



Apache

Trafodion

Client Installation Guide

Version 2.0.0

Table of Contents

1. About This Document	5
1.1. Intended Audience	5
1.2. New and Changed Information	5
1.3. Notation Conventions	5
1.4. Comments Encouraged	8
2. Introduction	9
2.1. Client Summary	9
2.1.1. JDBC-Based Clients	9
2.1.2. ODBC-Based Clients	9
2.2. Download Installation Package	10
2.2.1. Windows Download	10
2.2.2. Linux Download	11
3. Install JDBC Type-4 Driver	12
3.1. Installation Requirements	12
3.1.1. Java Environment	12
3.2. Installation Instructions	17
3.2.1. Install JDBC Type-4 Driver	17
3.3. Set Up Client Environment	18
3.3.1. Java Development	18
3.3.2. Configure Applications	19
3.4. Test Programs	20
3.5. Uninstall JDBC Type-4 Driver	22
3.6. Reinstall JDBC Type-4 Driver	24
4. Install trafci	25
4.1. Installation Requirements	25
4.1.1. Install Perl or Python	25
4.2. Installation Instructions	26
4.2.1. Run Executable JAR Installer	26
4.3. Post-Installation Instructions	38
4.3.1. Verify Installed Software Files	38
4.4. Test Launching trafci	39
4.4.1. Windows Example	40
4.4.2. Linux Example	41
4.5. Uninstall trafci	42
5. Configure DbVisualizer	43
5.1. Prerequisite Software	43
5.2. Configuration Instructions	43
5.2.1. Disable Connection Validation Select Option	43
5.2.2. Register JDBC Type-4 Driver	45
5.2.3. Connect to Trafodion	46
6. Configure Squirrel Client	47

6.1. Prerequisite Software	47
6.2. Configuration Instructions	48
6.2.1. Register JDBC Type-4 Driver	48
6.2.2. Connect to Trafodion	49
7. Install Linux ODBC Driver	51
7.1. Installation Requirements	51
7.2. Installation Instructions	51
7.2.1. Install/Reinstall Linux ODBC Driver	52
7.2.2. Set Up Client Environment	53
7.2.3. Enable Compression	54
7.2.4. Use Third-Party Driver Manager	55
7.2.5. Run Sample Program (<code>connect_test</code>)	56
7.2.6. Run Sample Program (<code>basicsql</code>)	58
8. Install odb	60
8.1. Installation Requirements	60
8.2. Installation Instructions	60
8.2.1. Install odb	60
8.3. Uninstall odb	60
9. Install Windows ODBC Driver	61
9.1. Installation Requirements	61
9.2. Installation Instructions	61
9.2.1. Install Windows ODBC Driver	61
9.2.2. Start the InstallShield wizard	62
9.2.3. Add a client data source	62
9.2.4. Set Up Client Environment	64
9.2.5. Enable Compression	65
9.2.6. Run Sample Program (<code>basicsql</code>)	67
9.3. Reinstall Windows ODBC Driver	68
9.4. Uninstalling Windows ODBC Driver	68
10. Appendix	69
10.1. <code>basicsql</code> (Sample ODBC Program)	69
10.1.1. <code>basicsql.cpp</code> Source Code	69
10.1.2. Windows Build and Run Files for <code>basicsql</code>	77

License Statement

Licensed to the Apache Software Foundation (ASF) under one or more contributor license agreements. See the NOTICE file distributed with this work for additional information regarding copyright ownership. The ASF licenses this file to you 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.

Acknowledgements

Microsoft®, Windows®, Windows NT®, Windows® XP, and Windows Vista® are U.S. registered trademarks of Microsoft Corporation. Intel® and Intel® Itanium® are trademarks of Intel Corporation in the U.S. and other countries. Java® is a registered trademark of Oracle and/or its affiliates. Motif, OSF/1, UNIX®, X/Open®, and the X device is a trademark of X/Open Company Ltd. in the UK and other countries.

OSF, OSF/1, OSF/Motif, Motif, and Open Software Foundation are trademarks of the Open Software Foundation in the U.S. and other countries. © 1990, 1991, 1992, 1993 Open Software Foundation, Inc.

The OSF documentation and the OSF software to which it relates are derived in part from materials supplied by the following: © 1987, 1988, 1989 Carnegie-Mellon University. © 1989, 1990, 1991 Digital Equipment Corporation. © 1985, 1988, 1989, 1990 Encore Computer Corporation. © 1988 Free Software Foundation, Inc. © 1987, 1988, 1989, 1990, 1991 Hewlett-Packard Company. © 1985, 1987, 1988, 1989, 1990, 1991, 1992 International Business Machines Corporation. © 1988, 1989 Massachusetts Institute of Technology. © 1988, 1989, 1990 Mentat Inc. © 1988 Microsoft Corporation. © 1987, 1988, 1989, 1990, 1991, 1992 SecureWare, Inc. © 1990, 1991 Siemens Nixdorf Informations systeme AG. © 1986, 1989, 1996, 1997 Sun Microsystems, Inc. © 1989, 1990, 1991 Transarc Corporation.

OSF software and documentation are based in part on the Fourth Berkeley Software Distribution under license from The Regents of the University of California. OSF acknowledges the following individuals and institutions for their role in its development: Kenneth C.R.C. Arnold, Gregory S. Couch, Conrad C. Huang, Ed James, Symmetric Computer Systems, Robert Elz. © 1980, 1981, 1982, 1983, 1985, 1986, 1987, 1988, 1989 Regents of the University of California. OSF MAKES NO WARRANTY OF ANY KIND

WITH REGARD TO THE OSF MATERIAL PROVIDED HEREIN, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. OSF shall not be liable for errors contained herein or for incidental consequential damages in connection with the furnishing, performance, or use of this material.

Revision History

Version	Date
2.0.0	To be announced.
1.3.0	January, 2016

Chapter 1. About This Document

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

1.1. Intended Audience

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

1.2. New and Changed Information

This manual shows updated versions for Trafodion Release 2.0.0. 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.

1.3. Notation Conventions

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.

```
SELECT
```

- lowercase letters

Lowercase letters, regardless of font, indicate variable items that you supply. Items not enclosed in brackets are required.

```
file-name
```


- [] Brackets

Brackets enclose optional syntax items.

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

- {} Braces

Braces enclose required syntax items.

```
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.

```
{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.

```
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.

```
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.

```
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.

```
match-value [NOT] LIKE _pattern  
    [ESCAPE esc-char-expression]
```

1.4. Comments Encouraged

We encourage 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 user@trafodion.incubator.apache.org.

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

Chapter 2. Introduction

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

2.1. Client Summary

2.1.1. JDBC-Based Clients

The following table lists JDBC-based clients supported by Trafodion.

Client Name	Description
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 .
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 .

2.1.2. ODBC-Based Clients

The following table lists ODBC-based clients supported by Trafodion.

Client Name	Description
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 ODBC Driver for Windows	[Not included in this release]¹ 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 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 .

¹ License issues prevent us from including the ODBC Driver for Windows in this release. Contact user@trafodion.incubator.apache.org for help obtaining the driver.

2.2. Download Installation Package

The Trafodion client software is available from the [Trafodion Download](#) page. There is one `Trafodion Clients` package per release listed under **<version> Binaries**.

The `Trafodion Clients` package consists of a zipped tar file that contains the Trafodion Clients tar file. The Trafodion Client binaries are located in the `Clients` folder, which contains the following files:

File	Usage
<code>odbc64_linux.tar.gz</code>	Trafodion odb tool.
<code>TFODBC64-*.exe</code>	[Not included in this release] ¹ Trafodion ODBC Driver for Windows.
<code>TRAF_ODBC_Linux_Driver_64.tar.gz</code>	Trafodion ODBC driver for Linux.
<code>trafci.zip</code>	The Trafodion command interpreter <code>trafci</code> .
<code>JDBCT4.zip</code>	Trafodion JDBC Type 4 Driver.

¹ License issues prevent us from including the ODBC Driver for Windows in this release. Contact user@trafodion.incubator.apache.org for help obtaining the driver.

2.2.1. Windows Download

Do the following:

1. Create a download folder on the client workstation. For example, `Trafodion Downloads`.
2. Open a Web browser and navigate to the Trafodion downloads site <http://http://trafodion.incubator.apache.org/download.html>.
3. Orient yourself to the binaries for the release you're installing. Click on the `Trafodion Clients` link to start downloading the Trafodion clients tar file to your workstation.
4. Place the `apache-trafodion-clients-*.tar.gz` file into the download folder.
5. Unpack the `apache-trafodion-clients-*.tar.gz` file using an unzip program of your choice. This creates an `apache-trafodion-clients-*.tar` file.
6. Unpack the `apache-trafodion-clients-*.tar` file using an unzip program of your choice. This creates the `clients` folder, which has the following content:

```
JDBCT4.zip odbc64_linux.tar.gz trafci.zip TRAF_ODBC_Linux_Driver_64.tar.gz
```

You use these files to install the different Trafodion clients.

2.2.2. Linux Download

Do the following:

1. Create a download directory on the client workstation. For example, `$HOME/trafodion-downloads`.
2. Open a Web browser and navigate to the Trafodion downloads site
<http://http://trafodion.incubator.apache.org/download.html>.
3. Orient yourself to the binaries for the release you're installing. Right-click on the `Trafodion Clients` link and select **Copy link address**.
4. Go to the download directory on the client workstation and use `wget` to download the client package using the link address you copied in step 3 above.
5. Unpack the `apache-trafodion-clients-*.tar.gz` using `tar`.

Example

```
$ mkdir $HOME/trafodion-downloads
$ cd $HOME/trafodion-downloads
$ wget <link to package>
$ tar -xzf apache-trafodion-clients-1.3.0-incubating-bin.tar.gz
$ cd clients
$ ls
JDBCT4.zip  odb64_linux.tar.gz  trafci.zip  TRAF_ODBC_Linux_Driver_64.tar.gz
$
```

You use these files to install the different Trafodion clients.

Chapter 3. Install JDBC Type-4 Driver

3.1. Installation Requirements

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

3.1.1. Java Environment

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.



If you plan to do Java-based development, install the Java Development Kit (JDK) rather than the Java Runtime Environment (JRE). These examples use JRE.

Verify Java Version

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

```
java -version
```

```
C:\> java -version

java version "1.7.0_45" # This is the version you need to check
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, please refer to [Installing a Supported Java Version](#) below.

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](#)
- [Setting the PATH to a Supported Java Version on Linux](#)

Install 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>

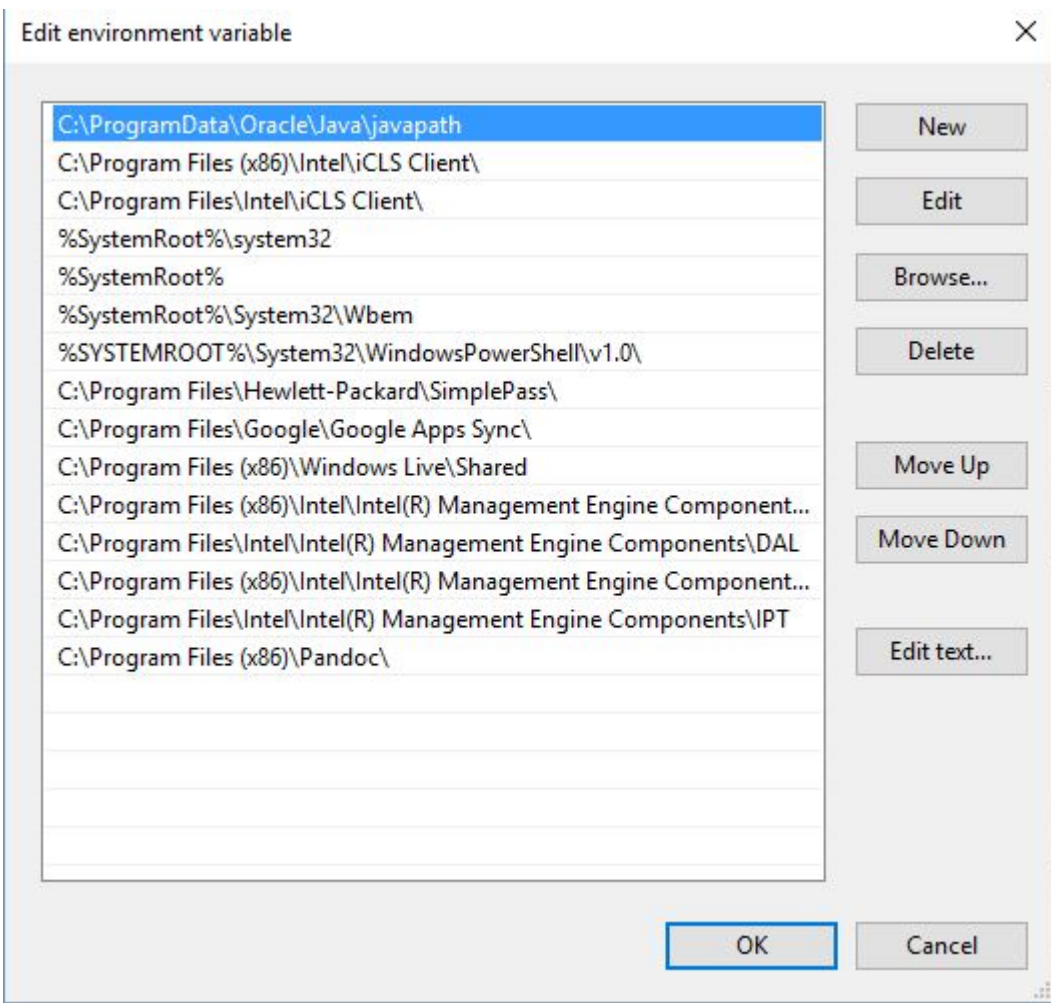
After installing the Java version, proceed with setting the Java path. Follow one of these sets of instructions, depending on the operating system of your client workstation:

- [Setting the PATH to a Supported Java Version on Windows](#)
- [Setting the PATH to a Supported Java Version on Linux](#)

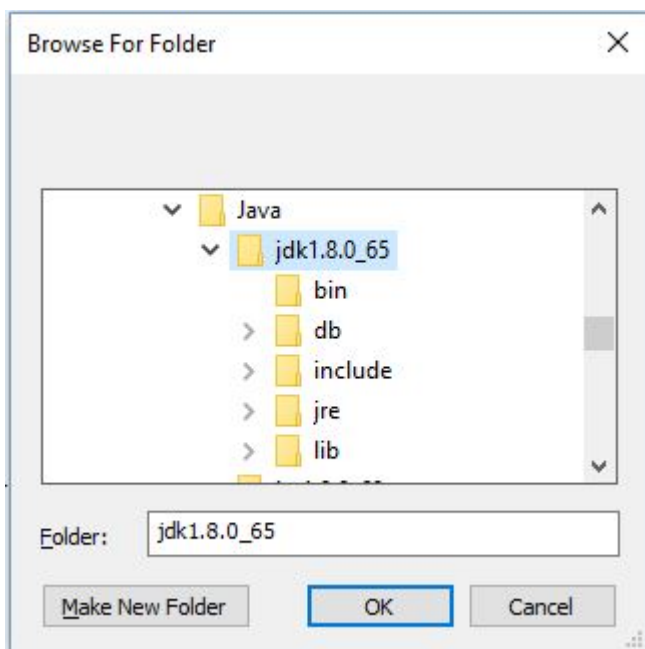
Set Windows PATH Variable

Windows 10

1. Right-click on the Windows icon on the menu bar. Select **System**.
2. Click on **Advanced System Settings**.
3. In the **System Properties** dialog box, click the **Advanced** tab.
4. Click the **Environment Variables** button.
5. Under **System** variables, select the variable named **Path**, and then click **Edit. . .**:



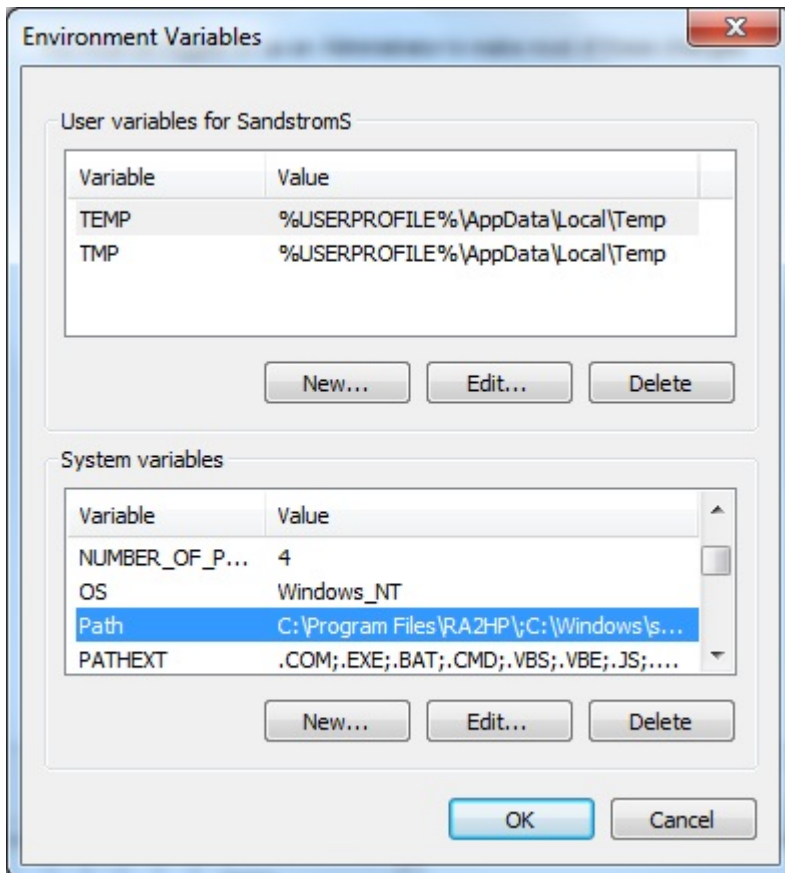
6. Click **Browse...** Find the directory where you installed Java and select it.



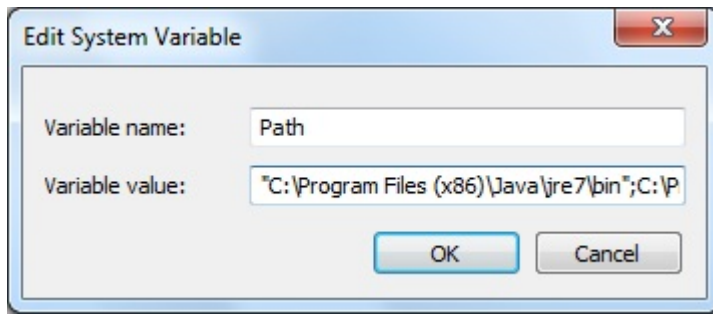
7. Click **OK** to close the browse window. Click **OK** to close the edit window.
8. Verify that the updated **Path** appears under **System** variables, and click **OK**.
9. In the **System Properties** dialog box, click **OK** to accept the changes.

Windows 8

1. Open system 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 start of the **Variable** value field and enter the path of the Java bin directory, ending with a semicolon (;):



Example

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



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.

Set Linux PATH Variable

1. Open the user profile (`.profile` or `.bash_profile` for the Bash shell) in the `$HOME` directory.

```
cd $HOME
vi .profile
```

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

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



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.

```
..profile
```

3.2. Installation Instructions

You download and extract the Trafodion client package using the instructions in [Download Installation Package](#) above.

3.2.1. Install JDBC Type-4 Driver

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

Example

```
unzip JDBCT4.zip -d $HOME/jdbc
```

The content of the target directory is as follows:

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.
/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.

3.3. Set Up Client Environment

3.3.1. Java Development

If you plan to write and run Java applications that use the Trafodion JDBC Type 4 Driver to connect to a Trafodion database, then 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=" <i>jdk-directory</i> " ¹	export JAVA_HOME= <i>jdk-directory</i>
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:

¹ Enclose the *jdk-directory* in quotes to ensure that Windows can find the directory correctly. You can use the `set <variable>` command to verify the setting.

3.3.2. Configure Applications

Edit the `t4jdbc.properties` file in the `samples` folder. Refer to the `README` file in the `samples` folder for instructions.

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 **23400** by default. See the [Trafodion Database Connectivity Services Reference Guide](#) for information about how to configure the DCS port.

Example

In this example, Trafodion authentication has not been enabled. Therefore, you can use a dummy user and password. If authentication is enabled, then use your user and password information.

```
catalog = TRAFODION
schema = SEABASE
user = usr
password = pwd

url = jdbc:t4jdbc://trafodion.host.com:23400/:
```



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

3.4. Test Programs

The README file in the `samples` folder provide information for how you build and run sample Java programs. You can use these programs to verify the setup of the Trafodion JDBC Type-4 driver. See the [Install JDBC Type-4 Driver](#) section above for information on the different sample programs that are included with the Trafodion JDBC Type-4 driver.

Example

Build and run the `StatementSample` test program to verify the JDBC Type-4 driver installation.

```
C:\Development Tools\Trafodion JDBCT4\samples>cd StatementSample

C:\Development Tools\Trafodion JDBCT4\samples\StatementSample>%JAVA_HOME%\bin\javac
-classpath ..\..\lib\jdbcT4.jar *.java ..\common\*.java
Note: ..\common\sampleUtils.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.
C:\Development Tools\Trafodion JDBCT4\samples\StatementSample>%JAVA_HOME%\bin\java
-classpath ..\..\lib\jdbcT4.jar -Dt4jdbc.properties=..\t4jdbc.properties
StatementSample
Mar 16, 2016 9:36:54 PM common.sampleUtils.getPropertiesConnection
INFO: DriverManager.getConnection(url, props) passed

Inserting TimeStamp

Simple Select

Printing ResultSetMetaData ...
No. of Columns 12
Column 1 Data Type: CHAR Name: C1
Column 2 Data Type: SMALLINT Name: C2
Column 3 Data Type: INTEGER Name: C3
Column 4 Data Type: BIGINT Name: C4
Column 5 Data Type: VARCHAR Name: C5
Column 6 Data Type: NUMERIC Name: C6
Column 7 Data Type: DECIMAL Name: C7
Column 8 Data Type: DATE Name: C8
Column 9 Data Type: TIME Name: C9
Column 10 Data Type: TIMESTAMP Name: C10
Column 11 Data Type: REAL Name: C11
Column 12 Data Type: DOUBLE PRECISION Name: C12

Fetching rows...

Printing Row 1 using getString(), getObject()
Column 1 - Row1                ,Row1
Column 2 - 100,100
Column 3 - 12345678,12345678
Column 4 - 123456789012,123456789012
Column 5 - Selva,Selva
Column 6 - 100.12,100.12
```

```
Column 7 - 100.12,100.12
Column 8 - 2000-05-06,2000-05-06
Column 9 - 10:11:12,10:11:12
Column 10 - 2000-05-06 10:11:12.000000,2000-05-06 10:11:12.0
Column 11 - 100.12,100.12
Column 12 - 100.12,100.12
```

```
Printing Row 2 using getString(), getObject()
```

```
Column 1 - Row2 ,Row2
Column 2 - -100,-100
Column 3 - -12345678,-12345678
Column 4 - -123456789012,-123456789012
Column 5 - Selva,Selva
Column 6 - -100.12,-100.12
Column 7 - -100.12,-100.12
Column 8 - 2000-05-16,2000-05-16
Column 9 - 10:11:12,10:11:12
Column 10 - 2000-05-06 10:11:12.000000,2000-05-06 10:11:12.0
Column 11 - -100.12,-100.12
Column 12 - -100.12,-100.12
```

```
Printing Row 3 using getString(), getObject()
```

```
Column 1 - TimeStamp ,TimeStamp
Column 2 - -100,-100
Column 3 - -12345678,-12345678
Column 4 - -123456789012,-123456789012
Column 5 - Selva,Selva
Column 6 - -100.12,-100.12
Column 7 - -100.12,-100.12
Column 8 - 2016-03-16,2016-03-16
Column 9 - 21:37:03,21:37:03
Column 10 - 2016-03-16 21:37:03.053,2016-03-16 21:37:03.053
Column 11 - -100.12,-100.12
Column 12 - -100.12,-100.12
```

```
End of Data
```

```
C:\Development Tools\Trafodion JDBC4\samples\StatementSample>
```


3.5. Uninstall 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>
```

Example

```
rm -rf ~/jdbc
```

- On Windows:

```
del <jdbc-installation-directory>  
rmdir <jdbc-installation-directory>
```

Example

Windows uninstall

```
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:\>
```

3.6. Reinstall JDBC Type-4 Driver

1. Close all applications running on the workstation, except the Web browser.
2. Download and extract the Trafodion client package using the instructions in [Download Installation Package](#) above.
3. Install the new Trafodion JDBC Type-4 driver. See [Install JDBC Type-4 Driver](#).
4. Set up the client environment. Please refer to: [Set Up Client Environment](#).

Chapter 4. Install trafci

4.1. Installation Requirements

The Trafodion Command Interface (trafci) runs on Windows and Linux. Before installing trafci on the client workstation, ensure that you've installed the JDBC Type-4 driver. See the [Install JDBC Type-4 Driver](#) chapter above.

4.1.1. Install 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.

4.2. Installation Instructions

You download and extract the Trafodion client package using the instructions in [Download Installation Package](#) above.

1. Change the directory to the `clients` subdirectory.
2. Extract the contents of the `trafci.zip` file to a location on your client workstation (for example, a folder named `trafci`) by using the `unzip` command (or the `extract` function of your compression software):

```
cd $HOME/trafodion-download/clients
unzip trafci.zip -d trafci
cd trafci
```

The command extracts these files:

- `README`
- `trafciInstaller.jar`

4.2.1. Run 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](#)
- [Command-Line Installation Steps](#)

Installer Wizard Steps



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 [Command-Line Installation Steps](#).

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](#).
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
```

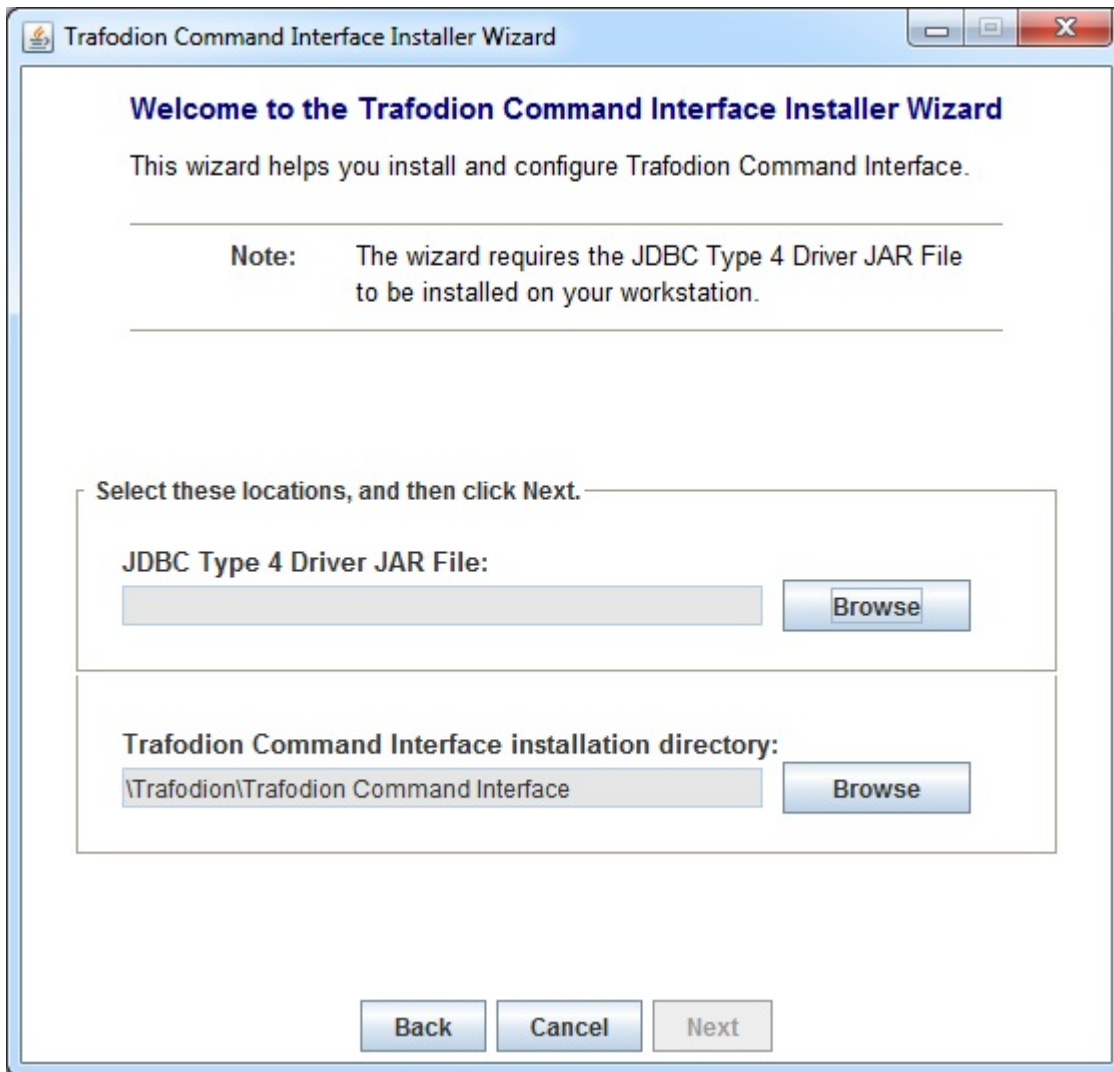
7. Proceed to [Using the Installer Wizard](#).

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.



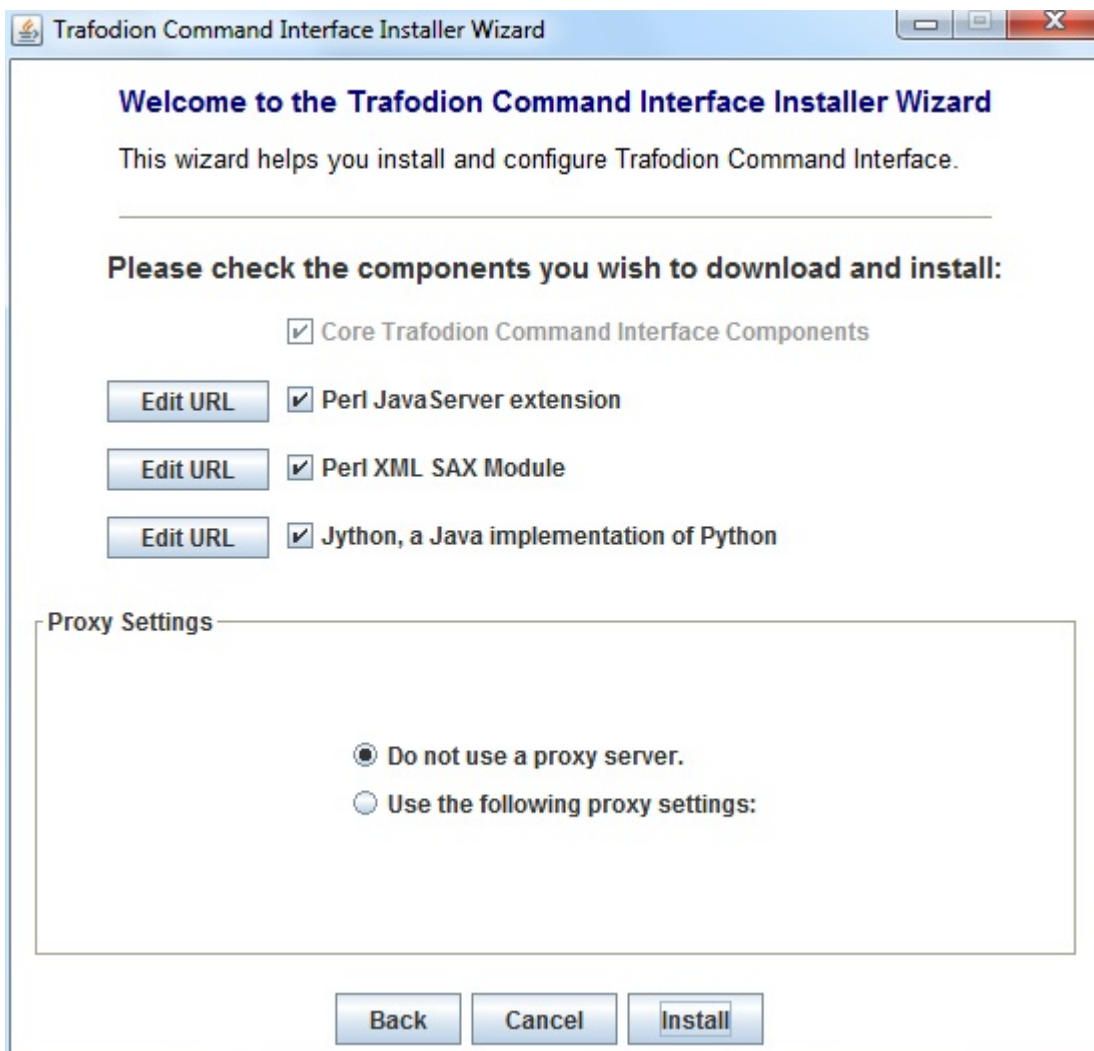
6. To install in the default location, proceed to Step 9. To install in your own preferred location, proceed to Step 7.

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.

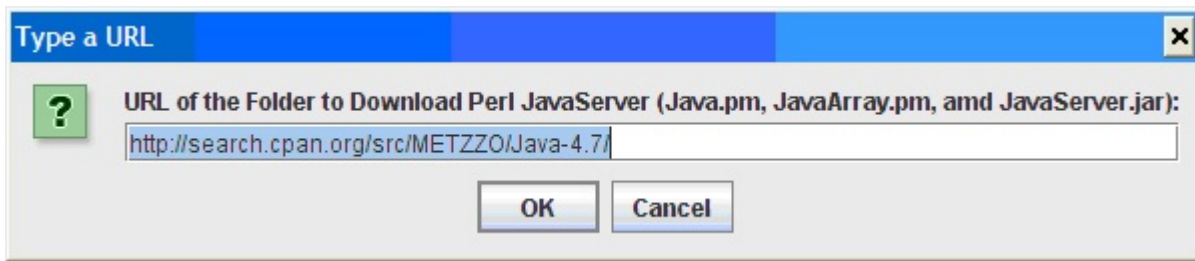
9. Click **Next** to review the open-source legal disclaimer.
10. If you agree to the terms and conditions, select the check box, and click **Next**.

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:

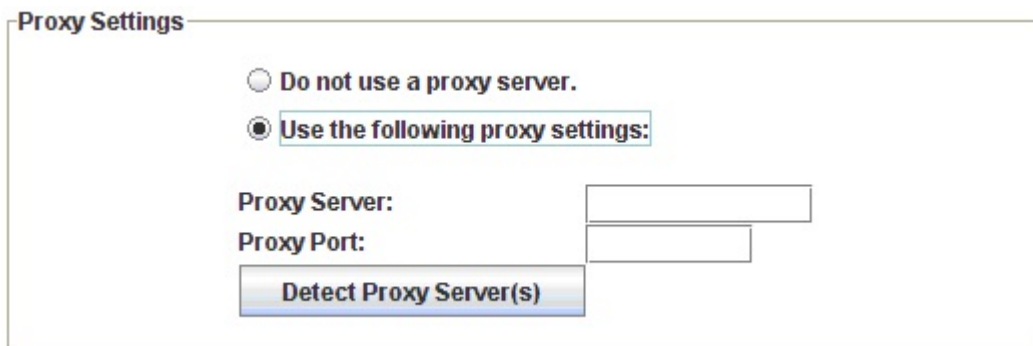


Type a new path, and click **OK**.

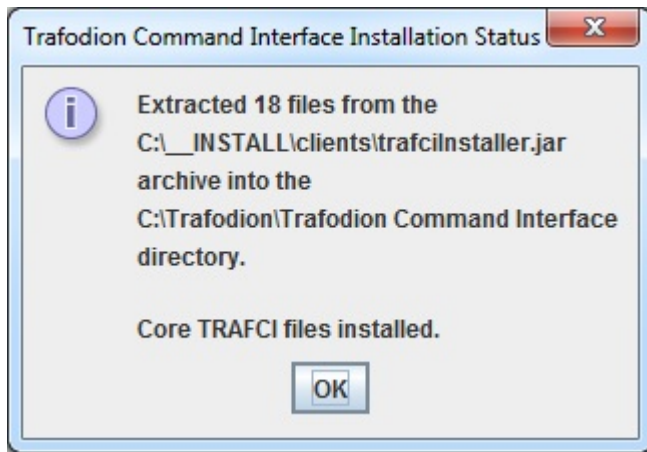


Perl and Python must be installed for the respective extensions to work.

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.

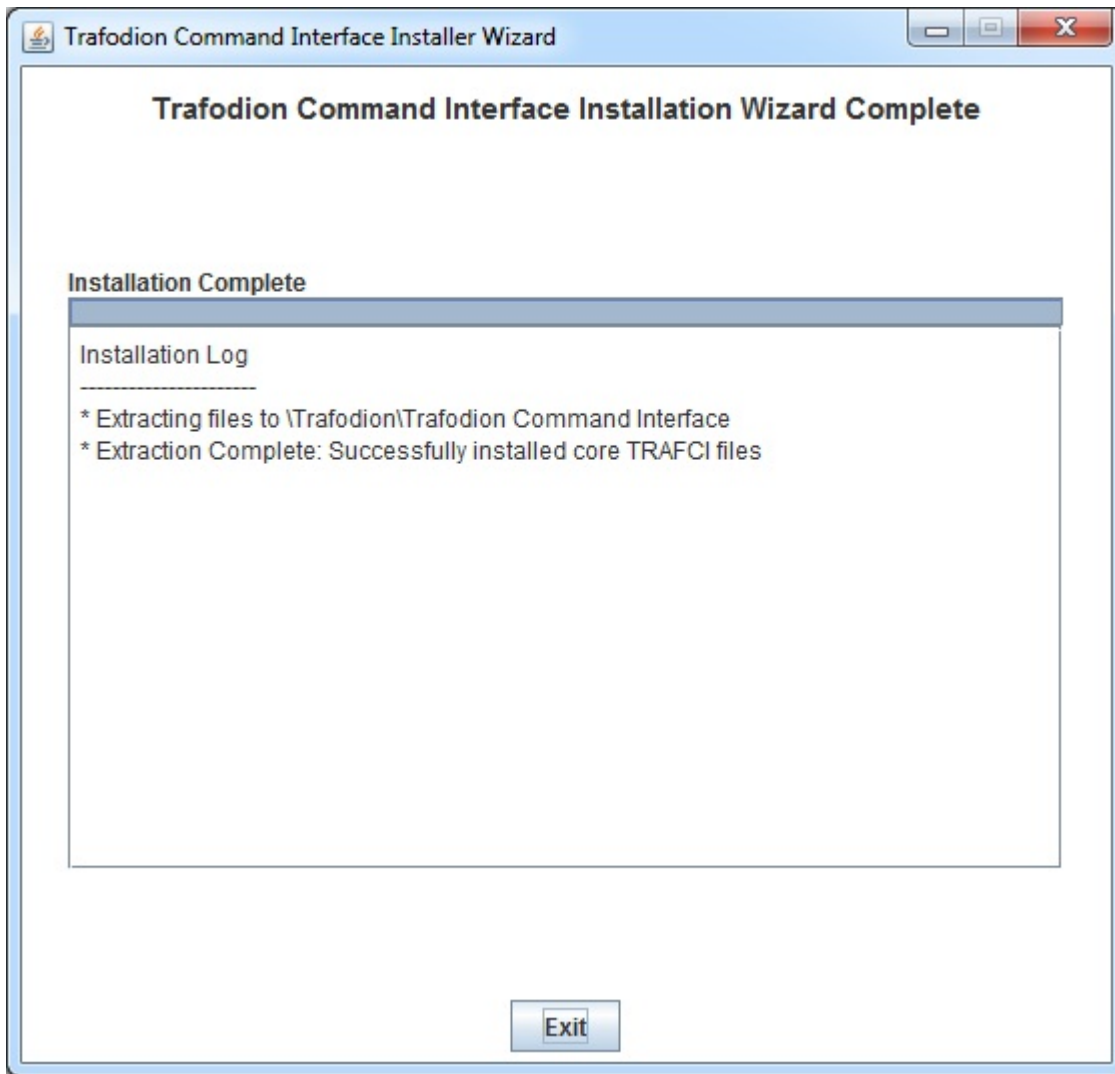


14. Click **Detect Proxy Server(s)** to try to auto-detect your proxy settings. If trafci detects one or more proxy servers, it displays them in a drop-down menu next to the **Detect Proxy Server(s)** button.
15. Click **Install** to start the installation.
16. After the core trafci files are installed, 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.

Example

```
$ cd $HOME/trafodion-download/clients/trafci
$ ls
README  trafciInstaller.jar
```

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.

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

`-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 a the JDBC Type 4                  **
****       Driver to be installed a 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.



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 TRAFICI 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]: Y

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

Install Perl XML SAX Module? [Y]: Y

Perl SAX XML Module URL (PerlSAX.pm)

Install Jython, a Java implementation of Python? [Y]: Y

Jython URL (jython_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.pml
..... 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%
Successfully Installed Jython. Successfully added settings.py
Trafodion Command Interface Installation Complete.
/home/myname/trafcitemp>
```


4.3. Post-Installation Instructions

4.3.1. Verify Installed Software Files

After downloading and running the installer file, verify that the trafci software files are installed in the correct locations:

Folder	Files	Description
bin	trafci	
	trafci.cmd	Windows launch file.
	trafci.pl	Perl wrapper script. <i>trafci-perl.pl</i> is renamed <i>trafci.pl</i> . To run this script, see the Trafodion Command Interface Guide .
	trafci.py	Python wrapper script. <i>trafci-python.py</i> is renamed as <i>trafci.py</i> . 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 <i>trafci.pl</i> . This script is retained for backward compatibility.
	trafci-python.py	Python wrapper script. This script has been modified to invoke <i>trafci.py</i> . 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.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 row counts. The file accepts a wild-card argument on the command line.
	sampleTables.py	Sample Python file that lists all tables and respective row counts. The file accepts a wild-card argument on the command line.

4.4. Test Launching 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](#)
- [Setting the PATH to a Supported Java Version on Linux](#)

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 and using trafci, such as choosing the look and feel of the interface or presetting launch parameters, see the [Trafodion Command Interface Guide](#).

4.4.1. Windows Example

On Windows, do the following:

1. Go to the directory where you installed trafci. For example, `c:\Trafodion\Trafodion Command Interface`
2. Go to the `bin` directory
3. Invoke the `trafci.cmd` file.
4. Answer prompts.

```
cd "c:\Trafodion\Trafodion Command Interface"
cd bin
trafci.cmd
<screen is cleared>

Welcome to Apache Trafodion Command Interface
Copyright (c) 2015 Apache Software Foundation

Host Name/IP Address: trafodion.host.com:23400
User Name: usr
Password:

Connected to Trafodion

SQL> show schemas ;

Welcome to Apache Trafodion Command Interface
Copyright (c) 2015 Apache Software Foundation

Host Name/IP Address: 10.1.30.28:23400
User Name: usr
Password:

Connected to Trafodion

SQL>show schemas;

SCHEMA NAMES
-----
SEABASE  _MD_      _REPOS_   _LIBMGR_

SQL>
```

4.4.2. Linux Example

On Linux, do the following:

1. Go to the directory where you installed trafci. For example, `$HOME/trafci`
2. Go to the `bin` directory
3. Invoke the `trafci.sh` file.
4. Answer prompts.

```
$ cd $HOME/trafci/bin
$ . ./trafci.sh -h trafodion.home.com:23400 -u usr -p pwd

Welcome to Apache Trafodion Command Interface
Copyright (c) 2015 Apache Software Foundation

Connected to Trafodion

SQL>show schemas;

SCHEMA NAMES
-----
SEABASE   _MD_     _REPOS_  _LIBMGR_

SQL>
```

4.5. Uninstall trafci

If you used the executable JAR file, `trafciInstaller.jar`, to install trafci, delete the entire folder/directory when you installed trafci.

Chapter 5. Configure DbVisualizer

5.1. Prerequisite Software

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

- Java Runtime Environment (JRE) 1.7 or higher. See [Java Environment](#) in the [Install JDBC Type-4 Driver](#) chapter above.

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 [Install JDBC Type-4 Driver](#) above.
- DbVisualizer 9.x.x. See the [DbVisualizer website](#).

5.2. Configuration Instructions

5.2.1. Disable Connection Validation Select Option

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:

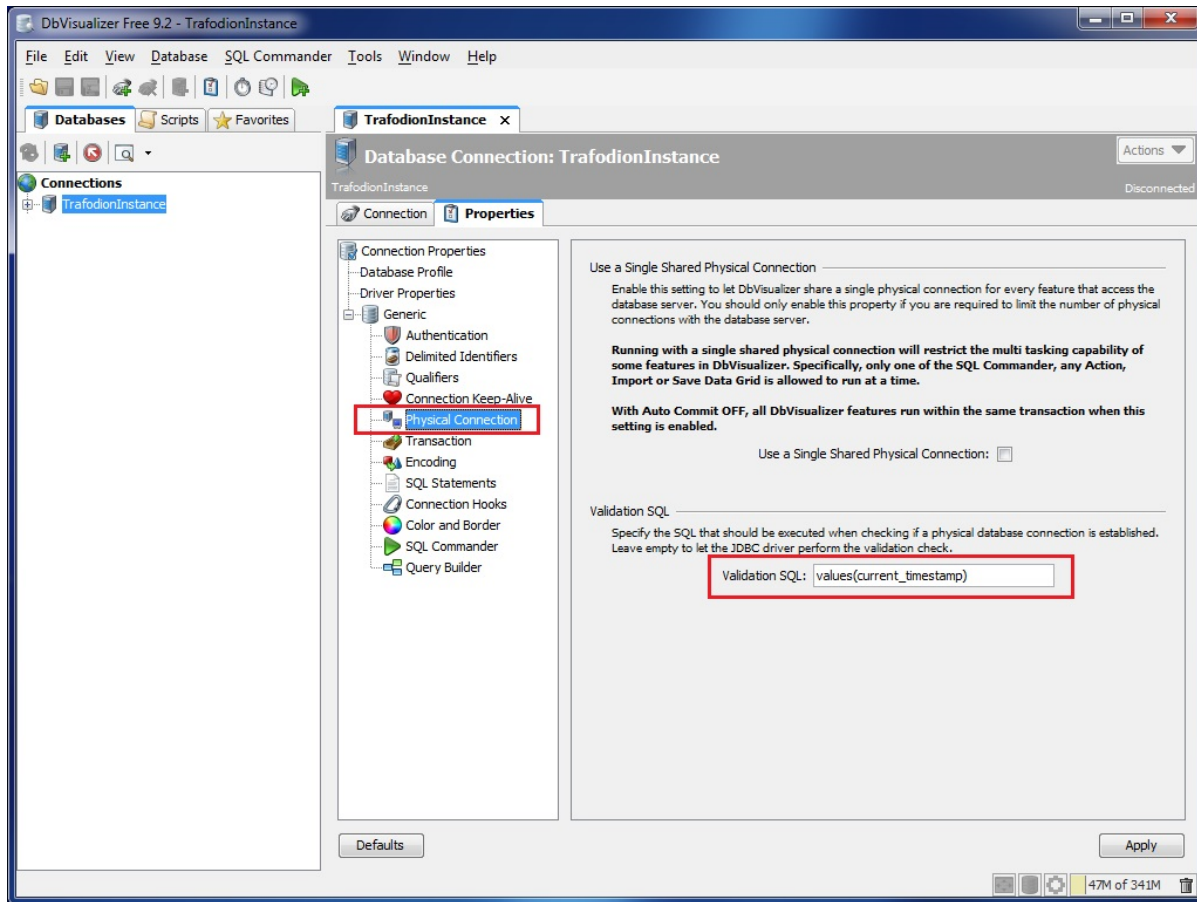
```
dbvis.generic.-ConnectionValidationSelect=disabled
```

DbVisualizer 9.2 (or a later version)

Set the **Physical Connection** property to keep your connections alive. Follow these steps:

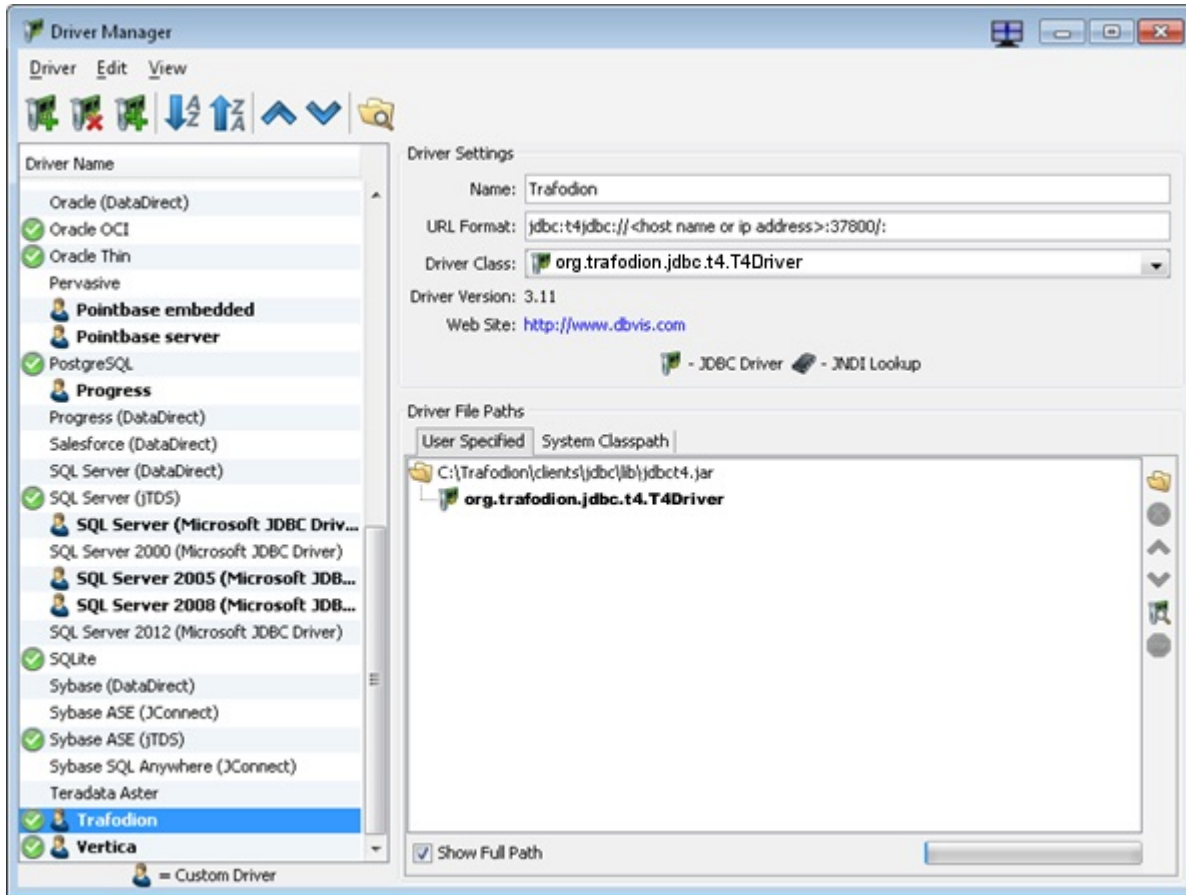
1. Double-click the database connection, select the **Properties** tab.
2. In the left navigation tree, expand the **Generic** connection properties, and select **Physical Connection**.
3. Under **Validation SQL**, enter `values(current_timestamp)` and click **Apply**.

Client Installation Guide



5.2.2. Register JDBC Type-4 Driver

Use the DbVisualizer Driver Manager and register the Trafodion JDBC Type-4 Driver.



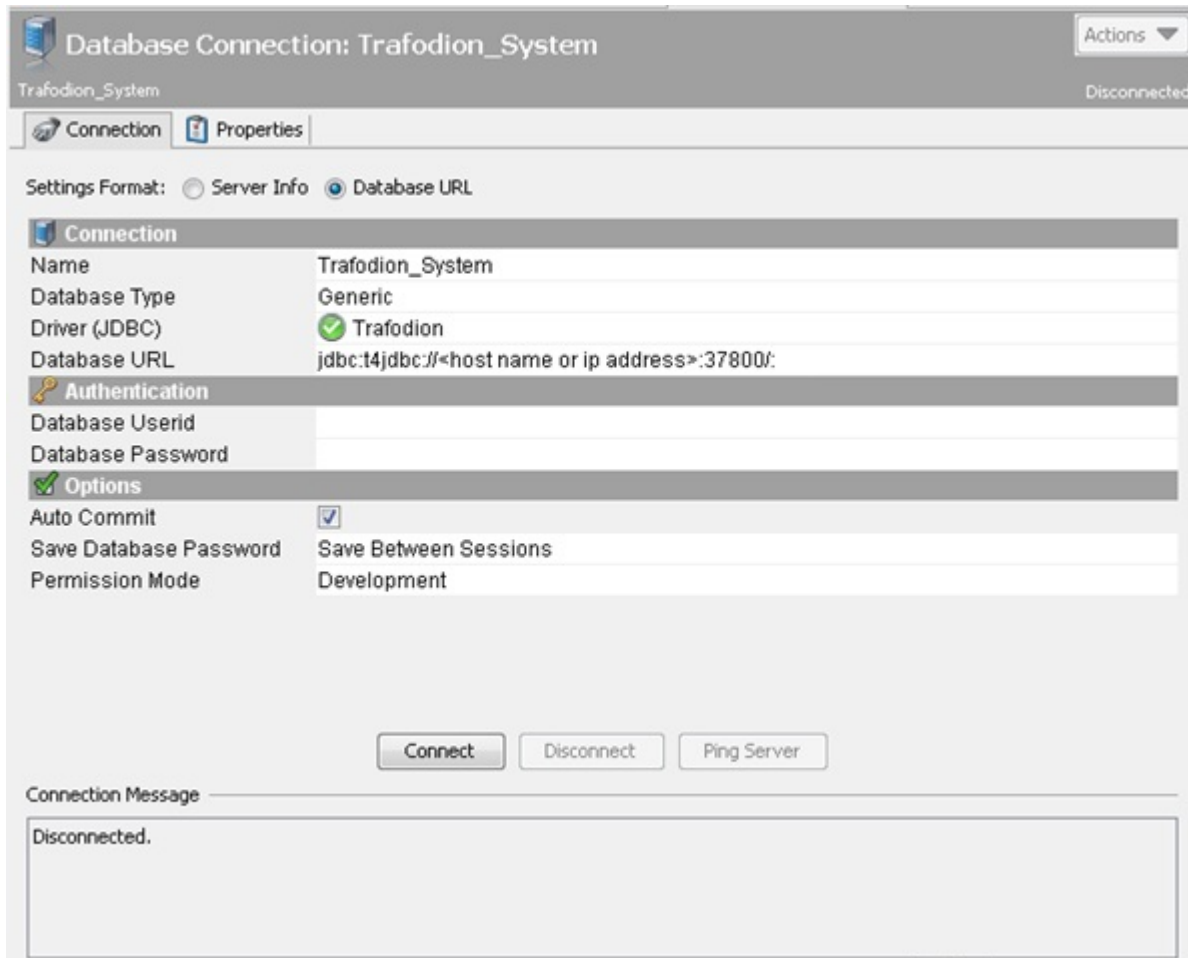
- Use the Open File icon and locate the Trafodion JDBC Type-4 Driver.
- Use the JDBC URL format:

```
jdbc:t4jdbc://<host-name or ip-address>:23400/:
```

NOTE: This example uses a modified port number (**37800**) rather than the default port number (**23400**).

5.2.3. Connect to Trafodion

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; for example: `jdbc:t4jdbc://<host-name or ip-address>:37800/:` (default is: **23400**).



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

Chapter 6. Configure SQuirreL Client

These configuration instructions apply to Trafodion Release 1.3.*n* and subsequent updates of Trafodion until otherwise indicated.

6.1. Prerequisite Software

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

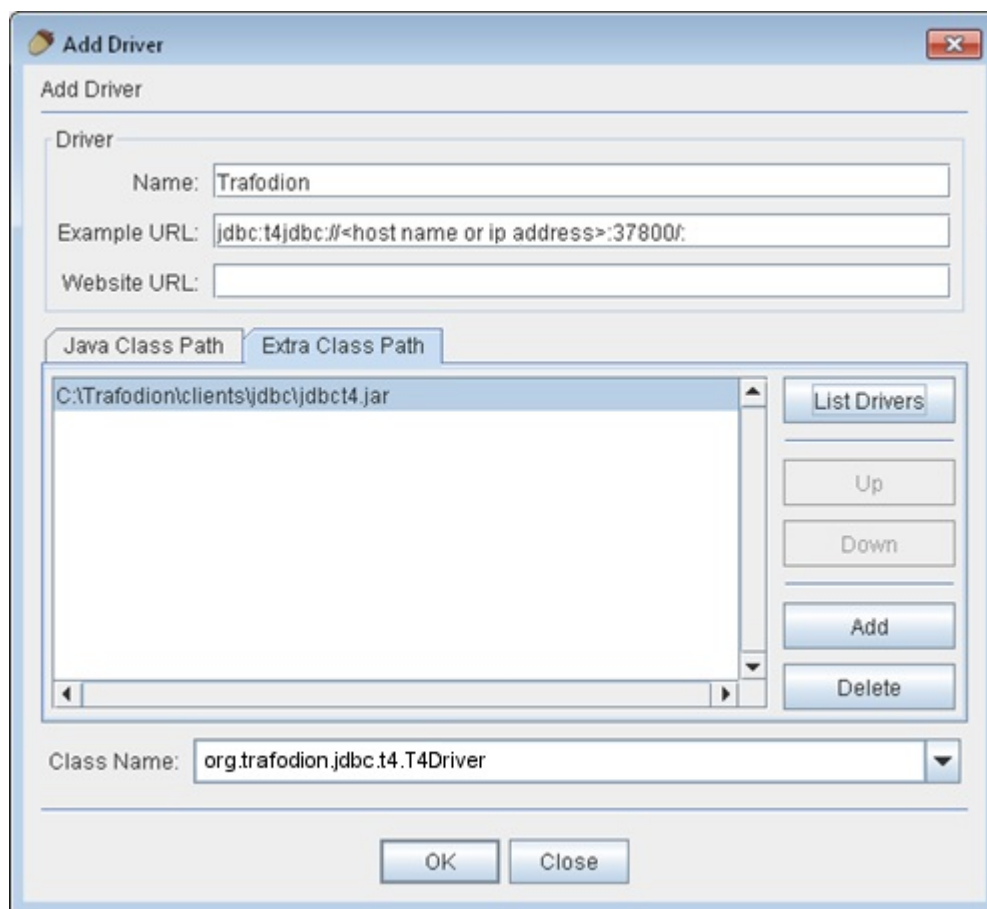
- Java Runtime Environment (JRE) 1.7 or higher. See [Java Environment](#) in the [Install JDBC Type-4 Driver](#) chapter above.
- Trafodion JDBC Type-4 Driver. See [Install JDBC Type-4 Driver](#) above.
- SQuirreL SQL Client 3.5.0. See the [SQuirreL SQL Client website](#).

6.2. Configuration Instructions

6.2.1. Register JDBC Type-4 Driver

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

1. Click on the **Add** button and locate the Trafodion JDBC Type-4 Driver.
2. Click on the **List Drivers** button to find the JDBC Driver Class Name.
3. Set the properties as shown below:



- Name: Trafodion
- Example URL: `jdbc:t4jdbc://host-name or ip-address:37800/`: (Default port number: **23400**)

6.2.2. Connect to Trafodion

Use the Add Alias dialog box and create an alias for your Trafodion System:

The screenshot shows the 'Add Alias' dialog box with the following fields and options:

- Name:** Trafodion_System
- Driver:** Trafodion (selected in a dropdown menu)
- URL:** jdbc:t4jdbc://<host name or ip address>:37800/
- User Name:** (empty field)
- Password:** (empty field)
- Auto logon
- Connect at Startup
- Properties** button
- Warning - Passwords are saved in clear text
- OK**, **Close**, and **Test** buttons at the bottom.

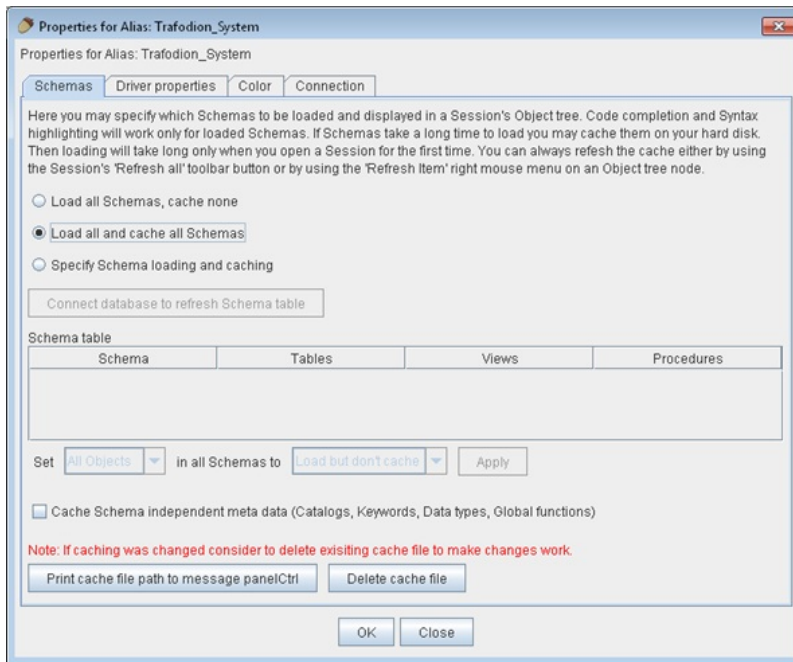
1. Edit the connection **URL** to match your Trafodion system's host name and port number:

Example

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

2. Click on the **Properties** button for the alias.

3. 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.

Chapter 7. Install Linux ODBC Driver

7.1. 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.x or CentOS 6.x.

7.2. Installation Instructions

You download and extract the Trafodion client package using the instructions in [Download Installation Package](#) above.

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

```
connect_test.cpp
install.sh
libicudataNv44.so.44
libicuucNv44.so.44
libtrafodbc_l64.so
libtrafodbc_l64_drvr.so
LICENSE
license.txt
MD5SUM
TRAFDSN
```

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

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



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, [Setting Up the Client Environment](#).

7.2.1. Install/Reinstall Linux ODBC Driver



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  
sudo ./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 data-source 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`).

7.2.2. Set Up 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 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:

```
sudo yum -y install libiodbc libiodbc-devel
```

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

7.2.3. Enable 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

7.2.4. Use Third-Party Driver Manager



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

- If you are using an external driver manager, then you must point to `libtrafodbc_drvr64.so` and not to `libtrafodbc64.so`.
- The driver, `libtrafodbc_164_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`).

7.2.5. Run Sample Program (`connect_test`)



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>
```

Example

```
export LD_LIBRARY_PATH=/usr/lib64
```

3. In the `/etc/odbc/TRAFODSN` file, add the correct IP address to the `Server` parameter for the `Default_DataSource`.

Example

```
[Default_DataSource]
Description = Default Data Source
Catalog = TRAFODION
Schema = SEABASE
DataLang = 0
FetchBufferSize = SYSTEM_DEFAULT
Server = TCP:1.2.3.4:23400 <- _Set IP Address_
SQL_ATTR_CONNECTION_TIMEOUT = SYSTEM_DEFAULT
SQL_LOGIN_TIMEOUT = SYSTEM_DEFAULT
SQL_QUERY_TIMEOUT = NO_TIMEOUT
```

4. Compile the sample program.

```
sudo 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...
```

7.2.6. Run Sample Program (basicsql)



The Basic SQL sample program is not currently bundled with the ODBC Linux driver. To obtain the source code for this program, see [basicsql \(Sample ODBC Program\)](#).

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/TRAFODSN` file, add the correct IP address to the `Server` parameter for the `Default_DataSource`. For example:

Example

```
[Default_DataSource]
Description = Default Data Source
Catalog = TRAFODION
Schema = SEABASE
DataLang = 0
FetchBufferSize = SYSTEM_DEFAULT
Server = TCP:1.2.3.4:23400
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
Data inserted.
Fetching data using SQLExecDirect, SQLFetch, SQLGetData
Data selected: 1000 CREATE REPORTS 2014-3-22
Basic SQL ODBC Test Passed!
```

Chapter 8. Install odb

8.1. 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](#).

8.2. Installation Instructions



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](#).

You download and extract the Trafodion client package using the instructions in [Download Installation Package](#) above.

8.2.1. Install 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:

```
mkdir $HOME/odb
tar -xzf odb64_linux.tar.gz -C $HOME/odb
```

The command extracts these files:

- `README`
 - `/bin/odb64luo` (the odb executable)
3. You are now ready to run the odb executable. For more information, see the [Trafodion odb User Guide](#).

8.3. Uninstall odb

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

```
rm -rf odb-installation-directory
```

Chapter 9. Install Windows ODBC Driver



License issues prevent us from including the ODBC Driver for Windows in this release. Contact the Trafodion user e-mail list (user@trafodion.incubator.apache.org) for help obtaining the driver.

9.1. Installation Requirements

Item	Requirement
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, Windows 10, or Windows Server 2008
Network Software	TCP/IP

9.2. Installation Instructions

9.2.1. Install Windows ODBC Driver



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

You download and extract the Trafodion client package using the instructions in [Download Installation Package](#) above.

The ODBC client installation file, `TFODBC64-* .exe`, which 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.

By default, a new version of the ODBC driver is 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 <i>version</i> Windows 64-bit

9.2.2. 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. Please refer to [Interactive Mode Installation](#) below.

Interactive Mode Installation

1. Double-click the `TFODBC64-* .exe` distribution file to start the InstallShield wizard.
2. On the **Welcome** page, click **Next**.
3. Read and select the **I accept the agreement** radio button. Click **Next**.
4. On the **Destination Folder** page, click **Install** to select the default location: `C:\Program Files\Trafodion\TRAFODBC version\`

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

5. Read and accept the Microsoft C++ license agreement by checking the **I agree to the license terms and conditions**. Click **Install**.
6. Click **Close**.
7. Click **Finish** to exit the installation wizard.

9.2.3. Add a client data source

1. Start the Microsoft ODBC Administrator:
 - On Windows 7: **Start>All Programs>Trafodion ODBC *version*>MS ODBC Administrator**
 - On Windows 8: Right-click the **Trafodion ODBC *version*** icon on the desktop and select MS ODBC Administrator.
 - On Windows 10: Right-click the Windows icon in the menu bar. Select **Settings**. Search for **Set up ODBC data sources (64-bit)**. Click on the found item.
2. In the **ODBC Data Source Administrator** dialog box, click **Add**.
3. Select **TRAF ODBC *version***, 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 default port number as **23400**¹. 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 **Finish**.

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

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

¹ Your specific installation may use a different port number. Check with your Trafodion administrator.

9.2.4. Set Up Client Environment

All client data sources connect to the pre-configured server data source on the database platform, which is `Default_DataSource`.

You can configure one data source only, `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**.
 - On Windows 7: **Start>All Programs>Trafodion ODBC version>MS ODBC Administrator**
 - On Windows 8: Right-click the **Trafodion ODBC version** icon on the desktop and select MS ODBC Administrator.
 - On Windows 10: Right-click the Windows icon in the menu bar. Select **Settings**. Search for **Set up ODBC data sources (64-bit)**. Click on the found item.
2. In the **ODBC Data Source Administrator** dialog box, select the **User DSN** tab, and click **Add**.
3. Select the **TRAF ODBC version** 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. By default, the port number is **23400**¹. Click **Next** to continue.
6. Select the default schema. If you do not select a schema, the default is `SEABASE`. 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 **Finish** to continue.

The **Test Trafodion ODBC Connection** dialog box appears, allowing you to test the connection using the data source that you created.

8. Click **Test Connection**.
9. When prompted, enter your user name and password, and, optionally, schema. Click **OK**.

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

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

¹ Your specific installation may use a different port number. Check with your Trafodion administrator.

9.2.5. Enable 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.
 - On Windows 7: **Start>All Programs>Trafodion ODBC version>MS ODBC Administrator**
 - On Windows 8: Right-click the **Trafodion ODBC version** icon on the desktop and select MS ODBC Administrator.
 - On Windows 10: Right-click the Windows icon in the menu bar. Select **Settings**. Search for **Set up ODBC data sources (64-bit)**. Click on the found item.
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, please refer to [Setting Up the Client Environment](#).

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.

9.2.6. Run Sample Program (basicsql)



The Basic SQL sample program is not currently bundled with the ODBC Windows driver. To obtain the source code and the build and run files for this program, please refer to [ODBC Sample Program](#).

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, please refer to [Set Up Client Environment](#).
5. From the command prompt, run the sample program by entering run or this command:

```
basicsql DefaultDataSource <username> <password>
```

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

Example

```
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!
```

9.3. Reinstall Windows ODBC Driver

To reinstall the driver, we recommend that you fully remove your ODBC driver and then install the new version. Please refer to [Uninstalling the Trafodion ODBC Driver for Windows](#) and then [Installing the Trafodion ODBC Driver for Windows](#).

9.4. Uninstalling Windows ODBC Driver

1. Start to remove the ODBC driver:

- On Windows 7: **Start>All Programs>Trafodion ODBC *version*>Remove TRAF ODBC *version***
- On Windows 8: Right-click the **Trafodion ODBC *version*** icon on the desktop and select **Remove TRAF ODBC *version***.
- On Windows 10: Right-click the Windows icon in the menu bar. Select **Control Panel**. Click on **Uninstall a program**. Locate **Trafodion ODBC64 *version*** and select it. Click on **Uninstall**.

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

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

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



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

Chapter 10. Appendix

10.1. basicsql (Sample ODBC Program)

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 [Windows Build and Run Files for 'basicsql'](#).

Copy and paste the code from this appendix into the recommended files. To build and run the sample program, see these instructions:

- On Windows: [Run Sample Program \(basicsql\)](#).
- On Linux: [Run Sample Program \(basicsql\)](#).

10.1.1. basicsql.cpp Source Code

You can download the `basicsql.cpp` example from

http://trafodion.incubator.apache.org/docs/client_install/resources/source/basicsql.cpp.

Alternatively, copy and paste the following code into a file named `basicsql.cpp`:

```
// @@@ START COPYRIGHT @@@
//
// Licensed to the Apache Software Foundation (ASF) under one
// or more contributor license agreements. See the NOTICE file
// distributed with this work for additional information
// regarding copyright ownership. The ASF licenses this file
// to you 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:"

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);
}
```

10.1.2. Windows Build and Run Files for `basicsql`

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 [Run Sample Program \(basicsql\)](#).

`build.bat` (Build Script)

You can download the `build.bat` example from

http://trafodion.incubator.apache.org/docs/client_install/resources/source/build.bat.

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


```

@echo off
REM @@@ START COPYRIGHT @@@
REM
REM Licensed to the Apache Software Foundation (ASF) under one
REM or more contributor license agreements. See the NOTICE file
REM distributed with this work for additional information
REM regarding copyright ownership. The ASF licenses this file
REM to you under the Apache License, Version 2.0 (the
REM "License"); you may not use this file except in compliance
REM with the License. You may obtain a copy of the License at
REM
REM http://www.apache.org/licenses/LICENSE-2.0
REM
REM Unless required by applicable law or agreed to in writing,
REM software distributed under the License is distributed on an
REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
REM KIND, either express or implied. See the License for the
REM specific language governing permissions and limitations
REM under the License.
REM
REM @@@ END COPYRIGHT @@@

CL /c /Zi /nologo /W3 /WX- /O2 /D "NDEBUG" /D "_CRT_SECURE_NO_DEPRECATED" /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" "odbc32.lib" /TLBID:1 /DYNAMICBASE /NXCOMPAT /MACHINE:X64
/ERRORREPORT:QUEUE basicsql.obj

```

To build the sample program on Windows, see the instructions in [Run Sample Program \(basicsql\)](#).

Run run.bat

You can download the run.bat example from

http://trafodion.incubator.apache.org/docs/client_install/resources/source/run.bat.

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

```
@echo off
REM @@@ START COPYRIGHT @@@
REM
REM Licensed to the Apache Software Foundation (ASF) under one
REM or more contributor license agreements. See the NOTICE file
REM distributed with this work for additional information
REM regarding copyright ownership. The ASF licenses this file
REM to you under the Apache License, Version 2.0 (the
REM "License"); you may not use this file except in compliance
REM with the License. You may obtain a copy of the License at
REM
REM http://www.apache.org/licenses/LICENSE-2.0
REM
REM Unless required by applicable law or agreed to in writing,
REM software distributed under the License is distributed on an
REM "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY
REM KIND, either express or implied. See the License for the
REM specific language governing permissions and limitations
REM under the License.
REM
REM @@@ END COPYRIGHT @@@

basicsql.exe Default_Datasource user1 pwd1
```

To run the sample program on Windows, see the instructions in [Run Sample Program \(basicsql\)](#).