

Objectivity/SQL++ ODBC Driver User's Guide

Release 10.2

Objectivity/SQL++ ODBC Driver User's Guide

Part Number: 10.2-ODBC-0

Release 10.2, October 13, 2011

The information in this document is subject to change without notice. Objectivity, Inc. assumes no responsibility for any errors that may appear in this document.

Copyright 1993–2011 by Objectivity, Inc. All rights reserved. This document may not be copied, photocopied, reproduced, translated, or converted to any electronic or machine-readable form in whole or in part without prior written approval of Objectivity, Inc.

Objectivity and Objectivity/DB are registered trademarks of Objectivity, Inc. Active Schema, Objectivity/DB Active Schema, Assist, Objectivity/Assist, ooAssistant, Objectivity/DB ooAssistant, Objectivity/DB Fault Tolerant Option, Objectivity/FTO, Objectivity/DB Data Replication Option, Objectivity/DRO, Objectivity/DB High Availability, Objectivity/HA, Objectivity/DB Hot Failover, Objectivity/DB In-Process Lock Server, Objectivity/IPLS, Objectivity/DB Open File System, Objectivity/OFS, Objectivity/DB Parallel Query Engine, Objectivity/PQE, Objectivity/DB Persistence Designer, Objectivity/DB Secure Framework, Objectivity/Secure, Objectivity/C++, Objectivity/C++ Data Definition Language, Objectivity/DDL, Objectivity/Dashboard, Objectivity/C++ Active Schema, Objectivity/C++ Standard Template Library, Objectivity/C++ STL, Objectivity/C++ Spatial Index Framework, Objectivity/Spatial, Objectivity for Java, Objectivity/.NET, Objectivity/.NET for C#, Objectivity/Python, Objectivity/Smalltalk, Objectivity/SQL++, Objectivity/SQL++ ODBC Driver, Objectivity/ODBC, Objectivity Event Notification Services, and Persistence Designer are trademarks of Objectivity, Inc.

Other trademarks and products are the property of their respective owners.

ODMG information in this document is based in whole or in part on material from *The Object Database Standard: ODMG 2.0*, edited by R.G.G. Cattell, and is reprinted with permission of Morgan Kaufmann Publishers. Copyright 1997 by Morgan Kaufmann Publishers.

The software and information contained herein are proprietary to, and comprise valuable trade secrets of, Objectivity, Inc., which intends to preserve as trade secrets such software and information. This software is furnished pursuant to a written license agreement and may be used, copied, transmitted, and stored only in accordance with the terms of such license and with the inclusion of the above copyright notice. This software and information or any other copies thereof may not be provided or otherwise made available to any other person.

RESTRICTED RIGHTS NOTICE: Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the Objectivity, Inc. license agreement and as provided in DFARS 227.7202-1(a) and 227.7202-3(a) (1998), and FAR 12.212, as applicable. Objectivity, Inc., 640 West California Avenue, Suite 210, Sunnyvale, CA 94086-3624.

Contents

Getting Help	5
How to Reach Objectivity Customer Support Before You Call	5 5
Chapter 1 Overview	7
About Objectivity/DB Databases	7
About ODBC Connections	8
Summary of Steps for Using Objectivity/ODBC	10
Chapter 2 Installation on Windows	11
System Requirements	11
Installing Objectivity/ODBC	12
Preparing to Use Objectivity/ODBC	13
Adding Objectivity/DB Data Sources	13
Configuring TCP/IP	15
Testing Objectivity/ODBC	16
Chapter 3 Installation on UNIX	19
System Requirements	19
Installing Objectivity/ODBC	20
Preparing to Use Objectivity/ODBC	21
Checking and Registering the Driver	22
Adding Objectivity/DB Data Sources	23
Configuring TCP/IP	25
Testing Objectivity/ODBC	26
Index	29

Getting Help

We have done our best to make sure all the information you need to install and operate each product is provided in the product documentation. However, we also realize problems requiring special attention sometimes occur.

How to Reach Objectivity Customer Support

You can contact Objectivity Customer Support by:

- **Telephone:** Call 1.408.992.7100 *or* 1.800.SOS.OBJY (1.800.767.6259) Monday through Friday between 6:00 A.M. and 6:00 P.M. Pacific Time, and ask for Customer Support.

The toll-free 800 number can be dialed *only* within the 48 contiguous states of the United States and Canada.

- **Fax:** Send a fax to Objectivity at 1.408.992.7171.
- **Electronic Mail:** Send electronic mail to help@objectivity.com.

Before You Call

If you need help from Customer Support, please have the following information ready before you contact Objectivity:

- Your name, company name, address and telephone number, fax number, and email address
- Description of your workstation environment, including the type of workstation, operating system version, compiler or interpreter, and windowing environment
- Information about the Objectivity product you are using, including the version of the Objectivity/DB libraries
- Detailed description of the problem you have encountered

Overview

This book describes how to install and use the Objectivity/SQL++ ODBC Driver (Objectivity/ODBC) with client applications such as Microsoft Access, which conform to the Microsoft Open Database Connectivity (ODBC) interface. Objectivity/ODBC enables ODBC-compliant applications to connect to an Objectivity/SQL++ ODBC server, which in turn connects to an Objectivity/DB federated database.

NOTE Objectivity/ODBC supports ODBC 3.0 and must be used with an Objectivity/SQL++ ODBC server that implements ODBC 3.0.

About Objectivity/DB Databases

In an Objectivity/DB system, data for a particular application is normally stored in one or more individual databases that are unified in a single *federated database*. A federated database provides a consistent logical view of the data, even when the component databases are physically distributed among multiple hosts on a network of computers. Physically, a federated database is represented by two files:

- A *system-database file*, which contains the first (or only) database in the federation
- A *boot file*, which contains information about the federated database, including its location

Objectivity/DB applications and tools locate and open a federated database by specifying its boot file.

About ODBC Connections

You can access the data in an Objectivity/DB federated database through your ODBC-compliant application. To do so, you must enable your application to make an ODBC connection to the federated database. An ODBC connection is made through the following components:

- An Objectivity/SQL++ ODBC server (*ODBC server*)—an executable that accepts the ODBC connection (possibly over a network), parses the SQL statements it receives, and translates these statements into Objectivity/DB operations.
- An Objectivity/SQL++ ODBC Driver (Objectivity/ODBC)—a DLL or shared library that forwards SQL statements from the application to the correct ODBC server.
- An ODBC driver-manager—a utility used by the application to load (and unload) the correct ODBC driver. (This component is provided transparently on Windows; it is separately installed third-party software on UNIX.)

Figure 1-1 illustrates these components.

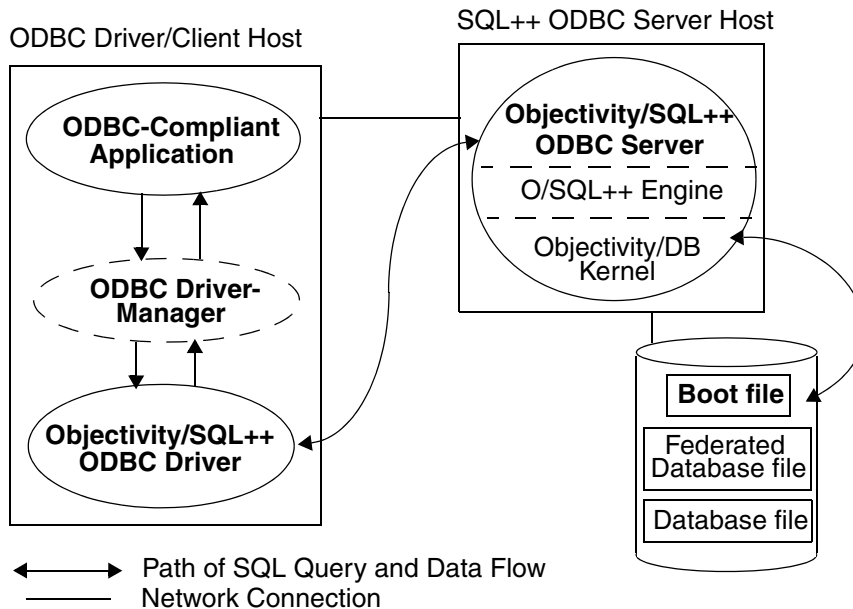


Figure 1-1 How an ODBC-compliant Application Accesses Objectivity/DB

The connection can be made when you add an Objectivity/SQL++ *data source* that describes, among other things, where to find the federated database's boot file and where to find the ODBC server that will access it.

When an ODBC-compliant application submits a SQL request to access an Objectivity/SQL++ data source, the application consults the ODBC driver-manager to determine which ODBC driver to use—in this case, Objectivity/ODBC. The ODBC driver then inspects the data source to determine which ODBC server to use, and forwards the SQL request to that server. The ODBC driver also forwards the name and location of the federated database's boot file.

Upon receiving a request, the ODBC server uses its *Objectivity/SQL++ engine* to parse the SQL statements and translate them into Objectivity/DB operations. These operations are then performed by the *Objectivity/DB kernel*, which actually opens the specified federated database and obtains the requested data.

You install Objectivity/ODBC and add data sources on the same *host* (machine) as your ODBC-compliant application. However, the ODBC server may be running on the same or a different host from the driver, and the federated database's files may be on the same or a different host from the ODBC server. Multiple ODBC servers running on different machines may access the same Objectivity/DB federated database; you can define different data sources for connecting to the same federated database through different ODBC servers.

Servers, drivers, and federated databases must all be on a network that uses the *TCP/IP network protocol* (a service that allows machines and processes on a network to communicate with each other). Objectivity/DB uses this protocol.

Summary of Steps for Using Objectivity/ODBC

To use Objectivity/ODBC, you or your system administrator must perform the following steps:

1. Confirm that the Objectivity/SQL++ ODBC server is properly configured and running. See *Objectivity/SQL++* book or the chapter on Objectivity/SQL++ installation in the *Installation and Platform Notes* for the appropriate platform.

WARNING

If the Objectivity/SQL++ ODBC server is not properly configured and running, your ODBC-compliant application will be unable to find the federated database.

2. Install the Objectivity/SQL++ ODBC Driver on the same host where you plan to use the ODBC-compliant application; see Chapter 2, “Installation on Windows,” and Chapter 3, “Installation on UNIX,” in this book.
3. Create a data source corresponding to the federated database you need to access; see “Adding Objectivity/DB Data Sources” on page 13 and on page 19.
4. Use the ODBC-compliant application:
 - a. Start the application.
 - b. Connect to the data source that you have just configured.
 - c. Submit your queries.

Installation on Windows

This chapter describes the requirements and steps for installing the Objectivity/SQL++ ODBC Driver (Objectivity/ODBC) on Windows platforms.

System Requirements

You can install Objectivity/ODBC on the Windows platforms listed in Table 2-1.

Table 2-1: Supported Windows Architectures for Objectivity/ODBC

Hardware	Operating System ^a	Abbreviation	Addressing Mode
x86	Windows XP Professional	Windows XP	32-bit
	Windows Server 2003	Windows 2003	32-bit
	Windows Server 2008	Windows 2008	32-bit
	Windows 7	Windows 7	32-bit
x86_64	Windows XP Professional x64	Windows XP x64	64-bit
	Windows Server 2008 R2	Windows 2008	64-bit
	Windows 7 x64	Windows 7 x64	64-bit

- a. See the Objectivity Technical Support Web site for the currently supported versions of the operating system. Contact Objectivity Customer Support to get access to this Web site.

Software Requirements

Objectivity/ODBC requires that the following software be installed on each computer that is to run an ODBC-compliant client application:

- Winsock-compatible TCP/IP software
- Microsoft ODBC Administrator

Microsoft TCP/IP and Microsoft ODBC Administrator are included with Windows operating systems.

Objectivity/ODBC requires that an Objectivity/SQL++ ODBC server be installed in the same network. The Objectivity/SQL++ ODBC server must implement ODBC 3.0.

A C++ development environment is required for building and running the sample application that verifies Objectivity/ODBC installation.

Installing Objectivity/ODBC

To install Objectivity/ODBC:

1. Verify that required software has been completely and correctly installed. See "Software Requirements" on this page.
2. Start the setup program (`setup.exe`) for installing Objectivity products. If you have an Objectivity distribution CD, place it in your computer's CD-ROM drive to start the setup program automatically. If the program fails to start, navigate to your CD-ROM drive and double-click on `setup.exe`
3. If any other Objectivity products are already installed on your computer, click **Modify**, click **Next**, and continue with step 4. Otherwise, if this is the first (or only) Objectivity product you are installing, select the directory (`installDir`) in which to install Objectivity/ODBC. Click **Next** and continue with step 4.
4. Select **Objectivity/SQL++ ODBC Driver**.

NOTE You must be licensed for every product you install.

5. Click **Next** and follow the prompts to complete the installation.

Preparing to Use Objectivity/ODBC

After the setup program completes, perform the following steps:

1. Familiarize yourself with your installation by reading “What the Setup Program Does” on page 13.
2. Add the desired Objectivity/DB federated databases as data sources. Follow the steps in “Adding Objectivity/DB Data Sources” on page 13.
3. Configure TCP/IP to enable Objectivity/ODBC to communicate with an Objectivity/SQL++ ODBC server. Follow the steps in “Configuring TCP/IP” on page 15.
4. (Optional) Test Objectivity/ODBC with an Objectivity/SQL++ ODBC server. Follow the steps in “Testing Objectivity/ODBC” on page 16.

What the Setup Program Does

The Objectivity/ODBC setup program:

- Installs files in subdirectories of the Objectivity/ODBC installation directory *installDir*, as shown in Table 2-2.

Table 2-2: Objectivity/ODBC Release Files in *installDir*

Subdirectory	Contains
bin	DLL for the ODBC driver
doc\odbcDriver.pdf	PDF file for the online book <i>Objectivity/SQL++ ODBC Driver User's Guide</i>
include	Include files for developing custom ODBC-compliant applications
samples\sql\odbc	Application for demonstration and testing

- Makes the driver available to the Microsoft ODBC Administrator.

Adding Objectivity/DB Data Sources

You enable ODBC-compliant client applications to access an Objectivity/DB federated database by adding a *data source* for it. In general, a data source identifies the data to be accessed and the means of accessing it (for example, host and network information). The data source you add for a federated database specifies its boot file and a host running a Objectivity/SQL++ ODBC server.

Before you add a data source for a federated database:

- Identify the Objectivity/SQL++ ODBC server that will access the federated database and obtain:
 - The TCP/IP name of the computer that runs the ODBC server.
 - The service name under which the ODBC server is running.
- Obtain the location and name of the federated database's boot file.
- Ensure that you have a valid user account on the host running the Objectivity/SQL++ ODBC server.

To add a data source for a federated database:

1. In Control Panel on the host where you installed Objectivity/ODBC (the driver), open Administrative Tools and click **Data Sources (ODBC)**.
(*Windows 7*) In Control Panel, open **System and Maintenance** (or **System and Security**), open **Administrative Tools**, and double-click **Data Sources (ODBC)**.
2. In ODBC Data Source Administrator, click **System DSN** (or **User DSN**, if you want to create a personal data source).
3. Click **Add**, select **Objectivity ODBC Driver** in the Create New Data Sources dialog, and click **Finish**.
4. In the resulting dialog, fill in the following fields and then click **OK**:

Data Source Name	A string that uniquely identifies the data source. This string will appear in the list of data sources in the connection dialog. The string may not exceed 32 characters.
Description	(Optional) A description of the data source.
Host	The TCP/IP name of the computer running the Objectivity/SQL++ ODBC server.
Database	The location and name of the boot file for the federated database, expressed as a fully qualified pathname beginning with a drive letter.
User ID	The username of your Objectivity/SQL++ account. This is normally your login account on the ODBC server host, provided that this account has been granted access rights to tables in the federated database (see your Objectivity/SQL++ database administrator). The Objectivity/SQL++ database administrator account (<code>sysupe</code>) has access to all tables.
Password	Password for the account you entered in the User ID field. The password you enter is <i>not</i> encoded before it is sent across the network.
Service	The service name of the ODBC server on its host (<code>oosqlnw</code>).

NOTE The **Host**, **Database**, and **Service** fields together must not exceed a string-length of 249 characters.

5. Click **OK**.
6. If you want to add other data sources, repeat steps 3 through 5.
7. When you are finished adding data sources, click **OK**.

Configuring TCP/IP

You or your system administrator must perform the following steps to enable your ODBC-compliant application to communicate with the Objectivity/SQL++ ODBC server across the network.

Identifying the ODBC Server's Host to TCP/IP

TCP/IP must be able to recognize the hostname you specify when you register a data source. That is, TCP/IP must be able to convert the hostname into an Internet address. Many sites use the TCP/IP `hosts` file to map hostnames to Internet addresses, although some sites use domain name servers for this purpose.

You should verify that the computer on which you installed Objectivity/ODBC recognizes a valid hostname for the computer on which the ODBC server is running. For example, to verify that your computer recognizes `hostname`, you can:

- Enter the following command at a command prompt on your computer:
`ping hostname`

Specifying the ODBC Server's Port Number

As installed, the Objectivity/SQL++ ODBC server and the Objectivity/SQL++ ODBC Driver communicate through a default TCP/IP port. If the ODBC server has been assigned a nondefault port number (for example, due to a port conflict), you must register the new port number with TCP/IP on each Objectivity/ODBC (driver) host.

To register the ODBC server's port number with TCP/IP:

1. Find the TCP port number assigned to the `oosqlnw` service on the host running the Objectivity/SQL++ ODBC server. For example, use the Objectivity Network Services tool on the ODBC server host.

2. On the host where you installed Objectivity/ODBC (the driver), open the TCP/IP `services` file. The location of this file depends on the TCP/IP vendor. The Microsoft TCP/IP file location is:
`%systemroot%\system32\drivers\etc\services`
3. Add the following entry to the TCP/IP `services` file, if such an entry does not already exist:
`oosqlnw portNumber/tcp # Objectivity/SQL++ server`
 where `portNumber` is the TCP port number you found in step 1.

Testing Objectivity/ODBC

You can verify the correct operation of Objectivity/ODBC in combination with an Objectivity/SQL++ ODBC server. The following test (which requires a C++ development environment) compiles and links a sample ODBC application and uses that application to access a demo federated database. The demo federated database is provided with Objectivity/SQL++ on the server host.

The sample application is a C++ application that calls ODBC 3.0 functions to query and modify a federated database. You can inspect the sample to see how such an application is compiled and linked.

To build and run the sample ODBC-compliant application:

1. Verify that the demo federated database and the Objectivity/SQL++ ODBC server have been set up for this test:
 - If your ODBC server runs on Windows, see “Preparing the ODBC Server for Testing” in Chapter 9 in *Installation and Platform Notes for Windows*.
 - If your ODBC server runs on UNIX, see “Preparing the ODBC Server for Testing” in Chapter 8 in *Installation and Platform Notes for UNIX*.

The Objectivity/SQL++ ODBC server must be running.

2. On the Objectivity/ODBC (driver) host, add a data source for the demo federated database.
3. Go to the Objectivity/SQL++ ODBC samples directory—for example, in a command window, enter:
`cd installDir\samples\sql\odbc`
 You may wish to make a backup copy of this directory.
4. Check whether the lock server is running; start it, if necessary.
5. Build and run the sample application. At the command prompt, enter:
`nmake`

If the Objectivity/SQL++ ODBC programming interface is set up correctly, you will see messages like these:

```
Welcome to the Objectivity/SQL++ ODBC demo
...
Building the ooODBCdemo executable
...
Running the demo...
Demo complete.
Comparing the results.
Test PASSED -- The expected results were achieved.
No errors.
```


Installation on UNIX

This chapter describes the requirements and steps for installing the Objectivity/SQL++ ODBC Driver (Objectivity/ODBC) on UNIX platforms.

System Requirements

You can install Objectivity/ODBC on the UNIX platform listed in Table 3-1.

Table 3-1: Supported UNIX Platform for Objectivity/ODBC

Architecture Name	Hardware
hprisc	HP 9000 Series 700/800
hpuxia64	Intel Itanium 2
linux86gcc3	x86 (586 or greater)
linux86_64	x86_64
solaris7	Sun SPARC
solaris86_64	x86_64
sparc64	Sun SPARC

NOTE See the release notes on the Objectivity Technical Support Web site for the currently supported operating system versions. Contact Objectivity Customer Support to get access to this Web site.

Software Requirements

Objectivity/ODBC requires that an Objectivity/SQL++ ODBC server be installed in the same network. The Objectivity/SQL++ ODBC server must implement ODBC 3.0.

A C++ development environment is required for building and running the sample application that verifies Objectivity/ODBC installation.

Objectivity/ODBC requires that a third-party ODBC driver-manager be installed on each UNIX workstation that is to run an ODBC-compliant client application. An ODBC driver-manager is the UNIX counterpart of the Microsoft ODBC Administrator. For convenience, you can use the ODBC driver-manager that is distributed with Objectivity/ODBC—namely, unixODBC DriverManager. This driver-manager is installed automatically when you install Objectivity/ODBC.

As distributed, unixODBC DriverManager provides a text-based user interface for registering the driver and adding data sources. If you prefer a graphical user interface, you can optionally obtain the entire unixODBC DriverManager, along with its dependencies and manuals, from the unixODBC Project Web site at www.unixodbc.org.

NOTE The steps in this chapter assume you are using the text-based unixODBC DriverManager as it is distributed and installed with Objectivity/ODBC.

Installing Objectivity/ODBC

To install Objectivity/ODBC:

1. Log in to your workstation as `root`.
2. Verify that required software has been completely and correctly installed (see “Software Requirements” on page 20).
3. If you have not done so already, obtain the Objectivity installation files—for example, by downloading them from the Objectivity Technical Support Web site.
4. Run the Objectivity installation script with a command such as the following:

```
./install.sh
```
5. When prompted, select **Custom Installation**; then select Objectivity/SQL++ ODBC Driver by entering its item number from the displayed list.

NOTE You must have a license for every product you install.

6. At the directory prompt, specify the directory (*installDir*) in which to install Objectivity/ODBC. (If Objectivity/DB is also installed on your machine, the Objectivity/DB installation directory is recommended—for example, */usr/object*.)

The installation script:

- Places the release files in subdirectories of *installDir/arch*, where *arch* is the architecture name for your platform.

Preparing to Use Objectivity/ODBC

After the installation script completes, perform the following steps:

1. Familiarize yourself with your installation by reading “Release Files for Objectivity/ODBC” on page 22.
2. Set the environment variable `OO_SQL_DIR` to the fully qualified pathname of the Objectivity/DB installation directory (*installDir/arch*). You may wish to set the variable in a startup file such as `.login`.

NOTE You must set the `OO_SQL_DIR` environment variable for every user that will interact with the provided ODBC driver-manager—for example, to list registered drivers or to add data sources. This variable enables the ODBC driver-manager to use *installDir/arch/etc/sql* as its system directory.

3. Verify that Objectivity/ODBC has been registered with the provided ODBC driver-manager. Follow the steps in “Checking and Registering the Driver” on page 22.
4. Add the desired Objectivity/DB federated databases as data sources. Follow the steps in “Adding Objectivity/DB Data Sources” on page 23.
5. Configure TCP/IP to enable Objectivity/ODBC to communicate with an Objectivity/SQL++ ODBC server. Follow the steps in “Configuring TCP/IP” on page 25.
6. (Optional) Test Objectivity/ODBC with an Objectivity/SQL++ ODBC server by browsing the provided demo federated database (see “Testing Objectivity/ODBC” on page 26).

Release Files for Objectivity/ODBC

When you install Objectivity/ODBC, its files are organized in subdirectories of the *installDir/arch* directory, as shown in Table 3-2.

Table 3-2: Release Files in *installDir/arch*

Subdirectory	Contains
bin	Tool for adding data sources and registering Objectivity/ODBC with unixODBC DriverManager
samples/sql/odbc	Application for demonstration and testing
doc	PDF file for the online book <i>Objectivity/SQL++ ODBC Driver User's Guide</i>
etc/sql	Template files used for adding data sources and registering Objectivity/ODBC with unixODBC DriverManager
include	Include files for developing custom ODBC-compliant applications
lib	Objectivity/ODBC shared library (the driver) and unixODBC shared libraries (the driver-manager)

Checking and Registering the Driver

When you install Objectivity/ODBC, it is automatically *registered* with the provided ODBC driver-manager (unixODBC DriverManager). Objectivity/ODBC must be registered so that unixODBC DriverManager can load it whenever an ODBC-compliant client application requests data from an Objectivity/DB federated database.

After the Objectivity installation script completes, you should query the provided ODBC driver-manager to check whether Objectivity/ODBC was registered correctly.

Checking Whether Objectivity/ODBC is Registered

To verify that Objectivity/ODBC is registered with unixODBC DriverManager:

1. Enter the following at a command prompt:

```
odbcinst -q -d
```
2. Look for the following driver name in the list of registered ODBC drivers:

```
ObjectivitySQL
```

3. If you do not see ObjectivitySQL listed as a registered ODBC driver:
 - a. Make sure the OO_SQL_DIR environment variable for your account is set to the fully qualified pathname of the Objectivity/DB installation directory (*installDir/arch*).
 - b. Repeat step 1.

Registering Objectivity/ODBC

If Objectivity/ODBC is not registered with unixODBC DriverManager or if you want to change some aspect of its registration, you can perform the registration steps yourself:

1. Make sure the OO_SQL_DIR environment variable for your account is set to the fully qualified pathname of the Objectivity/DB installation directory (*installDir/arch*).
2. Change your working directory to the Objectivity/SQL++ support directory.
Enter:

```
cd installDir/arch/etc/sql
```
3. Inspect the driver template file `odbcinst.ini_template`. If necessary, edit the file in a text editor—for example, to enter the correct pathname for the Objectivity/DB installation directory.
4. Register Objectivity/ODBC using the information in the driver template file.
Enter:

```
odbcinst -i -d -f odbcinst.ini_template
```
5. Check for the resulting driver file:

```
installDir/arch/etc/sql/odbcinst.ini
```

NOTE To get help for the `odbcinst` tool, run it with no options or arguments.

Adding Objectivity/DB Data Sources

You enable ODBC-compliant client applications to access an Objectivity/DB federated database by adding a *data source* for it. In general, a data source identifies the data to be accessed and the means of accessing it (for example, host and network information). The data source you add for a federated database specifies its boot file and a host running a particular Objectivity/SQL++ ODBC server.

You can add a *personal data source*, which is specific to applications running under your user account, or a *system data source*, which is available to applications running under any user account.

Before you add a data source for a federated database:

- Identify the Objectivity/SQL++ ODBC server that will access the federated database, and obtain:
 - The TCP/IP name of the host that runs the ODBC server.
 - The service name under which the ODBC server is running.
- Obtain the location and name of the federated database's boot file.
- Ensure that you have a valid user account on the host running the Objectivity/SQL++ ODBC server.

To add a data source for a federated database:

1. Log in under an appropriate user account. In particular, you must log in under your user account if you are adding a personal data source.
2. Make sure the `OO_SQL_DIR` environment variable for your account is set to the fully qualified pathname of the Objectivity/DB installation directory (*installDir/arch*).
3. Copy the provided data-source template file to a separate location and change your working directory to this new location. For example, enter the following, where *homeDir* represents your home directory:


```
cp installDir/arch/etc/sql/odbc.ini_template
   homeDir/odbc.ini_tmpl
cd
```
4. Open your copy of the data-source template file in a text editor.
5. Edit the first line so that it contains the desired data-source name enclosed in brackets—for example, `[MyObjyData]`. The data-source name should uniquely identify the data source.
6. Edit the subsequent lines so that each keyword has an appropriate value. Each keyword-value entry should consist of a single line:

```
Description = A description for the data source.

Driver       = The name under which Objectivity/ODBC is registered with the
              ODBC driver-manager—for example, ObjectivitySQL.

Host        = The TCP/IP name of the host machine running the
              Objectivity/SQL++ ODBC server (the oosqld process).

Database    = Fully qualified pathname of the federated database's boot file on
              the specified host.
```

User ID	=	The username of your Objectivity/SQL++ account. This is normally your login account on the ODBC server host, provided that this account has been granted access rights to tables in the federated database (see your Objectivity/SQL++ database administrator). The Objectivity/SQL++ database administrator account (<code>systpe</code>) has access to all tables.
Password	=	Password for the account you entered in the User ID field. The password you enter is <i>not</i> encoded before it is sent across the network.
Service	=	The service name of the ODBC server on its host (<code>oosqlnw</code>).

NOTE The Host, Database, and Service fields together must not exceed a string-length of 988 characters.

7. Save the data-source template file.
8. Add the data source using the information in the template file you edited.

To add a personal data source, enter:

```
odbcinst -i -s -f odbc.ini_template
```

To add a system data source, enter:

```
odbcinst -i -s -f odbc.ini_template -l
```

9. Verify that the data source was successfully added. Enter:

```
odbcinst -q -s
```

Adding a personal data source creates a file called `.odbc.ini` in your home directory. Adding a system data source creates a file called `installDir/arch/etc/sql/odbc.ini`.

Configuring TCP/IP

You or your system administrator must perform the following steps to enable your ODBC-compliant application to communicate with the Objectivity/SQL++ ODBC server across the network.

Identifying the ODBC Server's Host to TCP/IP

TCP/IP must be able to recognize the hostname you specify when you register a data source. That is, TCP/IP must be able to convert the hostname into an Internet address. Many sites use the TCP/IP `hosts` file to map hostnames to

Internet addresses, although some sites use domain name servers for this purpose.

You should verify that the computer on which you installed Objectivity/ODBC recognizes a valid hostname for the computer on which the ODBC server is running. For example, to verify that your computer recognizes *hostname*:

- Enter the following command at a command prompt on your computer:

```
ping hostname
```

Specifying the ODBC Server's Port Number

The Objectivity/SQL++ ODBC Driver and the Objectivity/SQL++ ODBC server must be able to communicate through the same TCP/IP port. Consequently, you must register the port number with TCP/IP on each Objectivity/ODBC (driver) host. To do so:

1. Find the TCP port assigned to the `oosqlnw` service on the host running the Objectivity/SQL++ ODBC server. The port is normally 1990.
2. On the host where you installed Objectivity/ODBC (the driver), open the TCP/IP `services` file (typically, `/etc/services`).
3. Add the following entry to the TCP/IP `services` file:

```
oosqlnw portNumber/tcp # Objectivity/SQL++ server
```

 where *portNumber* is the TCP port number you found in step 1—for example:

```
oosqlnw 1990/tcp # Objectivity/SQL++ server
```
4. Send a hang-up signal to your `inetd` process. For example, if `inetdPid` is the process identifier for the `inetd` process, enter:

```
kill -HUP inetdPid
```

Note: You must assign the *same* port to the `oosqlnw` service on every Objectivity/SQL++ ODBC Driver host.

Testing Objectivity/ODBC

You can verify the correct operation of Objectivity/ODBC in combination with an Objectivity/SQL++ ODBC server by running the following test. This test compiles and links a sample ODBC application and then runs it to access a demonstration federated database. You need Objectivity/C++ to run this test.

The demo federated database is provided with Objectivity/SQL++ on the ODBC server host.

The sample application is a C++ application that calls ODBC 3.0 functions to query and modify a federated database. You can inspect the sample to see how such an application is compiled and linked.

To build and run the sample ODBC-compliant application:

1. Verify that the demo federated database and the Objectivity/SQL++ ODBC server have been set up for this test:
 - If your ODBC server runs on Windows, see “Preparing the ODBC Server for Testing” in Chapter 9 in *Installation and Platform Notes for Windows*.
 - If your ODBC server runs on UNIX, see “Preparing the ODBC Server for Testing” in Chapter 8 in *Installation and Platform Notes for UNIX*.

The Objectivity/SQL++ ODBC server must be running.

2. On the host where you installed Objectivity/ODBC (the driver), change your working directory to the Objectivity/SQL++ support directory. Enter:


```
cd installDir/arch/etc/sql
```
3. In the Objectivity/SQL++ support directory, open the data-source template file `odbc.ini_template` in a text editor.
 - a. Make sure the data-source name on the first line is `[DEMO]`.
 - b. Update each entry as necessary to reflect your environment:

```
Description = DEMO DSN for Objectivity SQL++ ODBC demo
Driver       = driverName (probably ObjectivitySQL)
Host        = hostName (name of the host running your Objectivity/SQL++
              ODBC server)
Database    = Fully qualified boot-file path for the demo federated database
              under the Objectivity/DB installation directory (installDir) on
              hostName—that is:
              installDir\samples\sql\ooisql\DEMO (Windows)
              installDir/arch/samples/sql/ooisql/DEMO (UNIX)
User ID     = A username that has been granted access rights for the tables in
              the demo federated database. By default, this is the
              Objectivity/SQL++ database administrator account (sysstpe).
Password    = Password for the account you entered in the User ID field.
Service     = oosqlnw
```

4. Save the data-source template file.
5. Add the `[DEMO]` data source as a personal data source. Enter:


```
odbcinst -i -s -f odbc.ini_template
```

 For details, see “Adding Objectivity/DB Data Sources” on page 23.

6. Change your working directory to the Objectivity/ODBC samples directory. In a command window, enter:

```
cd installDir/arch/samples/sql/odbc
```

You may wish to make a backup copy of this directory.
7. Edit `makefile` in the Objectivity/ODBC samples directory.
 - Set `INSTALL_DIR` to be the location of the Objectivity/DB installation directory—for example:

```
INSTALL_DIR = /usr/object
```
 - Set `LS_HOST` to be the name of the host running the lock server—for example:

```
LS_HOST = myLockServerHost
```
8. Check whether the lock server is running; start it, if necessary.
9. Build and run the sample application. At the command prompt, enter:

```
make
```
10. Run the sample application. Enter:

```
./demo.sh
```

Press Return at each prompt to continue the demo.

If the Objectivity/SQL++ ODBC programming interface is set up correctly, you will see messages like these:

```
Welcome to the Objectivity/SQL++ ODBC demo
...
Building the ooODBCdemo executable
...
Running the demo...
Demo complete.
Comparing the results.
Test PASSED -- The expected results were achieved.
No errors.
```

Index

B

boot file 7

C

customer support 5

D

data source 9

- adding for federated database 13, 14, 23, 24
- personal or system 24

demo applications

- Objectivity/ODBC 16, 26

driver-manager, ODBC 8, 20

E

environment variables

- OO_SQL_DIR 21, 23, 24

F

federated database

- adding data sources 13, 14, 23, 24
- boot file 7

files

- boot 7
- system-database 7

H

host 9

hostname for ODBC server 15, 25

hprisc

- architecture 19

hpuxia64

- architecture 19

I

installing

- Objectivity/ODBC 11, 19

L

linux86_64

- architecture 19

linux86gcc3

- architecture 19

O

Objectivity/ODBC

- adding data sources 13, 14, 23, 24

- configuring TCP/IP 15, 25

- installing 11, 19

- ODBC driver-manager 20

- platforms for 11, 19

- preparing to use 13, 21

- registering with driver-manager 22

- release files 13, 22

- sample client application 16, 26

- software requirements 12, 20

system requirements 11, 19

TCP/IP port 26

testing 16, 26

Objectivity/SQL++

ODBC server

confirming correct setup 10

defined 8

registering port number 15, 26

specifying hostname 15, 25

Objectivity/SQL++ ODBC Driver (see Objectivity/ODBC)

ODBC (Open Database Connectivity) 7

ODBC driver (see Objectivity/ODBC)

ODBC driver-manager 8

ODBC server (see Objectivity/SQL++)

ODBC-compliant applications 7

odbcinst 23, 25

OO_SQL_DIR environment variable 21, 23, 24

oosqlnw service 15, 26

P

port number

registering, for ODBC server 15, 26

preparing to use

Objectivity/ODBC 13, 21

R

registering Objectivity/ODBC with a driver-manager 22

S

sample applications

Objectivity/ODBC 16, 26

solaris7

architecture 19

solaris86_64

architecture 19

sparc64

architecture 19

system-database

file 7

T

TCP/IP 9

configuring for Objectivity/ODBC 15, 25
services file 16, 26

U

unixODBC DriverManager 20

W

Windows 2003, abbreviation 11

Windows 2008, abbreviation 11

Windows XP x64, abbreviation 11

Windows XP, abbreviation 11