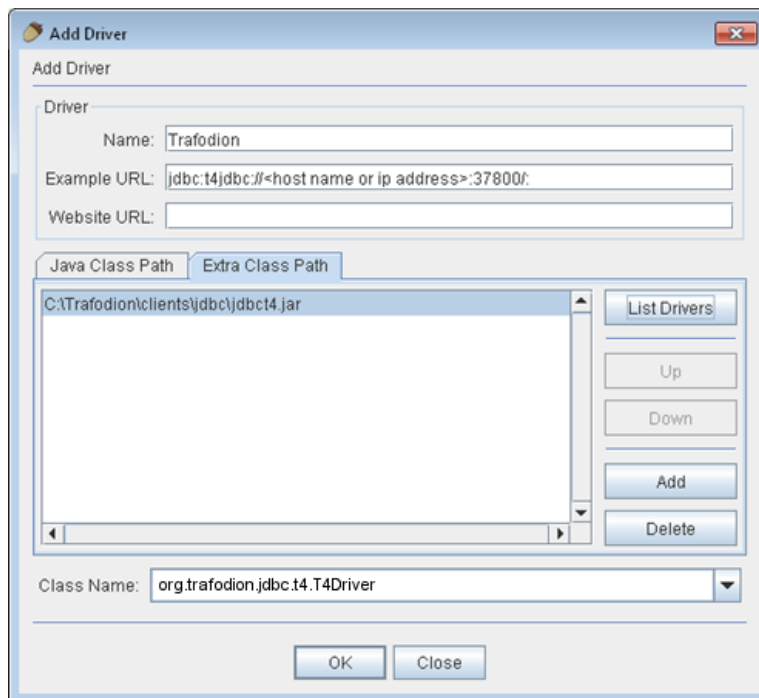
- ○ Name: `Trafodion`
- ○ Example URL: `jdbc:t4jdbc://host-name or ip-address:37800/:`

Connecting to Trafodion Through the SQuirreL SQL Client

Use the Add Alias dialog box and create an alias for your Trafodion System:
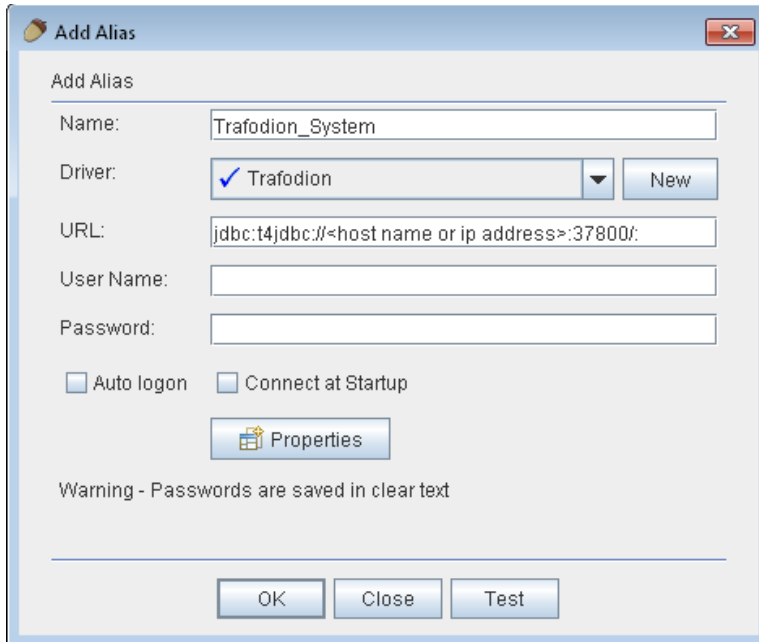
- Edit the connection URL to match your Trafodion system's host name and port number:
  `jdbc:t4jdbc://`*`host-name or ip-address`*`:37800/:`
- Click on the `Properties` button for the alias.
- In the Schemas tab, select the option `Load all and cache all schemas`.
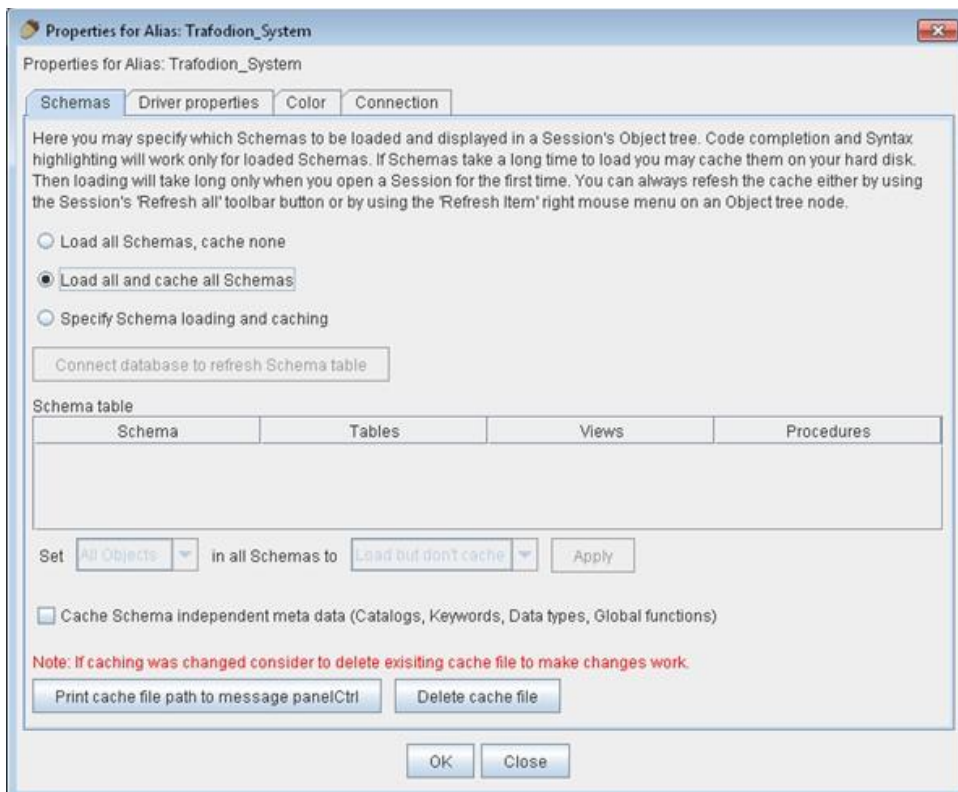
Once you have a successful connection, use the SQL tab and run a query to confirm the connection.

A ODBC Sample Program – Basic SQL

This appendix provides the source code for the ODBC sample program, basicsql, which is not currently bundled with the ODBC drivers. This appendix also provides the code for the script files that are needed to build and run the sample program on Windows. See . Copy and paste the code from this appendix into the recommended files. To build and run the sample program, see these instructions:

- On Windows:
- On Linux:

Source Code for the Sample Program—basicsql.cpp

Copy and paste the following code into a file named `basicsql.cpp`:

```cpp
// @@@ START COPYRIGHT @@@
//
//  Licensed under the Apache License, Version 2.0 (the "License");
//  you may not use this file except in compliance with the License.
//  You may obtain a copy of the License at
//
//       http://www.apache.org/licenses/LICENSE-2.0
//
//  Unless required by applicable law or agreed to in writing, software
//  distributed under the License is distributed on an "AS IS" BASIS,
//  WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
//  See the License for the specific language governing permissions and
//  limitations under the License.
//
// @@@ END COPYRIGHT @@@

#ifdef    linux
    #include <unistd.h>
#else
    #include <windows.h>
    #include <tchar.h>
#endif

//#include <stdarg.h>

#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <assert.h>

#include <sql.h>
#include <sqlext.h>

SQLHENV henv;
SQLHDBC hdbc;
SQLHSTMT hstmt;
SQLHWND hWnd;

#define MAX_SQLSTRING_LEN 1000
#define STATE_SIZE   6
#define MAX CONNECT STRING        256
#define TRUE    1
#define FALSE   0
#define ARGS    "d:u:p:"
```

```c
const char *SqlRetText(int rc)
{
 static char buffer[80];
 switch (rc)
 {
 case SQL_SUCCESS:
  return("SQL_SUCCESS");
 case SQL_SUCCESS_WITH_INFO:
  return("SQL_SUCCESS_WITH_INFO");
 case SQL_NO_DATA:
  return("SQL_NO_DATA");
 case SQL_ERROR:
  return("SQL_ERROR");
 case SQL_INVALID_HANDLE:
  return("SQL_INVALID_HANDLE");
 case SQL_STILL_EXECUTING:
  return("SQL_STILL_EXECUTING");
 case SQL_NEED_DATA:
  return("SQL_NEED_DATA");
 }
 sprintf(buffer,"SQL Error %d",rc);
 return(buffer);
}

void CleanUp()
{
 printf("\nConnect Test FAILED!!!\n");
 if(hstmt != SQL_NULL_HANDLE)
  SQLFreeHandle(SQL_HANDLE_STMT,hstmt);
 if(hdbc != SQL_NULL_HANDLE)
 {
  SQLDisconnect(hdbc);
  SQLFreeHandle(SQL_HANDLE_DBC,hdbc);
 }
 if(henv != SQL_NULL_HANDLE)
  SQLFreeHandle(SQL_HANDLE_ENV,henv);
 exit(EXIT_FAILURE);

}

void LogDiagnostics(const char *sqlFunction, SQLRETURN rc, bool exitOnError=true)
{
 SQLRETURN diagRC = SQL_SUCCESS;
 SQLSMALLINT recordNumber;
 SQLINTEGER nativeError;
 SQLCHAR messageText[SQL_MAX_MESSAGE_LENGTH];
 SQLCHAR sqlState[6];
 int diagsPrinted = 0;
 bool printedErrorLogHeader = false;

 printf("Function %s returned %s\n", sqlFunction, SqlRetText(rc));

 /* Log any henv Diagnostics */
 recordNumber = 1;
 do{
  diagRC = SQLGetDiagRec(SQL_HANDLE_ENV, henv,
recordNumber,sqlState,&nativeError,messageText,sizeof(messageText),NULL);
  if(diagRC==SQL_SUCCESS)
  {
   if(!printedErrorLogHeader){
    printf("Diagnostics associated with environment handle:\n");
    printedErrorLogHeader = true;
   }
   printf("\n\tSQL Diag %d\n\tNative Error: %ld\n\tSQL State:    %s\n\tMessage:      %s\n",
    recordNumber,nativeError,sqlState,messageText);
  }
  recordNumber++;
 } while (diagRC==SQL_SUCCESS);

   /* Log any hdbc Diagnostics */
 recordNumber = 1;
 printedErrorLogHeader = false;
 do{
  diagRC = SQLGetDiagRec(SQL_HANDLE_DBC, hdbc,
recordNumber,sqlState,&nativeError,messageText,sizeof(messageText),NULL);
  if(diagRC==SQL_SUCCESS)
  {
   if(!printedErrorLogHeader){
    printf("Diagnostics associated with connection handle:\n");
    printedErrorLogHeader = true;
   }
   printf("\n\tSQL Diag %d\n\tNative Error: %ld\n\tSQL State:    %s\n\tMessage:      %s\n",
    recordNumber,nativeError,sqlState,messageText);
  }
  recordNumber++;
 } while (diagRC==SQL_SUCCESS);

   /* Log any hstmt Diagnostics */
 recordNumber = 1;
 printedErrorLogHeader = false;
 do{
  diagRC = SQLGetDiagRec(SQL_HANDLE_STMT, hstmt,
```

```
                recordNumber,sqlState,&nativeError,messageText,sizeof(messageText),NULL);
    if(diagRC==SQL_SUCCESS)
    {
     if(!printedErrorLogHeader){
      printf("Diagnostics associated with statmement handle:\n");
      printedErrorLogHeader = true;
     }
     printf("\n\tSQL Diag %d\n\tNative Error: %ld\n\tSQL State:    %s\n\tMessage:        %s\n",
      recordNumber,nativeError,sqlState,messageText);
    }
    recordNumber++;
  } while (diagRC==SQL_SUCCESS);

  if(exitOnError && rc!=SQL_SUCCESS_WITH_INFO)
   CleanUp();
}

// Main Program
int main (int argc, char *argv[])
{
        unsigned char dsnName[20];
        unsigned char user[20];
        unsigned char password[20];
 SQLRETURN returnCode;
 bool   testPassed = true;
 SQLCHAR InConnStr[MAX_CONNECT_STRING];
 SQLCHAR  OutConnStr[MAX_CONNECT_STRING];
 SQLSMALLINT ConnStrLength;
 int errflag = 0;

 //optarg = NULL;
 if (argc != 4)
  errflag++;

 if (!errflag )
 {
                strcpy ((char *)dsnName, argv[1]);
                strcpy ((char *)dsnName, argv[1]);
                strcpy ((char *)user, argv[2]);
                strcpy ((char *)password, argv[3]);
        }
        if (errflag) {
  printf("Command line error.\n");
  printf("Usage: %s <datasource> <userid> <password>\n", argv[0] );
  return FALSE;
 }

 // Initialize handles to NULL
 henv = SQL_NULL_HANDLE;
 hstmt = SQL_NULL_HANDLE;
 hdbc = SQL_NULL_HANDLE;

 // Allocate Environment Handle
 returnCode = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &henv)",returnCode);

 // Set ODBC version to 3.0
 returnCode = SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLSetEnvAttr(henv, SQL_ATTR_ODBC_VERSION, (void*)SQL_OV_ODBC3, 0)",returnCode,false);

 // Allocate Connection handle
 returnCode = SQLAllocHandle(SQL_HANDLE_DBC, henv, &hdbc);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLAllocHandle(SQL_HANDLE_DBC, henv, &hdbc)", returnCode);

 //Connect to the database
 sprintf((char*)InConnStr,"DSN=%s;UID=%s;PWD=%s;%c",(char*)dsnName, (char*)user, (char*)password,'\0');
 printf("Using Connect String: %s\n", InConnStr);
 returnCode =
SQLDriverConnect(hdbc,hWnd,InConnStr,SQL_NTS,OutConnStr,sizeof(OutConnStr),&ConnStrLength,SQL_DRIVER_NOPROMPT);

 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLDriverConnect",returnCode);
 printf("Successfully connected using SQLDriverConnect.\n");

 //Allocate Statement handle
 returnCode = SQLAllocHandle(SQL_HANDLE_STMT, hdbc, &hstmt);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLAllocHandle(SQL_HANDLE_STMT, hdbc, &hstmt)", returnCode);

 printf("Drop sample table if it exists...\n");
 //Drop the test table if it exists
 //DROP IF EXISTS TASKS;
 returnCode = SQLExecDirect(hstmt, (SQLCHAR*)"DROP TABLE IF EXISTS TASKS", SQL_NTS);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLExecDirect of DROP", returnCode);

 printf("Creating sample table TASKS...\n");
 //Create a test table in default schema
 //CREATE TABLE TASKS (ID INT NOT NULL, TASK VARCHAR(10), LAST_UPDATE TIMESTAMP, PRIMARY KEY (C1));
 returnCode = SQLExecDirect(hstmt, (SQLCHAR*)"CREATE TABLE TASKS (ID INT NOT NULL, TASK CHAR(20), COMPLETED
```

```
 DATE, PRIMARY KEY (ID))", SQL_NTS);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLExecDirect of CREATE", returnCode);
 printf("Table TASKS created using SQLExecDirect.\n");

 printf("Inserting data using SQLBindParameter, SQLPrepare, SQLExecute\n");
 //Insert few rows into test table using bound parameters
 //INSERT INTO TASKS VALUES (?, ?, ?);
 SQLINTEGER intID;
 SQLLEN cbID = 0, cbTask = SQL_NTS, cbCompleted = 0;
 SQLCHAR strTask[200];
 SQL_DATE_STRUCT dsCompleted;

 returnCode = SQLBindParameter(hstmt, 1, SQL_PARAM_INPUT, SQL_C_SHORT, SQL_INTEGER, 0, 0, &intID, 0, &cbID);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLBindParameter 1", returnCode);

 returnCode = SQLBindParameter(hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR, SQL_CHAR, 0, 0, &strTask, 0, &cbTask);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLBindParameter 2", returnCode);

 returnCode = SQLBindParameter(hstmt, 3, SQL_PARAM_INPUT, SQL_C_TYPE_DATE, SQL_DATE, sizeof(dsCompleted), 0,
&dsCompleted, 0, &cbCompleted);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLBindParameter 3", returnCode);

 returnCode = SQLPrepare(hstmt, (SQLCHAR*)"INSERT INTO TASKS VALUES (?, ?, ?)", SQL_NTS);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLPrepare of INSERT", returnCode);

 intID = 1000;
 strcpy ((char*)strTask, "CREATE REPORTS");
 dsCompleted.year = 2014;
 dsCompleted.month = 3;
 dsCompleted.day = 22;

 returnCode = SQLExecute(hstmt);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLExecute", returnCode);
 printf("Data inserted.\n");

 //Select rows from test table and fetch the data
 //SELECT * from TASKS WHERE TASK LIKE '%REPORT%'
 printf("Fetching data using SQLExecDirect, SQLFetch, SQLGetData\n");
 returnCode = SQLExecDirect(hstmt, (SQLCHAR*)"SELECT ID, TASK, COMPLETED FROM TASKS", SQL_NTS);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLExecDirect of SELECT", returnCode);
 //loop thru resultset
 while (TRUE)
 {
  returnCode = SQLFetch(hstmt);
  if (returnCode == SQL_ERROR || returnCode == SQL_SUCCESS_WITH_INFO)
  {
   LogDiagnostics("SQLFetch", returnCode);
  }
  if (returnCode == SQL_SUCCESS || returnCode == SQL_SUCCESS_WITH_INFO)
  {
   SQLGetData(hstmt, 1, SQL_C_SHORT, &intID, 0, &cbID);
   SQLGetData(hstmt, 2, SQL_C_CHAR, strTask, 20, &cbTask);
   SQLGetData(hstmt, 3, SQL_C_TYPE_DATE, &dsCompleted, sizeof(dsCompleted), &cbCompleted);
   printf("Data selected: %d %s %d-%d-%d\n", intID, strTask, dsCompleted.year, dsCompleted.month, dsCompleted.day);

  }
  else
   break;
 }
 //Free Statement handle
 returnCode = SQLFreeHandle(SQL_HANDLE_STMT, hstmt);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLFreeHandle(SQL_HANDLE_STMT, hstmt)", returnCode);
 hstmt = SQL_NULL_HANDLE;

 //Disconnect
 returnCode = SQLDisconnect(hdbc);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLDisconnect(hdbc)", returnCode);

 //Free Connection handle
 returnCode = SQLFreeHandle(SQL_HANDLE_DBC, hdbc);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLFreeHandle(SQL_HANDLE_DBC, hdbc)", returnCode);
 hdbc = SQL_NULL_HANDLE;

 //Free Environment handle
 returnCode = SQLFreeHandle(SQL_HANDLE_ENV, henv);
 if(!SQL_SUCCEEDED(returnCode))
  LogDiagnostics("SQLFreeHandle(SQL_HANDLE_ENV, henv)", returnCode);
 henv = SQL_NULL_HANDLE;

 printf("Basic SQL ODBC Test Passed!\n");
 exit(EXIT_SUCCESS);
}
```

Build and Run Files for the Sample Program on Windows

The script files that are needed to build and run the sample program on Windows are not currently bundled with the ODBC driver. Copy and paste the code from this appendix into the recommended files. To build and run the sample program on Windows, see the instructions in "Running the Sample Program (basicsql)" (page 13).

Build Script—build.bat

Copy and paste the following code into a file named `build.bat`, which is used to build the sample program on Windows:

```
CL /c /Zi /nologo /W3 /WX- /O2 /D "NDEBUG" /D "_CRT_SECURE_NO_DEPRECATE" /D "_MBCS" /Gm /EHsc /GS /fp:precise
/Zc:wchar_t /Zc:forScope /Fo"./" /Gd /errorReport:queue basicsql.cpp

link /OUT:"./basicsql.exe" /NOLOGO "kernel32.lib" "user32.lib" "gdi32.lib" "winspool.lib" "comdlg32.lib"
"advapi32.lib" "shell32.lib" "ole32.lib" "oleaut32.lib" "uuid.lib" "odbc32.lib" "odbccp32.lib" /TLBID:1
/DYNAMICBASE /NXCOMPAT /MACHINE:X64 /ERRORREPORT:QUEUE basicsql.obj
```

To build the sample program on Windows, see the instructions in "Running the Sample Program (basicsql)" (page 13).

Run Script—run.bat

Copy and paste the following code into a file named `run.bat`, which is used to run the sample program on Windows:

```
basicsql.exe Default_Datasource user1 pwd1
```

To run the sample program on Windows, see the instructions in "Running the Sample Program (basicsql)" (page 13).

Index