INFORMIX®-Open

User Manual

for Microsoft® Windows™ Environments and UNIX®

Version 2.5
December 1996
Part No. 000-8985

Published by INFORMIX® Press

Informix Software, Inc.
4100 Bohannon Drive
Menlo Park, CA 94025

Documentation Team: Evelyn Eldridge-Diaz, Geeta Karmarkar, Patrice O'Neill, Katherine Schaefer

Table of Contents

Introduction

This chapter introduces the *INFORMIX-Open User Manual*. Read this chapter for an overview of the information provided in this manual and for an understanding of the conventions used throughout this manual.

About This Manual

The *INFORMIX-Open User Manual* explains how to use INFORMIX-Open to allow your Informix products that use open database connectivity (ODBC) to access an Informix, Oracle, or Sybase relational database and describes the special features of INFORMIX-Open.

Organization of this Manual

The *INFORMIX-Open User Manual* includes the following chapters:

- Chapter 1, "INFORMIX-Open Basics," introduces INFORMIX-Open and discusses the basic concepts of the ODBC standard on which it is based.
- Chapter 2, "Informix Driver for UNIX," describes the Informix driver that allows you to connect your ODBC-compliant, UNIX client application to an Informix database server. The chapter discusses how to add, modify, and connect to an Informix data source.
- Chapter 3, "Informix Driver for Windows," describes the Informix driver that allows you to connect your ODBC-compliant, Windows client application to an Informix database server. The chapter discusses how to add, modify, and connect to an Informix data source.

- Chapter 4, "Oracle Driver for UNIX," describes the Oracle driver that allows you to connect your ODBC-compliant, UNIX client application to an Oracle database server. The chapter discusses how to add, modify, and connect to an Oracle data source.
- Chapter 5, "Oracle Driver for Windows," describes the Oracle driver that allows you to connect your ODBC-compliant, Windows client application to an Oracle database server. The chapter discusses how to add, modify, and connect to an Oracle data source.
- Chapter 6, "Sybase Driver for UNIX," describes the Sybase driver that allows you to connect your ODBC-compliant, UNIX client application to a Sybase database server. The chapter discusses how to add, modify, and connect to a Sybase data source.
- Chapter 7, "Sybase Driver for Windows," describes the Sybase driver that allows you to connect your ODBC-compliant, Windows client application to a Sybase database server. The chapter discusses how to add, modify, and connect to a Sybase data source.
- Appendix A lists and describes the syntax for the five types of scalar functions specified by ODBC.

Types of Users

This manual is for database administrators who are using Informix products that use ODBC to access relational databases on Informix, Oracle, or Sybase database servers. The manual assumes that you are familiar with your computer operating system and the structure of relational databases.

Software Dependencies

Each chapter lists individual software dependencies. Depending on your database configuration, you must have a compatible database server installed on your system or network. The following list shows examples of compatible database servers:

- INFORMIX-OnLine Dynamic Server (Version 7.x)
- Oracle 7.1x
- Sybase 10x

For a complete listing of compatible database servers, see the release notes for INFORMIX-Open.

New Features of This Product

This section highlights the major new features implemented in Version 2.5 of INFORMIX-Open.

The following new features are common to all INFORMIX-Open drivers:

- Support for additional connection attributes (new attributes vary)
- Support for Windows 95 and Windows NT environments
- Double-byte character support through standard language support in Windows evironments

  To enable this functionality, you must have a double-byte localized Microsoft Windows environment.
- Global Language Support (GLS)

  INFORMIX-Open includes the GLS libraries that allow Informix applications to handle different languages, cultural conventions, and code sets.

The Informix driver supports the following ODBC API level-2 functions:

- SQLMoreResults
- SQLColumnPrivileges
- SQLTablePrivileges
- SQLParamOptions
- SQLProcedures

The Oracle driver supports the following ODBC API level-2 functions:

- SQLForeignKeys
- SQLParamOptions
- SQLProcedureColumns
- SQLProcedures

The Sybase driver supports the following ODBC API level-2 functions:

- SQLColumnPrivileges
- SQLForeignKeys
- SQLParamOptions
- SQLProcedureColumns
- SQLProcedures
- SQLTablePrivileges

Documentation Conventions

This section describes the conventions that are used in this manual. By becoming familiar with these conventions, you will find it easier to gather information from this and other volumes in the documentation set.

The following conventions are covered:

- Typographical conventions
- Icon conventions
- Example-code conventions
- Screen-illustration conventions

Typographical Conventions

This manual uses the following standard set of conventions to introduce new terms, illustrate screen displays, describe command syntax, and so forth.

Convention	Meaning
KEYWORD	All keywords appear in uppercase letters in a serif font.
italics	Within text, new terms and emphasized words appear in italics. Within syntax diagrams, values that you are to specify appear in italics.
boldface	Identifiers (names of classes, objects, constants, events, functions, program variables, forms, labels, and reports), environment variables, database names, filenames, table names, column names, icons, menu items, command names, and other similar terms appear in boldface.
`monospace`	Information that the product displays and information that you enter appear in a monospace typeface.
KEYSTROKE	Keys that you are to press appear in uppercase letters in a sans serif font.
♦	This symbol indicates the end of feature- or platform-specific information.
→	This symbol indicates a menu item. For example, "Choose **Tools→Options**" means chose the **Options** item from the **Tools** menu.

Tip: *When you are instructed to "enter" characters or to "execute" a command, immediately press* RETURN *after the entry. When you are instructed to "type" the text or to "press" other keys, no* RETURN *is required.*

Icon Conventions

Throughout the documentation, you will find text that is identified by several different types of icons. This section describes these icons.

Comment Icons

Comment icons identify warnings, important notes, or tips. This information is always displayed in italics.

Icon	Description
	The *warning* icon identifies vital instructions, cautions, or critical information.
	The *important* icon identifies significant information about the feature or operation that is being described.
	The *tip* icon identifies additional details or shortcuts for the functionality that is being described.

Feature and Platform Icons

Feature and platform icons identify paragraphs that describe feature-specific or platform-specific information. The following table describes the feature and platform icons that are used in this manual.

Icon	Description
GLS	Identifies information that is specific to a database or application that uses the Informix Global Language Support (GLS) feature for support of a nondefault locale.
UNIX	Identifies information that is specific to 32-bit support for the UNIX environment.

Icon	Description
Windows 3.1	Identifies information that is specific to 16-bit support for the Windows 3.1x environment.
Windows NT	Identifies information that is specific to the Windows NT environment.
Windows 95	Identifies information that is specific to the Windows 95 environment.

These icons can apply to a row in a table, one or more paragraphs, or an entire section. A ♦ symbol indicates the end of the feature- or platform-specific information.

Example Code Conventions

Examples of C and SQL code occur throughout this manual. The code is not specific to any single Informix application development tool, except where it is noted otherwise.

An ellipsis (...) in example code indicates that arguments can be repeated several times. Also note that a column of three dots in an example indicates that additional code would be added in a full application, but it is not necessary to show more code to describe the concept being discussed.

Screen-Illustration Conventions

The illustrations in this manual represent a generic rendition of various windowing environments. The details of dialog boxes, controls, and windows were deleted or redesigned to provide this generic look. Therefore, the illustrations in this manual depict the windowing environment a little differently than the way it appears on your screen.

Additional Documentation

This section describes the following pieces of the documentation set:

- Printed documentation
- On-line documentation
- Related reading

Printed Documentation

The following related Informix documents complement the information in this manual:

- You, or whoever installs your Informix products, should refer to the installation guide that comes with this product to ensure that your Informix product is properly set up before you begin to work with it.
- If you are using an Informix database server, you or your system administrator need the administrator's guide for that database server. If you are using an Oracle or Sybase database server, refer to your documentation for that database server.
- The *Guide to GLS Functionality* describes how to use nondefault locales with Informix products. A locale contains language- and culture-specific information that Informix products use when they format and interpret data.
- The *Informix Error Messages* manual provides information on the cause and corrective action for each numbered error message that is returned by an Informix database server. For error messages that are returned from an Oracle or Sybase database server, refer to your Oracle or Sybase documentation.

On-Line Documentation

Several different types of on-line documentation are available:

- On-line error messages
- Documentation notes, release notes, and machine notes

Error Message Files

If you are using an Informix database server, Informix software products provide ASCII files that contain all of the Informix error messages and their corrective actions. To read the error messages in the ASCII file, Informix provides scripts that let you display error messages on the screen (**finderr**) or print formatted error messages (**rofferr**). See the Introduction to the *Informix Error Messages* manual for a detailed description of these scripts.

To display the error messages in Windows NT, double click the **finderr** icon in the program group for the database server that you are using.

For information on native error messages generated by Oracle or Sybase database servers, refer to your Oracle or Sybase documentation.

Documentation Notes, Release Notes and Machine Notes

In addition to the INFORMIX-Open manuals, the following on-line files might supplement the information in this manual. For UNIX, these files are located in the **$INFORMIXDIR/release/en_us/0333** directory. For Windows, these files are located in the **$INFORMIXDIR\release** directory.

On-Line File	Purpose
Documentation notes	The documentation notes file describes features that are not covered in the manuals or that have been modified since publication. For UNIX, this file is **OPENDOC_2.5**. For Windows, this file is **opendoc.250**.
Release notes	The release notes file describes feature differences from earlier versions of Informix products and how these differences might affect current products. This file also contains information about any known problems and their workarounds. For UNIX, this file is **OPENREL_2.5**. For Windows, this file is **opennote.250**.
Machine notes	The machine notes file describes any special actions that are required to configure and use Informix products on your computer. This file is **OPEN_2.5**. Windows platforms do not have machine notes. ♦

UNIX

Please examine these files because they contain vital information about application and performance issues.

Related Reading

For additional information on ODBC, refer to the *Microsoft ODBC Programmer's Reference and SDK Guide*, Version 2.0 (Microsoft Press, 1994).

Compliance with Industry Standards

INFORMIX-Open is based on the Microsoft Open Database Connectivity (ODBC) Specification, Version 2.5, which, in turn, is based on the X/Open and SQL Access Group Call Level Interface (CLI) Specification. The ODBC and CLI specifications provide a common and open interface through which ANSI-compliant SQL is passed.

Informix Welcomes Your Comments

Please tell us what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about corrections or clarifications that you would find useful. Please include the following information:

- The name and version of the manual that you are using
- Any comments that you have about the manual
- Your name, address, and phone number

Write to us at the following address:

Informix Software, Inc.
SCT Technical Publications Department
4100 Bohannon Drive
Menlo Park, CA 94025

If you prefer to send electronic mail, our address is:

doc@informix.com

Or, send a facsimile to the Informix Technical Publications Department at:

415-926-6571

We appreciate your feedback.

Chapter

1

INFORMIX-Open Basics

This chapter introduces INFORMIX-Open and describes its relationship to ODBC. The chapter describes the initialization file format, lists ODBC application-programming interface (API) and structured query language (SQL) grammar-conformance levels, and describes the driver error-message format.

What Is INFORMIX-Open?

INFORMIX-Open is a set of libraries that allows your Informix ODBC-compliant application to connect to and interact with an Informix, Oracle, or Sybase database server.

INFORMIX-Open is based on the Microsoft Open Database Connectivity (ODBC) 2.5 Specification. The Microsoft specification, which is based on work done by the SQL Access Group and X/Open, defines a vendor-independent application-programming interface (API) for accessing data stored in relational and nonrelational databases. The ODBC architecture consists of four components.

Component	Function
Database application	Calls functions in the ODBC API to access a data source.
Driver manager	Provides information to an application—such as a list of available data sources and drivers— loads drivers dynamically as they are needed, and provides argument and state transition checking.

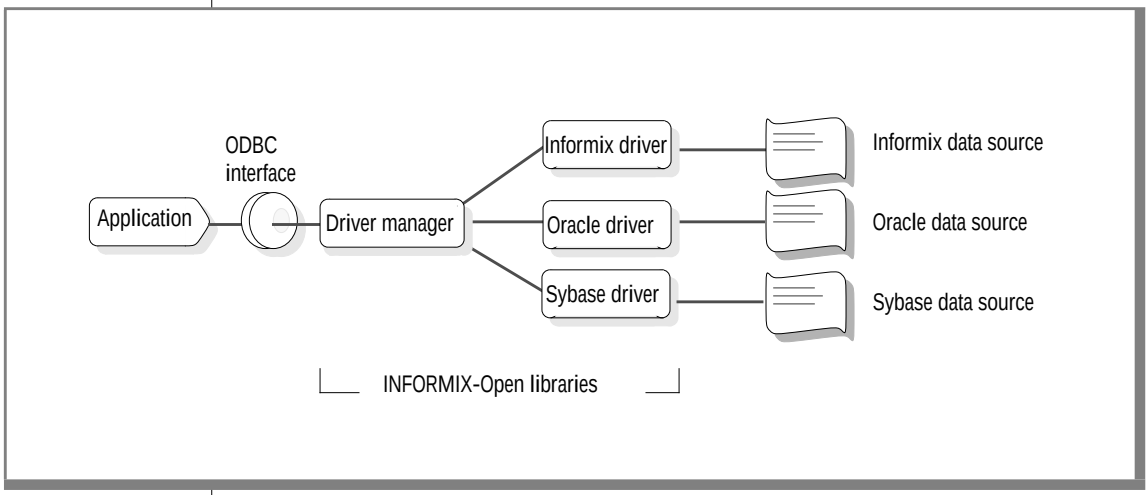
(1 of 2)

Component	Function
Drivers	Each driver processes ODBC function calls and manages exchanges between an application and a data source.
Data source	Contains the data that an application needs to access. The data source includes the data, the database management system (DBMS) in which the data is stored, the platform on which the DBMS resides, and the network (if any) used to access the DBMS.

(2 of 2)

INFORMIX-Open consists of two ODBC components—the driver manager and a set of three drivers as shown in Figure 1-1. The driver manager and each driver appear to an application as one unit that processes ODBC function calls. For example, the Informix driver (the driver manager and the Informix driver) allows you to access, query, and update data in an Informix database.

Figure 1-1
INFORMIX-Open and ODBC Architecture



INFORMIX-Open is available in one version for Microsoft Windows environments and another version for UNIX. The structure of the initialization file that the drivers use, the API functions and SQL grammar that the drivers support, and the error message formats, are the same in both versions. For Windows environments, the drivers are dynamically linked libraries (DLL). For UNIX, the drivers are shared libraries.

Understanding the odbc.ini File

The **odbc.ini** file is an initialization file that the INFORMIX-Open ODBC drivers and driver manager use. Although this file has a slightly different name depending on whether you use a UNIX or Windows product, the format and structure of the file is the same.

UNIX

For UNIX users, **odbc.ini** is a text file called **.odbc.ini**. The file is located in your home directory. In addition, a template **.odbc.ini** file resides in the **$INFORMIXDIR/cli** directory where your INFORMIX-Open software is installed. UNIX users are responsible for modifying their **.odbc.ini** file using a text editor. For driver-specific **.odbc.ini** changes and information, refer to the appropriate driver chapter in this manual. ♦

Windows 3.1

For Windows 3.1, **odbc.ini** is a text file called **ODBC.INI**. The file is located in the **WINDOWS** directory. Before you can connect to a data source, you must use the ODBC Administrator to add the connection information for the data source to this file. ♦

Warning: *Windows 3.1 users should never modify the **ODBC.INI** file directly. To modify the contents of this file, use the ODBC Administrator. If you modify the **ODBC.INI** file directly, data-source configuration errors might result.*

Windows NT

Windows 95

For Windows NT and Windows 95, **ODBC.INI** is a subkey of the HKEY_CURRENT_USER key within the Registry. When you access the Registry using this subkey, the ODBC structure is the same as the structure described in "odbc.ini File Format" on page 1-6. Before you can connect to a data source, you must use the ODBC Administrator to add the connection information for the data source to the **ODBC.INI** subkey. ♦

odbc.ini File Format

The **odbc.ini** file consists of the following sections.

Section	Function
ODBC Data Sources	Lists the name of each data source and describes its associated driver.
Data Source Specification	Each data source listed in the ODBC Data Sources section has a section that contains additional information about that data source.
Default Data Source Specification	Optional. Specifies the default data source to use when none is specified at connection time.
ODBC Options	Specifies ODBC options that can be enabled or disabled.

ODBC Data Sources

Each entry in the ODBC Data Sources section lists a data source and a description of the driver it uses. This section has the following format:

```
[ODBC Data Sources]
data_source_name1=driver_description1
data_source_name2=driver_description2
.
.
.
```

Each entry lists the following information.

Element	Description
data_source_name	identifies the data source that the driver accesses. You define the *data_source_name*. The name can be any name that you choose.
driver_description	describes the database driver that accesses the data source. This field is optional.

For example, to define a data source called **stores7,** which uses the Informix driver, the **odbc.ini** entry might look like the following example:

```
[ODBC Data Sources]
stores7=Informix driver
```

Data Source Specification

Each data source listed in the ODBC Data Sources section has its own Data Source Specification section. This section has the following format:

```
[data_source_name]
Driver=driver_path
TranslationDLL=translation_path
TranslationName=translator_name
TranslationOption=translation_option
attribute=attribute_value
.
.
.
```

Each entry lists the following information.

Element	Description
data_source_name	is the name of the data source, as specified in the ODBC Data Sources section of this file.
driver_path	is the full path to the driver shared library (UNIX) or DLL (Windows).
translation_path	is the full path of the translation shared library (UNIX) or DLL (Windows). This field is optional.
translator_name	is the name of the translator as listed in the ODBC Translators section of the **.odbcinst.ini** file. This field is optional.
translation_option	shows the ASCII representation of the 32-bit integer translation option. This field is optional.
attribute	specifies optional connection information. Each driver has its own set of attributes. For driver-specific attributes, refer to "Adding and Modifying Data Sources" in the chapter for the appropriate driver.
attribute_value	specifies the value for the attribute.

For example, the Data Source Specification entry for the **stores7** data source might look like the following example:

```
[Stores7]
Driver=/usr/informix/cli/dlls/IXinf708.so
Database=stores7
LogonID=marvin
```

In this example, the data source called **stores7** accesses the required attribute, database (called **stores7**), that resides on an Informix database server. The optional attribute, **LogonID**, specifies the default user ID as **marvin**. To connect to this data source, an application uses the Informix driver for Solaris called **IXinf708.so**.

Default Data Source Specification

The Default Data Source Specification, which is optional, contains information about the default data source. This data source, called Default, has the same format as any other Data Source Specification section. However, the default data source is not listed in the ODBC Data Sources section.

The following example shows a Default Data Source Specification entry for an Informix database called **Agencies**:

```
[Default]
Driver=/usr/informix/cli/dlls/IXinf708.so
Database=Agencies
LogonID=marvin
```

ODBC Options

The ODBC Options section indicates whether tracing is enabled or disabled. When tracing is enabled, all ODBC function calls made from an application can be logged to the specified trace file. This section has the following format:

```
[ODBC]
Trace=0|1
TraceFile=tracefile_path
TraceAutoStop=0|1
```

This section lists the following information:

Trace	indicates whether tracing is enabled. If Trace is set to `0`, tracing is disabled. If Trace is set to `1`, tracing is enabled.
TraceFile	specifies the full path to the trace file that logs the ODBC function calls. If a trace file is not specified and tracing is enabled, logging information is written to the **sql.log** file located in your current directory.
TraceAutoStop	indicates whether tracing is enabled or disabled when an application calls the **SQLFreeEnv** function. If TraceAutoStop is set to `0`, tracing is not automatically disabled. If TraceAutoStop is set to `1`, the Trace attribute in the **odbc.ini** file is set to `0`. Additionally, on Windows all other concurrent ODBC applications have tracing disabled. The default value is `1`.

odbc.ini File Examples

The following example shows a sample Windows 3.1 **ODBC.INI** file:

```
[ODBC Data Sources]
Stores7=Informix Driver
Pubs2=Sybase Driver

[Default]
Driver=C:\WINDOWS\SYSTEM\IVINF509.DLL
Database=vendors
LogonID=mary
```

```
[Stores7]
Driver=C:\WINDOWS\SYSTEM\IVINF509.DLL
Hostname=odin
Service=sqlexec
Database=stores7
LogonID=mary

[Pubs2]
Driver=C:\WINDOWS\SYSTEM\IVSYB09.DLL
ServerName=Cyber10
DatabaseName=pubs2
LogonID=mary

[ODBC]
Trace=1
TraceFile=C:\WINDOWS\LOG\TRACE.LOG
TraceAutoStop=0
```

The following example shows a sample UNIX **.odbc.ini** file:

```
[ODBC Data Sources]
Stores7=Informix Driver
ORASRV=Oracle Driver

[Default]
Driver=/usr/informix/cli/dlls/IXinf708.so
Database=Agencies

[Stores7]
Driver=/usr/informix/cli/dlls/IXinf708.so
Description=Informix 7.x ODBC Driver
Database=stores7
LogonID=marvin
Password=

[ORASRV]
Driver=/usr/informix/cli/dlls/IXor708.so
Description=Oracle 7.x ODBC Driver
ServerName=T:mickey:ORASRV
LogonID=marvin
Password=

[ODBC]
Trace=1
TraceFile=/tmp/trace.log
TraceAutoStop=0
```

ODBC Conformance Levels

ODBC defines two different conformance standards for drivers—the API conformance standard and the SQL conformance standard. Each conformance standard consists of levels.

API Conformance Levels

The API conformance standard consists of three levels:

Core API	A set of core functions that correspond to the functions in the X/Open and SQL Access Group Call-Level Interface specification
Level 1 API	Core API functionality plus all level-1 functionality
Level 2 API	Core and level-1 functionality plus all level-2 functionality

The INFORMIX-Open ODBC drivers are at level-1 conformance. In addition, each driver supports a set of level-2 functions. For a list of the level-2 functions that a specific driver supports, refer to the "ODBC Conformance Levels" section in the chapter for the appropriate driver.

The following table lists all ODBC core, level-1, and level-2 API functions.

Core	Level 1	Level 2
SQLAllocConnect	SQLBindParameter	SQLBrowseConnect
SQLAllocEnv	SQLColumns	SQLColumnPrivileges
SQLAllocStmt	SQLDriverConnect	SQLDataSources
SQLBindCol	SQLGetConnectOption	SQLDescribeParam
SQLCancel	SQLGetData	SQLDrivers
SQLColAttributes	SQLGetInfo	SQLExtendedFetch
SQLConnect	SQLGetFunctions	SQLForeignKeys

(1 of 2)

Core	Level 1	Level 2
SQLDescribeCol	SQLGetStmtOption	SQLMoreResults
SQLDisconnect	SQLGetTypeInfo	SQLNativeSql
SQLError	SQLParamData	SQLNumParams
SQLExecDirect	SQLPutData	SQLParamOptions
SQLExecute	SQLSetConnectOption	SQLPrimaryKeys
SQLFetch	SQLSetStmtOption	SQLProcedureColumns
SQLFreeConnect	SQLSpecialColumns	SQLProcedures
SQLFreeEnv	SQLStatistics	SQLSetPos
SQLFreeStmt	SQLTables	SQLSetScrollOptions
SQLGetCursorName		SQLTablePrivileges
SQLNumResultCols		
SQLPrepare		
SQLRowCount		
SQLSetCursorName		
SQLTransact		

(2 of 2)

SQL Conformance Levels

SQL conformance consists of the following three levels:

- **Minimum**. This level is designed to meet the basic level of ODBC conformance.
- **Core**. This level roughly corresponds to the X/Open and SQL Access Group SQL CAE specification (1992).
- **Extended**. This level provides common DBMS extensions to SQL.

The Sybase driver meets minimum SQL conformance, and the Informix and Oracle drivers meets core SQL conformance. In addition, each driver supports several extended SQL statements, expressions, and data types. For a list of the Extended SQL that a specific driver supports, refer to the "ODBC Conformance Levels" section in the chapter for the appropriate driver.

Minimum SQL Conformance

The minimum level of SQL conformance consists of the following statements, expressions, and data types:

- Data Definition Language (DDL) statements: CREATE TABLE and DROP TABLE
- Data Manipulation Language (DML) statements: simple SELECT, INSERT, UPDATE, SEARCHED, and DELETE SEARCHED
- Simple expressions (such as A > B + C)
- Data types: CHAR, VARCHAR, or LONG VARCHAR

Core SQL Conformance

The core level of SQL conformance consists of the following statements, expressions, and data types:

- Minimum SQL grammar and data types
- DDL statements: ALTER TABLE, CREATE INDEX, DROP INDEX, CREATE VIEW, DROP VIEW, GRANT, and REVOKE
- DML statement: full SELECT
- Expressions: subquery and set functions such as SUM and MIN
- Data types: DECIMAL, NUMERIC, SMALLINT, INTEGER, REAL, FLOAT, and DOUBLE PRECISION

Extended SQL Conformance

The extended level of SQL conformance consists of the following statements, expressions, and data types:

- Minimum and core SQL grammar and data types
- DML statements: outer joins, positioned UPDATE, positioned DELETE, SELECT FOR UPDATE, and unions
- Expressions: scalar functions such as SUBSTRING, ABS, date, time, and time-stamp literals
- Data types: BIT, TINYINT, BIGINT, BINARY, VARBINARY, LONG VARBINARY, DATE, TIME, and TIMESTAMP
- Batch SQL statements
- Procedure calls

Error Messages

The application you are using might allow you to access error messages that INFORMIX-Open returns. In INFORMIX-Open, **SQLError** returns error messages that follow the standard ODBC format. Error messages not only explain the error but also provide the identity of the component in which the error occurred. The following sections explain this format.

Error Text Format

Error messages returned by **SQLError** come from two sources: data sources and INFORMIX-Open components (the driver manager or a driver). Consequently, the error text returned by **SQLError** has two formats.

If an INFORMIX-Open component receives an error message from a data source, it identifies the data source as the source of the error and identifies itself as the component that received the error. For errors that occur in a data source, the error text appears in the following format:

```
[vendor_identifier][INFORMIX-Open_component_identifier]
[data_source_identifier]data_source_supplied_text
```

If the source of an error is an INFORMIX-Open component, the error message identifies the component. For these errors, the error text appears in the following format:

```
[vendor_identifier][INFORMIX-Open_component_identifier]
component_supplied_text
```

The following table shows the meaning of each element.

Element	Description
vendor_identifier	Identifies the vendor of the component in which the error occurred or that received the error directly from the data source.
INFORMIX-Open_*component_identifier*	Identifies the component in which the error occurred or that received the error directly from the data source.
data_source_identifier	Identifies the data source. For multiple-tier drivers, this is the DBMS product.
component_supplied_text	Error text generated by the INFORMIX-Open component (the driver manager or the driver)
data_source_supplied_text	Error text generated by the data source

The brackets ([ ]) are included in the error text; they do not indicate optional items.

Sample Error Messages

The following examples show how various components in a connection might generate the text of error messages and how INFORMIX-Open might return the error messages to the application with **SQLError**.

Sample Error Returned from the Driver

An Informix driver sends requests to a DBMS and returns information to the application through the driver manager. Because it is the component that interfaces with the driver manager, the driver formats and returns arguments for **SQLError**.

For example, if an Informix driver encounters an invalid ODBC function call sequence, it might return the following arguments for **SQLError**:

```
SQLSTATE : S1010
Native Error: 0
MSG      : [INTERSOLV] [ODBC Informix Driver] Function
           sequence not found.
```

Because the error occurred in the driver, it adds prefixes to the error text for the vendor (INTERSOLV) and the driver (ODBC Informix Driver).

Sample Error Returned from the Driver Manager

The driver manager can also generate error messages. For example, if an invalid driver is entered in the **.odbc.ini** file, the driver manager might format and return the following arguments for **SQLError**:

```
SQLSTATE : IM003
NATIVE ERROR: 0
MSG      : [INTERSOLV] [ODBC DLL] Driver specified by
           data source name not found.
```

Because the error occurred in the driver manager, it adds prefixes to the error text for its vendor (INTERSOLV) and its identifier (ODBC DLL).

Sample Error Returned from the Data Source

If the DBMS could not find the table **EMPLOYEE**, the driver might format and return the following arguments for **SQLError**:

```
SQLPrepare has Failed.
SQLState = S0002
NativeError = -206
ErrorMsg = "[INTERSOLV][ODBC Informix Driver][Informix]The
specified table (EMPLOYEE) is not in the database."
```

Because the error occurred in the data source, the driver added a prefix for the data-source identifier (Informix) to the error text. Because the driver component interfaced with the data source, it adds prefixes to the error text for its vendor (INTERSOLV) and identifier (ODBC Informix Driver). For a description of an error that the data source returns, refer to DBMS-specific documentation. In this case, look up the native-error value in the *Informix Error Messages* manual.

Chapter

2

Informix Driver for UNIX

This chapter describes the Informix driver for UNIX, and explains how to complete the following tasks:

- Set up the Informix driver
- Add or modify an Informix data source
- Connect to an Informix data source

In addition, the chapter includes information on the ODBC API and SQL conformance levels, data-type mappings, and isolation and locking levels that the Informix driver supports.

Setting Up the Informix Driver

The Informix driver is installed during the driver-installation procedure. The name of the current driver is **IXinf708**. The driver might have a platform-specific extension; for example, the driver for Solaris is **IXinf708.so**.

You must perform the following setup tasks before you can use the driver:

- Confirm that your system has the appropriate software installed.
- Set environment variables to reflect the appropriate directory paths.

System Requirements

To use the Informix driver, you will need the appropriate client-software library:

- INFORMIX-NET (Version 5.01)
- INFORMIX-Connect (Version 7.2x)

Depending on your Informix database configuration, you must have a compatible Informix database server installed on your system or network. The following list shows examples of compatible database servers:

- INFORMIX-OnLine Dynamic Server (Version 7.x)
- INFORMIX-SE (Version 7.x)

In this chapter, OnLine refers to INFORMIX-OnLine Dynamic Server.

For information on these products, refer to your Informix documentation set. For a current list of compatible database servers, see the release notes for INFORMIX-Open.

GLS

To take advantage of the GLS feature, you must connect to an Informix database server of Version 7.2 or later. For more information, see the *Guide to GLS Functionality*. ♦

Setting Environment Variables

Set the **INFORMIXDIR** environment variable to the fully qualified path of the directory where your Informix product is installed. The **INFORMIXSERVER** environment variable must specify the default database server for the user. This value must correspond to a valid dbservername entry in the **sqlhosts** file. The UNIX environment variable, **PATH**, indicates the directories that are searched for executable programs. Your **PATH** setting must include the path to your **$INFORMIXDIR/bin** directory.

You can set these variables at the command line. For example, if your Informix directory path is **/usr/informix** and the name of your default database server is **online_one**, in the C shell you can use the following commands:

```
setenv INFORMIXDIR /usr/informix
setenv INFORMIXSERVER online_one
setenv PATH ${PATH}:${INFORMIXDIR}/bin
```

In the Bourne or Korn shells, you can use the following commands:

```
INFORMIXDIR=/usr/informix; export INFORMIXDIR
INFORMIXSERVER=online_one; export INFORMIXSERVER
PATH=${PATH}:$INFORMIXDIR/bin; export PATH
```

If you set these variables at the command line, you must reset them whenever you log on to your system. To set these variables automatically, add the lines to your **.login** or **.cshrc** files (C shell) or your **.login** or **.profile** (Bourne or Korn shells). As an alternative, you can set these variables in the **.open.csh** (C shell) or **.open.sh** (Bourne or Korn shells) file that is installed with INFORMIX-Open. Source or execute this file to set the environment variables.

Important: *All drivers included in INFORMIX-Open use the Informix GLS environment variables. If you want to use a different locale (for example, French), set the DB_LOCALE environment variable to a valid locale name. For more information on valid locales, refer to the "Guide to GLS Functionality," If the data appears to be in a locale that is different from the one that you have set, check that IV_LOCALE, IV_GLS_LCDIR, IV_GLS_REGISTRY, and GL_PATH are not set.*

Adding and Modifying Data Sources

To connect to a data source, the driver manager checks your **.odbc.ini** file for specific connection information. This file contains information about each data source. You must modify your **.odbc.ini** file using a text editor.

Adding a Data Source

Before you can connect to a data source, you must add an entry for that data source in your **.odbc.ini** file. For complete information on the format and contents of the **.odbc.ini** file, refer to "Understanding the odbc.ini File" on page 1-5.

Required Data-Source Configuration Information

When you add a data source, you must provide two pieces of information in its Data Source Specification section.

Attribute	Value
Driver	The full path to your driver shared library
Database	The name of the database to which you want to connect by default This name can include the database server qualifier.

All other connection information is optional.

Optional Data-Source Configuration Information

When you add a data source, you can define additional connection defaults. You can specify two types of connection options in the Data-Source Specification section:

- Options that define names
- Options that define cursor behavior

Options That Define Names

The following table lists the names that you can set as default connection options for a data source. Some of the attributes have default values.

Attribute	Value
Description	A long description that identifies the data source
LogonID	Your user name as specified on the Informix database server

Attribute	Value
HostName	The name of the computer on which the Informix database server resides
Service	The name assigned to the Informix database server process running on your UNIX computer Commonly, the service is **sqlexec**. Confirm the service name with your system administrator.
ServerName	The name of the database server on which the database that you want to access resides

Options That Define Cursor Behavior

The following table lists the types of cursor behavior that you can set as default connection options for a data source. The table lists the default values.

Attribute	Value
CursorBehavior	A value that determines the cursor behavior after the transaction ends ■ 0 = Closed. This is the default setting. ■ 1 = Preserve. Choose this setting to hold the cursors at the current position when the transaction ends. This setting might affect the performance of your database operations.
EnableInsert-Cursors	A value that determines whether the driver can use insert cursors during parameterized inserts ■ 0 = No insert cursors. This is the default setting. ■ 1 = Enable insert cursors. This value improves performance when an application uses the same statement for multiple insert operations. This option enables insert data to be buffered in memory before being written to disk.
EnableScrollable-Cursors	A value that determines if the driver provides scrollable cursors ■ 0 = No scrollable cursors. This is the default setting. ■ 1 = Enable scrollable cursors. If this value is set, select lists must not include long columns, such as SQL_LONGVARCHAR or SQL_LONGVARBINARY.

To add a data source

1. Edit your **.odbc.ini** file using a text editor such as the **vi** editor.

   If you do not have this file in your home directory, copy the default **.odbc.ini** file from the **$INFORMIXDIR/cli** directory into your home directory.

2. Under the ODBC Data Sources section, add an entry for your data source.

   Each entry in this section lists the data source and a description of the driver that the data source uses. Use the following format for data-source entries:

   ```
   [ODBC Data Sources]
   data_source_name=driver_description
   .
   .
   .
   ```

   The following table shows the meaning of each element.

Element	Description
data_source_name	identifies the data source that the Informix driver accesses. You define the *data_source_name*. The name can be any name that you choose.
driver_description	describes the driver that accesses the data source. This field is optional. Set this field to `Informix Driver`.

   For identification purposes, you might want to make the data-source name the same as the name of the database to which you want to connect by default. Then, when you connect to a data source, you know the specific database to which you are connecting. The following example uses the name **stores7**.

   To associate the **stores7** data source with the Informix driver, you would make the following entry in the Data Sources section of your **.odbc.ini** file:

   ```
   [ODBC Data Sources]
   Stores7=Informix Driver
   ```

3. After the ODBC Data Sources section, add an entry for each data source that you specified.

   Each data source listed in the ODBC Data Sources section of your **.odbc.ini** file requires a Data Source Specification section. Use the following format for Data Source Specification entries:

```
[data_source_name]
Driver=driver_path
Database=database_name
attribute=value
.
.
.
```

   The following table shows the meaning of each element.

Element	Description
data_source_name	is the name of the data source, as specified in the ODBC Data Sources section of your **.odbc.ini** file.
driver_path	is the full path to the Informix shared library.
database_name	is the name of the database that the data source accesses. The name can include the Informix database server qualifier.
attribute	specifies optional configuration information. For a list of the attribute that the Informix driver supports, see "Optional Data-Source Configuration Information" on page 2-6.
value	specifies the value for the attribute.

   The required fields in this section are *data_source_name*, *driver_path*, and *database_name*.

   The attributes that you enter in this section become the default connection values for this data-source.

   When the database is available on a database server that is different from the database server identified by the **INFORMIXSERVER** environment variable, the database attribute can be specified as follows:

```
Database=database_name@server_name
```

Important: *The section name for the Data Source Specification must match the data-source name listed in the ODBC Data Sources section of your* ***.odbc.ini*** *file.*

For example, the entry for the **stores7** data source might look like the following example:

```
[Stores7]
Driver=/usr/informix/cli/dlls/IXinf708.so
Database=stores7
```

In this example, the data source and database have the same name: **stores7**.

Tip: *Data-source names are not case sensitive; that is,* ***stores7*** *and* ***Stores7*** *refer to the same data source.*

.odbc.ini Sample Entry for the Informix Driver

A complete **.odbc.ini** data-source entry for the **stores7** data source described in "To add a data source" on page 2-8 would look like the following example:

```
[ODBC Data Sources]
Stores7=Informix Driver

[Stores7]
Driver=/usr/informix/cli/dlls/IXinf708.so
Database=stores7
```

Modifying a Data Source

To edit a data source, use a text editor such as the **vi** editor. Open your **.odbc.ini** file and modify the appropriate lines in that file. The sections that make up this file are described in "Adding a Data Source" on page 2-5.

Connecting to a Data Source

On UNIX, you must supply a connection string to connect to a data source. A connection string consists of a number of attributes that specify how a driver connects to a data source. An attribute identifies a specific piece of information that the driver must know before it can make the appropriate data-source connection.

A connection string has the following format:

```
"DSN=data_source_name[;attribute=value[;attribute=value]...]"
```

You must specify the *data_source_name*, but all other attributes are optional. If you do not specify an attribute, that attribute defaults to the one that is specified in the Data Source Specification section (for the data source specified in the connection string) of your **.odbc.ini** file.

The following table lists long and short names of the attributes that you can include in the Informix driver connection string. With the exception of the data-source name (which must be specified by DSN), you can use either the long or short names in the connection string. For a description of these attributes, see "Optional Data-Source Configuration Information" on page 2-6. The attribute long names correspond to the attributes listed in this section.

Attribute Long Name	Attribute Short Name
DataSourceName	DSN
Database	DB
HostName	HOST
LogonID	UID
Password	PWD
ServerName	SRVR
Service	SERV
CursorBehavior	CB
EnableInsertCursors	EIC
EnableScrollableCursors	ESC

The following example shows a valid connection string for connecting to the **stores7** data source. The string specifically requests a connection to the **stores7** database that resides on a computer named **rainbow**. The user ID is **mary**.

```
"DSN=Stores7;DB=stores7;HOST=rainbow;UID=mary;PWD=secret"
```

ODBC Conformance Levels

ODBC defines the API and the SQL conformance standards for drivers. *API conformance* refers to the functions that a driver supports. *SQL conformance* refers to the SQL statements, expressions, and data types that the driver supports. Each conformance standard consists of three levels. For more information on these standards, see "ODBC Conformance Levels" on page 1-11.

API Conformance Level

The Informix driver supports all core and level-1 API functions listed in "API Conformance Levels" on page 1-11.

- The Informix driver supports the following options for the **SQLGetConnectOption** and **SQLSetConnectOption** level-1 functions.

SQL_ACCESS_MODE	SQL_PACKET_SIZE
SQL_AUTOCOMMIT	SQL_TXN_ISOLATION
SQL_ODBC_CURSORS	SQL_TRANSLATE_DLL
SQL_OPT_TRACE	SQL_TRANSLATE_OPTION
SQL_OPT_TRACEFILE	

  The driver supports SQL_ACCESS_MODE for only the **SQLGetConnectOption** function.

- The Informix driver supports the following options for the **SQLSetStmtOption** and **SQLGetStmtOption** level-1 functions.

SQL_BIND_TYPE	SQL_MAX_ROWS
SQL_CONCURRENCY	SQL_NOSCAN
SQL_CURSOR_TYPE	SQL_RETRIEVE_DATA
SQL_KEYSET_SIZE	SQL_ROWSET_SIZE
SQL_MAX_LENGTH	SQL_ROW_NUMBER

In addition, the Informix driver supports the following level-2 functions.

SQLBrowseConnect	SQLNumParams
SQLColumnPrivileges	SQLParamOptions
SQLDataSources	SQLPrimaryKeys
SQLDrivers	SQLProcedures

(1 of 2)

SQLBrowseConnect	SQLNumParams
SQLExtendedFetch	SQLSetScrollOptions
SQLMoreResults	SQLTablePrivileges
SQLNativeSql	

(2 of 2)

The driver supports scrollable cursors with **SQLExtendedFetch** if the connection attribute EnableScrollableCursors is set to 1.

SQL Conformance Level

The Informix driver meets the core level of SQL conformance that is listed in "SQL Conformance Levels" on page 1-12. In addition, the Informix driver supports the following ODBC extensions to SQL:

- DML statements: outer joins, positioned UPDATE, positioned DELETE, SELECT FOR UPDATE, and UNIONS
- Date and time-stamp data
- The following numeric functions.

abs	acos	asin	atan
atan2	cos	cot	exp
log	log10	mod	power
round	sin	sqrt	tan
truncate			

- The following date functions.

curdate	dayofweek	now
dayofmonth	month	year

- The following string functions.

concat	LTRIM
length	RTRIM

- The following system functions.

database	user

- Procedure calls

For more information on ODBC numeric, date, string, and system functions, see Appendix A, "Scalar Functions."

Mapping Data Types

The Informix driver maps Informix data types to their appropriate ODBC SQL data types. The following table lists the Informix data type and its corresponding ODBC SQL data type.

Informix Data Type	ODBC SQL Data Type
BYTE*	SQL_LONGVARBINARY
CHAR, CHARACTER	SQL_CHAR
DATE	SQL_DATE
DATETIME year to fraction(5)	SQL_TIMESTAMP
DECIMAL, DEC, NUMERIC	SQL_DECIMAL
FLOAT	SQL_DOUBLE
INTEGER, INT	SQL_INTEGER
INTERVAL	SQL_CHAR
MONEY	SQL_DECIMAL
SERIAL	SQL_INTEGER
SMALLFLOAT, REAL	SQL_REAL
SMALLINT	SQL_SMALLINT
TEXT*	SQL_LONGVARCHAR
VARCHAR,* CHARACTER VARYING*	SQL_VARCHAR

* Not supported for INFORMIX-SE databases.

Supported Isolation and Lock Levels

The Informix driver supports isolation-levels 0 (Read Uncommitted), 1 (Read Committed), and 3 (Serializable) if connected to an OnLine database server. The default setting is 1. INFORMIX-SE supports isolation-level 0 (Read Uncommitted) only.

The Informix driver also supports an alternative isolation-level 1, called cursor stability. Your Informix ODBC-compliant application can use this isolation level by calling **SQLSetConnectOption** (1040, 1).

Additionally, if transaction logging is not enabled for your database, transactions are not supported by the driver (the driver is always in auto-commit mode). Each statement is treated as if it is a single transaction.

The Informix driver also supports page-level locking. For more information on isolation and lock levels, see the *Informix Guide to SQL: Tutorial*.

Chapter

3

Informix Driver for Windows

This chapter describes the Informix driver for Windows environments, and explains how to complete the following tasks:

- Set up the Informix driver
- Add or modify an Informix data source
- Connect to an Informix data source

In addition, the chapter includes information on the ODBC API and SQL conformance levels, data-type mappings, and isolation and locking levels that the Informix driver supports.

Setting Up the Informix Driver

The Informix driver is installed during the driver-installation procedure. For Windows 3.1, the current driver is **IVINF509.DLL**. For Windows NT and Windows 95, the current driver is **IXINF709.DLL**.

You must perform the following setup tasks before you can use the driver:

- Confirm that your system has the appropriate software installed.
- Confirm that your environment-variable settings reflect the appropriate directory paths.

System Requirements

To use the Informix driver, you will need the appropriate client-software library:

- INFORMIX-NET (Version 5.01)
- INFORMIX-Connect (Version 7.2x)

Depending on your Informix database configuration, you must have a compatible Informix database server installed on your system or network. The following list shows examples of compatible database servers:

- INFORMIX-OnLine (Version 5.x) (Compatible with Windows 3.1)
- INFORMIX-OnLine Dynamic Server (Version 7.x)
- INFORMIX-SE (Version 7.x)

In this chapter, OnLine refers to either INFORMIX-OnLine or INFORMIX-OnLine Dynamic Server.

For information on these products, refer to your Informix documentation set. For a current list of compatible database servers, see the release notes for INFORMIX-Open.

GLS

To take advantage of the Global Language Support (GLS) feature, you must connect to an Informix database server, Version 7.2 or later. For more information, see the *Guide to GLS Functionality*. ♦

Setting Environment Variables

When you install INFORMIX-Open, the **INFORMIXDIR** environment variable is set to the full path of the directory where your Informix product is installed.

```
SET INFORMIXDIR=C:\INFORMIX
```

For Windows 3.1 and Windows 95, this information is set in the **AUTOEXEC.BAT** file. For Windows NT, this information is set in the Registry.

INFORMIX-Open provides double-byte character support through standard language support in Windows environments. (You must have a double-byte, localized Microsoft Windows environment.) Before you use the Informix driver, ensure that you configure the Windows environment for the language and cultural conventions that you require. For Windows 3.1 and Windows NT, in the **Program Manager**, choose **Main→Control Panel→International** to set these conventions. For Windows 95, set the appropriate language in the Control Panel, Regional Settings dialog box.

Adding and Modifying Data Sources

To add and configure data sources, use the ODBC Administrator. The ODBC Administrator then updates your **ODBC.INI** file in Windows 3.1 or your Registry in Windows 95 and Windows NT to reflect your data-source connection information.

Adding a Data Source

Windows3.1

In Windows 3.1, the **ODBC.INI** file is an initialization file used by the ODBC drivers and is located in the **WINDOWS** directory. This file contains information about each data source and its associated driver. Before you can connect to a data source, you must add the connection information for the data source to the **ODBC.INI** file. The ODBC Administrator adds the information to this file for you; never modify the **ODBC.INI** file directly. For complete information on the format and contents of this file, refer to "Understanding the odbc.ini File" on page 1-5. ♦

Windows NT

Windows 95

In Windows NT and Windows 95, **ODBC.INI** is a subkey of the HKEY_CURRENT_USER key within the Registry. When you access the Registry using this subkey, the ODBC structure is the same as the structure described in the "odbc.ini File Format" on page 1-6. Before you can connect to a data source, you must use the ODBC Administrator to add the connection information for the data source to the Registry. ♦

Required Data-Source Configuration Information

When you add a data source, you must provide two pieces of information.

Attribute	Value
Data Source Name	The name of the data source you want to access
Database	The name of the database to which you want to connect by default This name can include the database server qualifier.

All other connection information is optional.

Optional Data-Source Configuration Information

When you add a data source, you can define additional connection defaults. You can specify three types of options in your data-source setup:

- Options that define names
- Options that define cursor behavior
- Options that define operation handling

Options That Define Names

The following table lists the names that you can set as default connection options for a data source. Some of the attributes have default values.

Attribute	Value
Description	A long description that identifies the data source
Default User Name	The host user ID
Host Name	The name of the computer on which the Informix database server resides
Service Name	The name assigned to the Informix database server process running on your UNIX computer. Commonly, the service is **sqlexec**. Confirm the service name with your system administrator.

(1 of 2)

Attribute	Value
Server Name	The name of the database server on which the database that you want to access resides (available only for Windows 95 and Windows NT)
Get DB List from Informix	A value that determines from where the driver requests the database list to be returned: ■ 1 = Request the database list from the database server. This is the default setting. ■ 0 = Use the database list that is specified by the user at driver setup.
Database List	The list of databases that will be displayed in the logon dialog box (that is, the databases to which you will be able to connect) if Get DB List From Informix is set to 0 If more than one name is specified, the names must be separated by commas.

(2 of 2)

Options That Define Cursor Behavior

The following table lists the types of cursor behavior that you can set as default connection options for a data source. The table lists the default values.

Attribute	Value
Cursor Behavior	A value that determines the cursor behavior after the transaction ends: ■ 0 = Closed. This is the default setting. ■ 1 = Preserve. Choose this setting to hold the cursors at the current position when the transaction ends. This setting might affect the performance of your database operations.
Enable Insert Cursors	A value that determines whether the driver can use insert cursors during parameterized inserts: ■ 0 = Do not enable insert cursors. This is the default setting. ■ 1 = Enable insert cursors. This value improves performance when an application uses the same statement for multiple insert operations. This option enables insert data to be buffered in memory before being written to disk.
Enable Scrollable Cursors	A value that determines if the driver provides scrollable cursors. ■ 0 = No scrollable cursors. This is the default setting. ■ 1 = Enable scrollable cursors. If this value is set, select lists must not include long columns, such as SQL_LONGVARCHAR or SQL_LONGVARBINARY.

Options That Define Operation Behavior

The following table lists the types of operation behavior you can set as default connection options for a data source. The table lists the default values.

Attribute	Description
Protocol Type	The protocol used to communicate with the database server. Values can be IPX, TCP/IP, or PIPE for Windows 3.1. For Windows NT and Windows 95, values can be xxSOCTCP, xxSOCSPX, or SEIPCPIP where xx represents an abbreviation for the type of database server to which you are connecting. For example, SESOCSPX indicates that you are using an INFORMIX-SE with sockets and IPX.
Yield Proc (For Windows 3.1 only)	A value that determines whether you can work in other Windows applications when the Informix driver is busy. ■ 0 = Peek and dispatch. This setting causes the Informix driver to check the Windows message queue and send any messages to the appropriate Windows application. ■ 1 = No yielding. This setting does not let you work in other applications. This is the default setting. ■ 3 = Dispatch with the Windows Yield function. This setting turns control over to the Windows kernel. The Windows kernel checks the message queue and sends any messages to the appropriate application window. The recommended value is 1. ♦

Windows 3.1

Warning: *Windows 3.1 users should never modify the* ***ODBC.INI*** *file directly. To modify the contents of this file, use the ODBC administrator. If you modify the* ***ODBC.INI*** *file directly, data-source configuration errors might result.*

To add a data source

1. Invoke the ODBC Administrator.

   The Data Sources dialog box appears, as Figure 3-1 illustrates.

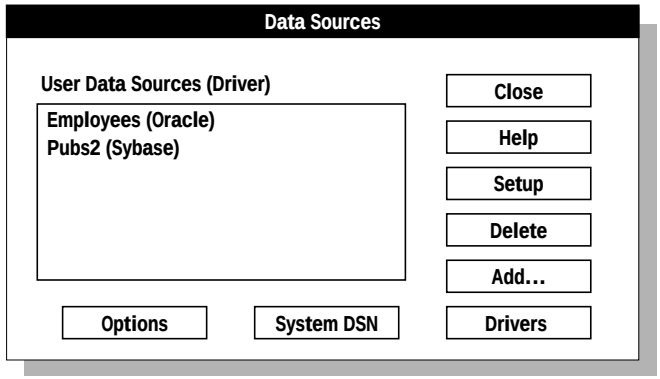


Figure 3-1
The Data Sources Dialog Box

2. Click **Add**.

   The Add Data Source dialog box appears, as Figure 3-2 illustrates.

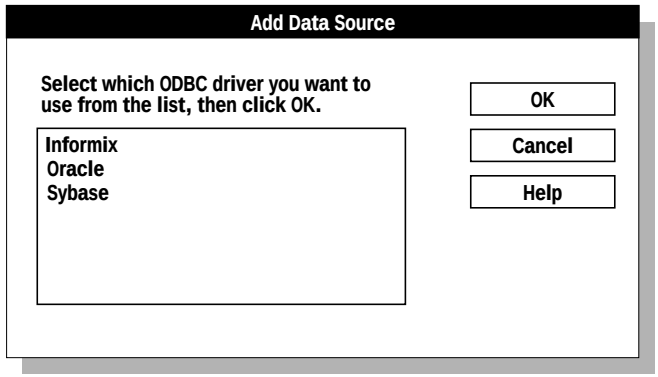


Figure 3-2
The Add Data Source Dialog Box

3. Select the Informix driver from the Installed ODBC Drivers list.
4. Click **OK**.

   The ODBC INFORMIX Setup dialog box appears, as Figure 3-3 illustrates.

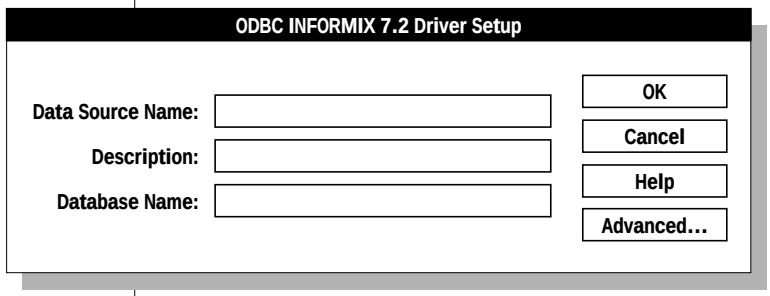


Figure 3-3
The ODBC INFORMIX Driver Setup Dialog Box

5. In the **Data Source Name** text box, enter the name of the data source you want to access.

   You define the data-source name; that is, it can be any name that you choose.
6. In the **Description** text box, type a long description of your data source. This section is optional.

7. In the **Database Name** text box, enter the name of the database to which you want to connect by default.

   You now have entered enough information to connect to the data source:

   - To add the data source, click **OK.**

     The Data Sources dialog box appears, as Figure 3-1 on page 3-10 illustrates. You can then click **Add** to add another data source or click **Close** to exit the Data Sources dialog box.

   - To add optional connection information about the data source, click **Advanced**.

     The ODBC INFORMIX 7.2 Advanced Driver Setup dialog box appears, as Figure 3-4 illustrates. Check that the values in this dialog box conform to your **setnet** settings.

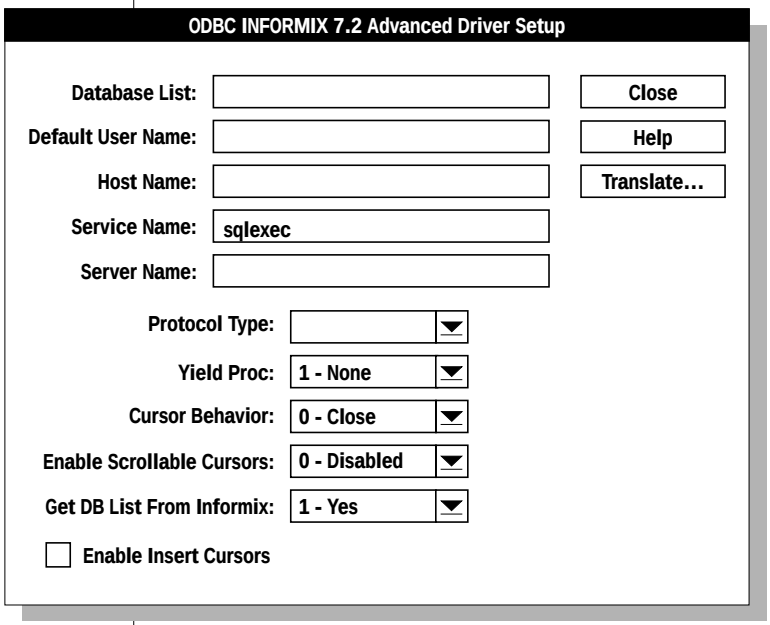


Figure 3-4
The ODBC INFORMIX Advanced Driver Setup Dialog Box

8. Enter the information you want to specify about this data source.

   You can also choose options from the drop-down list boxes. For a description of the advanced connection options that are available, see "Optional Data-Source Configuration Information" on page 3-6.

9. To specify a translator, click **Translate**.

   The Select Translator dialog box appears, as Figure 3-5 illustrates.

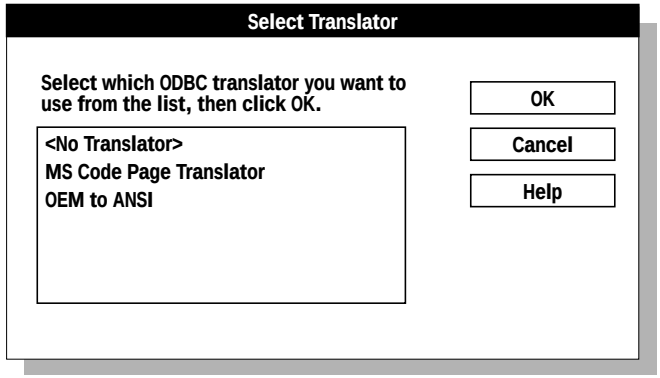


Figure 3-5
The Select Translator Dialog Box

10. Select the translator you want from the list.
11. Click **OK** to return to the ODBC INFORMIX Advanced Driver Setup dialog box, as Figure 3-4 on page 3-12 illustrates.
12. Click **Close** to return to the ODBC INFORMIX Driver Setup dialog box, as Figure 3-3 on page 3-11 illustrates.
13. To accept your specifications for this data source, click **OK**.

    When you click **OK** in the ODBC INFORMIX Driver Setup dialog box, the Data Sources dialog box appears, as Figure 3-1 on page 3-10 illustrates.
14. To add another data source, click **Add**. To exit the Data Sources dialog box, click **Close**.

After you click **OK** in the ODBC INFORMIX Driver Setup dialog box, the ODBC Administrator updates your **ODBC.INI** file (or Registry). The information that you entered in any of the setup dialog boxes becomes the new default data-source connection information for this data source.

Modifying a Data Source

Use the ODBC Administrator to make all modifications to your data source. To make basic modifications to the defaults for a data source, see "Using Dialog Boxes to Connect to a Data Source" on page 16. To make more detailed modifications to a data source, complete the following steps.

To modify a data source

1. Invoke the ODBC Administrator.

   The Data Sources dialog box appears, as Figure 3-6 illustrates.

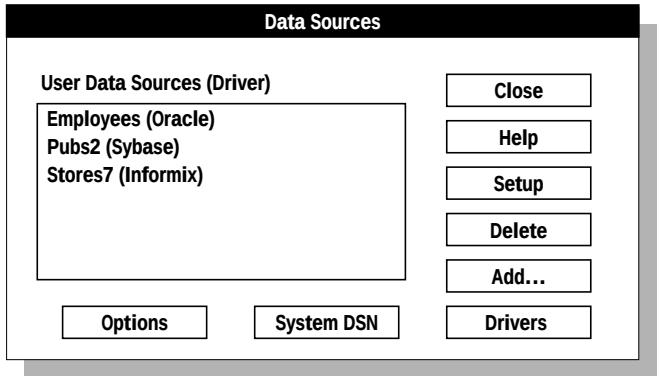


Figure 3-6
The Data Sources Dialog Box

2. In the Data Sources dialog box, select the Informix data source that you want to modify and then click **Setup**.

   The ODBC INFORMIX Driver Setup dialog box appears, as Figure 3-7 illustrates. The values that appear are the default entries specified for this data-source connection.

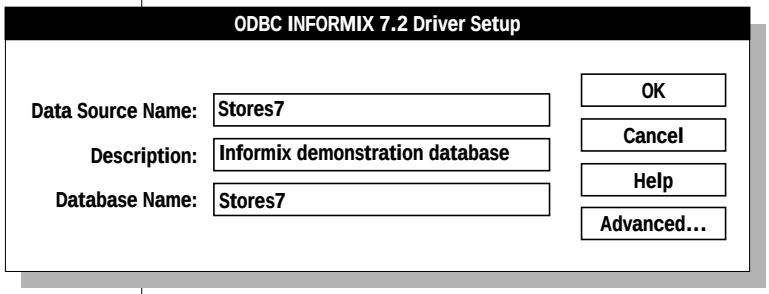


Figure 3-7
A Completed ODBC INFORMIX Driver Setup Dialog Box

3. Modify the applicable data-source text boxes in this dialog box and the Advanced Driver Setup dialog box if necessary. For more information regarding available options, see "Adding a Data Source" on page 3-5.

4. When you are finished, click **OK** in the ODBC INFORMIX Driver Setup dialog box.

   The ODBC Administrator updates the **ODBC.INI** file.

   When you connect to this data source using either a dialog box or connection string, the values that you entered appear as the new default entries for the data-source connection.

Connecting to a Data Source

An Informix ODBC-compliant application can transfer connection information in several ways. For example, the application might have the driver always prompt the user for connection information, or the application might expect a connection string that specifies the data-source connection. How you connect to a data source depends on the connection method that your application uses: both methods are described in the following sections.

Using Dialog Boxes to Connect to a Data Source

One common way applications connect to a data source is through the Data Source dialog box, as Figure 3-8 illustrates. If your application is set up to use a dialog box, the Data Source dialog box appears and prompts you for the appropriate data-source connection information.

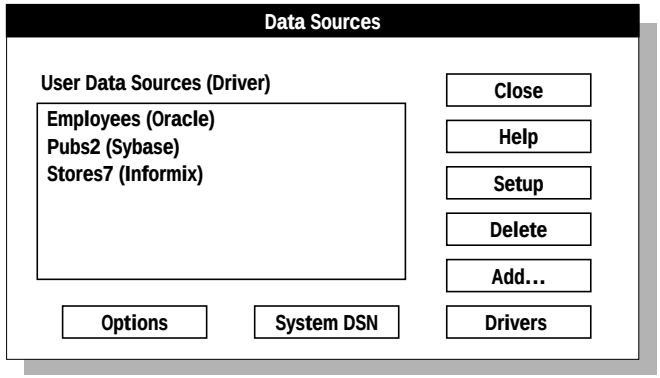


Figure 3-8
The Data Sources Dialog Box

To connect to a data source

1. In the Data Sources dialog box, select an Informix data source.
2. Click **OK**.

   The Logon to Informix dialog box appears, as Figure 3-9 illustrates. The connection information that appears is the default information listed in your **ODBC.INI**.

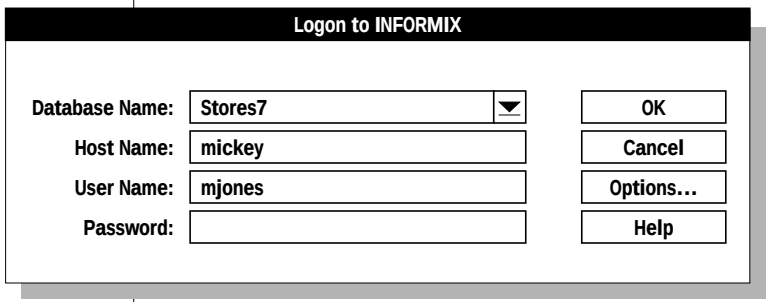


Figure 3-9
The Logon to INFORMIX Dialog Box

3. To accept the default values and connect to the data source, click **OK**.
   or
   To modify the default values for this data source, continue to step 4.
4. In the **Database Name** text box, type the name of the database that contains the tables that you want to access.

   You can also click the down arrow to choose a name from a drop-down list.
5. In the **Host Name** text box, type the name of the computer on which your Informix database server resides.
6. In the **User Name** text box, type your user name as specified on the database server.
7. In the **Password** text box, type your password for the database server to which you want to connect.
8. To display the INFORMIX Server Options dialog box, click **Options**.

   This dialog box, as Figure 3-10 illustrates, lets you specify options that are described in "Optional Data-Source Configuration Information" on page 3-6.

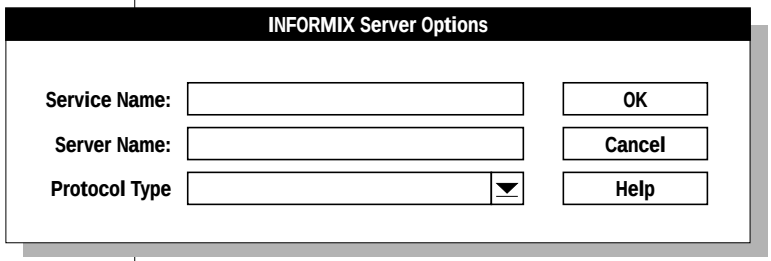


Figure 3-10
The INFORMIX Server Options Dialog Box

9. Click **OK** to accept any changes you make and return to the Logon to Informix dialog box.
10. Click **OK** to connect to the data source and to update these values in **ODBC.INI**.

    Once the connection information is verified, your application can use the Informix driver to access the information that the data source contains.

Using a Connection-String to Connect to a Data Source

An application can also connect to a data source by requiring that you supply a connection string. A connection string consists of a number of attributes that specify how a driver connects to a data source. An attribute identifies a specific piece of information that the driver must know before it can make the appropriate data-source connection.

A connection string has the following format:

```
"DSN=data_source_name[;attribute=value[;attribute=value]...]"
```

You must specify the *data_source_name;* however, all other attributes are optional. If you do not specify an attribute, that attribute defaults to the one that is specified in the Data Source Specification section (for the data source specified in the connection string) of your **ODBC.INI** file in Windows 3.1 or your Registry in Windows NT and Windows 95.

The following table lists the long and short names of the attributes that you can include in the Informix driver connection string. With the exception of the data-source name (which must be specified by DSN), you can use either the long or short names in the connection string. For a description of these attributes, see "Required Data-Source Configuration Information" and "Optional Data-Source Configuration Information" on page 3-6. The attribute long names correspond to the attributes listed in this section.

Attribute Long Name	Attribute Short Name
DataSourceName	DSN
Database	DB
HostName	HOST
LogonID	UID
Password	PWD
Service	SERV
ServerName	SRVR
CursorBehavior	CB

(1 of 2)

Attribute Long Name	Attribute Short Name
EnableInsertCursors	EIC
EnableScrollableCursors	ESC
Protocol	PRO (Windows 3.1 and Windows NT only)
YieldProc	YLD (Windows 3.1 only)

(2 of 2)

The following example shows a valid connection string for connecting to the **stores7** data source. The string specifically requests a connection to the **stores7** database that resides on a computer named **rainbow**. The user ID is **mary**.

```
"DSN=Stores7;DB=stores7;HOST=rainbow;UID=mary;PWD=secret"
```

ODBC Conformance Levels

ODBC defines the API and the SQL conformance standards for drivers. *API conformance* refers to the functions that a driver supports. *SQL conformance* refers to the SQL statements, expressions, and data types that the driver supports. Each conformance standard consists of three levels. For more information on these standards, see "ODBC Conformance Levels" on page 1-11.

API Conformance Level

The Informix driver supports all core and level-1 API functions listed in "API Conformance Levels" on page 1-11.

- The Informix driver supports the following options for the **SQLGetConnectOption** and **SQLSetConnectOption** level-1 functions.

SQL_ACCESS_MODE	SQL_PACKET_SIZE
SQL_AUTOCOMMIT	SQL_TXN_ISOLATION
SQL_ODBC_CURSORS	SQL_TRANSLATE_DLL
SQL_OPT_TRACE	SQL_TRANSLATE_OPTION
SQL_OPT_TRACEFILE	

  The driver supports SQL_ACCESS_MODE for only the **SQLGetConnectOption** function.

- The Informix driver supports the following options for the **SQLSetStmtOption** and **SQLGetStmtOption** level-1 functions.

SQL_BIND_TYPE	SQL_MAX_ROWS
SQL_CONCURRENCY	SQL_NOSCAN
SQL_CURSOR_TYPE	SQL_RETRIEVE_DATA
SQL_KEYSET_SIZE	SQL_ROWSET_SIZE
SQL_MAX_LENGTH	SQL_ROW_NUMBER

In addition, the Informix driver supports the following level-2 functions.

SQLBrowseConnect	SQLNumParams
SQLColumnPrivileges	SQLParamOptions
SQLDataSources	SQLPrimaryKeys
SQLDrivers	SQLProcedures
SQLExtendedFetch	SQLSetScrollOptions
SQLMoreResults	SQLTablePrivileges
SQLNativeSql	

The driver supports scrollable cursors with **SQLExtendedFetch** if the connection attribute EnableScrollableCursors is set to 1.

SQL Conformance Level

The Informix driver meets the core level of SQL conformance that is listed in "SQL Conformance Levels" on page 1-12. In addition, the Informix driver supports the following ODBC extensions to SQL:

- DML statements: outer joins, positioned UPDATE, positioned DELETE, SELECT FOR UPDATE, and UNIONS
- Date and time-stamp data

- The following numeric functions.

abs	acos	asin	atan
atan2	cos	cot	exp
log	log10	mod	power
round	sin	sqrt	tan
truncate			

- The following date functions.

curdate	dayofweek	now
dayofmonth	month	year

- The following string functions.

concat	LTRIM
length	RTRIM

- The following system functions.

database	user

- Procedure calls

For more information on ODBC numeric, date, string, and system functions, see Appendix A, "Scalar Functions."

Mapping Data Types

The Informix driver maps Informix data types to their appropriate ODBC SQL data types. The following table lists the Informix data type and its corresponding ODBC SQL data type.

Informix Data Type	ODBC SQL Data Type
BYTE*	SQL_LONGVARBINARY
CHAR, CHARACTER	SQL_CHAR
DATE	SQL_DATE
DATETIME year to fraction(5)	SQL_TIMESTAMP
DECIMAL, DEC, NUMERIC	SQL_DECIMAL

(1 of 2)

Informix Data Type	ODBC SQL Data Type
FLOAT	SQL_DOUBLE
INTEGER, INT	SQL_INTEGER
INTERVAL	SQL_CHAR
MONEY	SQL_DECIMAL
SERIAL	SQL_INTEGER
SMALLFLOAT, REAL	SQL_REAL
SMALLINT	SQL_SMALLINT
TEXT*	SQL_LONGVARCHAR
VARCHAR,* CHARACTER VARYING*	SQL_VARCHAR

(2 of 2)

* Not supported for INFORMIX-SE databases.

Supported Isolation and Lock Levels

The Informix driver supports isolation-levels 0 (Read Uncommitted), 1 (Read Committed), and 3 (Serializable) if connected to an OnLine database server. The default setting is 1. INFORMIX-SE supports isolation-level 0 (Read Uncommitted) only.

The Informix driver also supports an alternative isolation-level 1, called cursor stability. Your Informix ODBC-compliant application can use this isolation level by calling **SQLSetConnectOption** (1040,1).

Additionally, if transaction logging has not been enabled for your database, then transactions are not supported by the driver (the driver is always in auto-commit mode). Each statement is treated as if it is a single transaction.

The Informix driver also supports page-level locking. For more information on isolation and lock levels, see the *Informix Guide to SQL: Tutorial*.

Chapter

4

Oracle Driver for UNIX

This chapter describes the Oracle driver for UNIX, and explains how to complete the following tasks:

- Set up the Oracle driver
- Add or modify an Oracle data source
- Connect to an Oracle data source

In addition, the chapter includes information on the ODBC API and SQL conformance levels, data-type mappings, and isolation and locking levels that the Oracle driver supports.

Setting Up the Oracle Driver

The Oracle driver is installed during the driver-installation procedure. The name of the current driver is **IXor708**. The driver might have a platform-specific extension; for example, the driver for Solaris is **IXor708.so**.

You must perform the following setup tasks before you can use the driver:

- Confirm that your system has the appropriate software installed.
- Set the **ORACLE_HOME** environment variable to reflect the appropriate directory path.
- Build the Required Oracle SQL*Net Driver

System Requirements

To use the Oracle driver, you must have the following Oracle software installed on your system or network:

- ORACLE7 Server
- ORACLE SQL*Net (Version 1 or Version 2)

For information on these Oracle products, refer to your Oracle documentation set.

Setting Environment Variables

Set the **ORACLE_HOME** environment variable to the fully qualified path of the directory where your ORACLE7 server software is installed. You can set this variable at the command line.

For example, if your Oracle directory path is **/apps/Oracle**, in the C shell you can use the following command:

```
setenv ORACLE_HOME /apps/Oracle
```

In the Bourne or Korn shells, you can use the following command:

```
ORACLE_HOME=/apps/Oracle;export ORACLE_HOME
```

If you set this variable at the command line, you must reset it whenever you log on to your system. To set these variables automatically, add the lines to your **.login** or **.cshrc** files (C shell) or your **.login** or **.profile** (Bourne or Korn shells). As an alternative, you can set these variables in the **.open.csh** (C shell) or **.open.sh** (Bourne or Korn shells) file that is installed with INFORMIX-Open. Source or execute this file to set the environment variables.

For additional setting information, refer to your Oracle client-software documentation.

Important: *All drivers included in INFORMIX-Open use the Informix GLS environment variables. If you want to use a different locale (for example, French), set the DB_LOCALE environment variable to a valid locale name. For more information on valid locales, refer to the "Guide to GLS Functionality." If the data appears to be in a locale that is different from the one that you have set, check that IV_LOCALE, IV_GLS_LCDIR, IV_GLS_REGISTRY, and GL_PATH are not set.*

Building the Required Oracle SQL*Net Driver

For users who are connecting to an Oracle7.1 database server on AIX or Solaris, the Oracle driver requires a one-time site link to build an Oracle SQL*Net driver. This site link binds your unique Oracle SQL*Net configuration into the file, which is used by the Oracle driver to access local and remote Oracle databases. For specific instructions, refer to the INFORMIX-Open machine notes.

Adding and Modifying Data Sources

To connect to a data source, the driver manager looks at your **.odbc.ini** file for specific connection information. This file contains information about each data source. You must modify your **.odbc.ini** file using a text editor.

Adding a Data Source

Before you can connect to a data source, you must add an entry for that data source in your **.odbc.ini** file. For complete information on the format and contents of the **.odbc.ini** file, refer to "Understanding the odbc.ini File" on page 1-5.

Required Data-Source Configuration Information

When you add a data source, you must provide two pieces of information in its Data Source Specification section.

Attribute	Value
Driver	The full path to your driver shared library
ServerName	The SQL*Net connection string that designates the database server and database to be accessed If the SQL*Net connection string contains semicolons, you must enclose it in quotation marks.

All other connection information is optional.

The SQL*Net Connection String

The SQL*Net connection string designates the database server and database to be accessed. For local database servers, the SQL*Net connection string has the following form:

```
ServerName=database_name
```

In this case, *database_name* identifies your Oracle database.

For remote database servers, the information required varies depending on the SQL*Net driver you are using.

To use SQL*Net, Version 1.x

The connection string for SQL*Net, Version 1.x has the following format:

```
ServerName=driver_prefix:computer_name[:sid]
```

The following table shows the meaning of each element.

Element	Description
driver_prefix	A string that indicates the network protocol The driver prefix can be as follows: `P` (named pipes), `X` (SPX), `B` (NetBIOS), `T` (TCP/IP), `D` (DECNet), `A` (Oracle Async), `AT` (AppleTalk), or `TNS` (SQL*Net 2.0). Check your Oracle documentation for other protocols.
computer_name	The name of the Oracle Listener on your network
sid	The Oracle System Identifier This string identifies the instance of Oracle running on the host and is required when an application connects to systems that support more than one instance of an Oracle database.

For example,

```
ServerName=T:mickey:employees
```

In this example, the SQL*Net string specifies that the remote database is called **employees** and that it resides on the host called **mickey**. The string specifies the **TCP/IP** network protocol.

To use SQL*Net, Version 2.x

The connection string for SQL*Net, Version 2.x has the following format:

```
ServerName=oracle_tsn
```

For example,

```
ServerName=bcc_tsn
```

In this example, the SQL*Net string specifies the name of the Oracle service (**bcc_tsn**) that points to the remote database. The database server checks the **tsnnames.ora** file to determine the connection database.

For more information, see your Oracle SQL*Net documentation.

Optional Data-Source Configuration Information

When you add a data source, you can define additional connection defaults. The following table lists the configuration options that the Oracle driver supports. Some of the attributes have default values.

Attribute	Value
Description	A long description that identifies the data source
LogonID	Your user ID as specified on the Oracle database server
LockTimeOut	A value that specifies whether Oracle should wait for a lock to be freed before raising an error when processing a SELECT...For UPDATE statement: ■ 0 = Do no t wait. ■ -1 = Wait. This is the default setting.

(1 of 2)

Attribute	Value
ArraySize	An integer value from 0 to 65536 that specifies the number of bytes the driver uses for fetching multiple rows The default is 60000. Larger values increase throughput by reducing the number of times the driver fetches data across the network. Smaller values increase response time, as there is less delay waiting for the database server to transmit data.
PacketSize	A value that controls the packet size for TCP/IP connections The following integers are the available values: ■ 1024 ■ 2048 ■ 4096 ■ 8192 All other values are ignored. Use this option only when the ServerName option designates T (TCP/IP) as the network protocol.
Catalog Comments	A value that determines whether the driver returns the contents of the COMMENTS column for catalog functions: ■ 0 = Do not return comments. This is the default setting. ■ 1 = Return comments. This setting might affect the performance of your data catalog operations.

(2 of 2)

To add a data source

1. Edit your **.odbc.ini** file using a text editor such as the **vi** editor.

   If you do not have this file in your home directory, copy the default **.odbc.ini** file from the **$INFORMIXDIR/cli** directory into your home directory.

2. Under the ODBC Data Sources section, add an entry for your data source.

   Each entry in this section lists the data source and a description of the driver that the data source uses. Use the following format for data-source entries:

   ```
   [ODBC Data Sources]
   data_source_name=driver_description
   .
   .
   .
   ```

   The following table shows the meaning of each element.

Element	Description
data_source_name	identifies the data source that the Oracle driver accesses. You define the *data_source_name*. The name can be any name that you choose.
driver_description	describes the driver that accesses the data source. This field is optional. Set this field to `Oracle Driver`.

   For identification purposes, you might want make the data-source name the same as the name of the database to which you want to connect by default. Then, when you connect to a data source, you know the specific database to which you are connecting. The following example uses the name **employees**.

   For example, to associate the **employees** data source with the Oracle driver, you would make the following entry in the ODBC Data Sources section of your **.odbc.ini** file:

   ```
   [ODBC Data Sources]
   employees=Oracle Driver
   ```

3. After the ODBC Data Sources section, add an entry for each data source that you specified.

   Each data source listed in the ODBC Data Sources section of your **.odbc.ini** file requires a Data Source Specification. Use the following format for Data Source Specification entries:

```
[data_source_name]
Driver=driver_path
ServerName=sql_net_string
attribute=value
.
.
.
```

   The following table shows the meaning of each element.

Element	Description
data_source_name	is the name of the data source, as specified in the ODBC Data Sources section of your **.odbc.ini** file.
driver_path	is the full path to the ORACLE shared library.
sql_net_string	is the SQL*Net connection string that designates the Oracle database server and database that you want to access. For information on the format of the SQL*Net connection string, see "The SQL*Net Connection String" on page 4-6.
attribute	specifies additional configuration information. For a list of the attributes that the Oracle driver supports, see "Optional Data-Source Configuration Information" on page 4-7.
value	specifies the value for the *attribute*.

   The required fields in this section are *data_source_name*, *driver_path*, and *sql_net_string*.

   The attributes that you enter in this section become the default connection values for this data-source.

Important: *The section name for the Data Source Specification must match the data-source name listed in the ODBC Data Sources section of your* ***.odbc.ini*** *file.*

An entry for the **Employees** data source might look like the following example:

```
[Employees]
Driver=/usr/informix/cli/dlls/IXor708.so
ServerName=T:mickey:employees
LogonID=cindy
```

This example uses SQL*Net, Version 1 syntax.

The following example shows the same entry for the **Employees** data source, this time using SQL*Net, Version 2 syntax:

```
[Employees]
Driver=/usr/informix/cli/dlls/IXor708.so
ServerName=bcc_tsn
LogonID=cindy
```

Tip: *Data-source names are case insensitive; that is,* ***employees*** *and* ***Employees*** *refer to the same data source.*

.odbc.ini Sample Entry for the Oracle Driver

A complete **.odbc.ini** data-source entry for the Employees data source described in "To add a data source" on page 4-9 would look like the following example:

```
[ODBC Data Sources]
employees=Oracle Driver

[Employees]
Driver=/usr/informix/cli/dlls/IXor708.so
ServerName=T:mickey:employees
LogonID=cindy
```

This example uses SQL*Net, Version 1 syntax.

Modifying a Data Source

To edit a data source, use a text editor such as the **vi** editor. Open your **.odbc.ini** file and modify the appropriate lines in that file. The sections that make up this file are described in "Adding a Data Source" on page 4-5.

Connecting to a Data Source

On UNIX, you must supply a connection string to connect to a data source. A connection string consists of a number of attributes that specify how a driver connects to a data source. An attribute identifies a specific piece of information that the driver must know before it can make the appropriate data-source connection.

A connection string has the following format:

```
"DSN=data_source_name[;attribute=value[;attribute=value]...]"
```

You must specify the *data_source_name*; however, all other attributes are optional. If you do not specify an attribute, that attribute defaults to the one that is specified in the Data Source Specification section (for the data source specified in the connection string) of your **.odbc.ini** file.

The following table lists the long and short names of the attributes that you can include in the Oracle driver connection string. With the exception of the data-source name (which must be specified by DSN), you can use either the long or short names in the connection string. For a description of these attributes, see "Adding a Data Source" on page 4-5.

Attribute Long Name	Attribute Short Name
DataSourceName	DSN
ServerName	SRVR
LogonID	UID
Password	PWD
LockTimeOut	LTO
ArraySize	AS
PacketSize	PS
Catalog Comments	CC

The following example shows a valid connection string for connecting to the **Employees** data source. The string (written in SQL*Net, Version 1 syntax) specifically requests a connection to the **employees** database that resides on a database server named **mickey**. The user ID is **cindy**.

```
"DSN=Employees;SRVR=T:mickey:employees;UID=cindy;PWD=secret"
```

ODBC Conformance Levels

ODBC defines the API and the SQL conformance standards for drivers. *API conformance* refers to the functions that a driver supports. *SQL conformance* refers to the SQL statements, expressions, and data types that the driver supports. Each conformance standard consists of three levels. For more information on these standards, see "ODBC Conformance Levels" on page 1-11.

API Conformance Level

The Oracle driver supports the core and level-1 API functions listed in "API Conformance Levels" on page 1-11.

- The Oracle driver supports the following options for the **SQLGetConnectOption** and **SQLSetConnectOption** level-1 functions.

SQL_ACCESS_MODE	SQL_TRANSLATE_DLL
SQL_AUTOCOMMIT	SQL_TRANSLATE_OPTION
SQL_OPT_TRACE	SQL_TXN_ISOLATION
SQL_OPT_TRACEFILE	

  The driver supports SQL_ACCESS_MODE for only the **SQLGetConnectOption** function.

- The Oracle driver supports the following options for the **SQLSetStmtOption** and **SQLGetStmtOption** level-1 functions.

SQL_BIND_TYPE	SQL_NOSCAN
SQL_CONCURRENCY	SQL_RETRIEVE_DATA
SQL_CURSOR_TYPE	SQL_ROWSET_SIZE
SQL_KEYSET_SIZE	SQL_ROW_NUMBER
SQL_MAX_LENGTH	SQL_SIMULATE_CURSOR
SQL_MAX_ROWS	

In addition, the Oracle driver supports the following level-2 functions.

SQLBrowseConnect	SQLNumParams
SQLDataSources	SQLParamOptions
SQLDrivers	SQLPrimaryKeys
SQLExtendedFetch (forward)	SQLProcedureColumns
SQLForeignKeys	SQLProcedures
SQLMoreResults	SQLSetScrollOptions
SQLNativeSql	

SQL Conformance Level

The Oracle driver meets the core level of SQL conformance that is listed in "SQL Conformance Levels" on page 1-12. In addition to the ODBC extensions supported by the ORACLE7 Server, the Oracle driver supports the following ODBC extensions to SQL:

- Date, time, and time-stamp data
- Outer joins
- The following numeric functions.

abs	floor	power	sqrt
ceiling	log	round	tan
cos	log10	sign	truncate
exp	mod	sin	

- The following date functions.

curdate	dayofyear	monthname	week
dayname	hour	now	year
dayofmonth	minute	quarter	
dayofweek	month	second	

- The following string functions.

ascii	left	repeat	space
char	length	replace	substring
concat	locate	right	ucase
insert	locate _2	rtrim	
lcase	ltrim	soundex	

- The following system functions.

ifnull	user

For more information on ODBC numeric, date, string, and system functions, see Appendix A, "Scalar Functions."

Mapping Data Types

The Oracle driver maps ORACLE7 Server data types to their appropriate ODBC SQL data types. The following table lists the ORACLE7 Server data type and its corresponding ODBC SQL data type.

ORACLE7 Server Data Type	ODBC SQL Data Type
CHAR	SQL_CHAR
DATE	SQL_TIMESTAMP
LONG	SQL_LONGVARCHAR
LONG RAW	SQL_LONGVARBINARY
NUMBER	SQL_DOUBLE
NUMBER (p,s)	SQL_DECIMAL
RAW	SQL_VARBINARY
VARCHAR2	SQL_VARCHAR

Important: *The Oracle driver does not support the FLOAT or the MLSLABEL ORACLE7 data types.*

Supported Isolation and Lock Levels

The Oracle driver supports isolation-level 2 (repeatable read) only. The driver supports record-level locking. For more information on isolation and locking levels, see the *Informix Guide to SQL: Tutorial* and your Oracle database server documentation.

Chapter

5

Oracle Driver for Windows

This chapter describes the Oracle driver for Windows environments, and explains how to complete the following tasks:

- Set up the Oracle driver
- Add or modify an Oracle data source
- Connect to an Oracle data source

In addition, the chapter includes information on the ODBC API and SQL conformance levels, data-type mappings, and isolation and locking levels that the Oracle driver supports.

Setting Up the Oracle Driver

The Oracle driver is installed during the driver-installation procedure. For Windows 3.1, the current driver is **IVOR709.DLL**. For Windows NT and Windows 95, the current driver is **IXOR709.DLL**.

You must perform the following setup tasks before you can use the driver:

- Confirm that your system has the appropriate software installed.
- Confirm that your environment-variable settings reflect the appropriate directory paths.

System Requirements

To use the Oracle driver, you must have the following Oracle products installed on your system or network:

- ORACLE7 Server
- ORACLE SQL*Net (Version 1 or Version 2)

For information on these Oracle products, refer to your Oracle documentation set.

Setting Environment Variables

INFORMIX-Open provides double-byte character support through standard language support in Windows environments. (You must have a double-byte, localized Microsoft Windows evironment.) Before you use the Informix driver, ensure that you configure the Windows environment for the language and cultural conventions that you require. For Windows 3.1 and Windows NT, in the **Program Manager**, choose **Main→Control Panel→International** to set these conventions. For Windows 95, set the appropriate language in the Control Panel, Regional Settings dialog box.

For additional setting information, refer to your Oracle client-software documentation.

Adding and Modifying Data Sources

To add and configure data sources, use the ODBC Administrator. The ODBC Administrator then updates your **ODBC.INI** file in Windows 3.1 or your Registry in Windows 95 and Windows NT to reflect your data-source connection information.

Adding a Data Source

Windows3.1

In Windows 3.1, the **ODBC.INI** file is an initialization file used by the ODBC drivers and is located in the **WINDOWS** directory. This file contains information about each data source and its associated driver. Before you can connect to a data source, you must add the connection information for the data source to the **ODBC.INI** file. The ODBC Administrator adds the information to this file for you; never modify the **ODBC.INI** file directly. For complete information on the format and contents of this file, refer to "Understanding the odbc.ini File" on page 1-5. ♦

Windows NT

Windows 95

In Windows NT and Windows 95, **ODBC.INI** is a subkey of the HKEY_CURRENT_USER key within the Registry. When you access the Registry using this subkey, the ODBC structure is the same as the structure described in the "odbc.ini File Format" on page 1-6. Before you can connect to a data source, you must use the ODBC Administrator to add the connection information for the data source to the Registry. ♦

Required Data-Source Configuration Information

When you add a data source, you must provide two pieces of information.

Attribute	Value
Data Source Name	The name of the data source you want to access
ServerName	The SQL*Net connection string that designates the database server and database to be accessed If the SQL*Net connection string contains semicolons, you must enclose it in quotation marks.

All other connection information is optional.

*The SQL*Net Connection String*

The SQL*Net connection string designates the database server and database to be accessed. For local database servers, the SQL*Net connection string has the following form:

```
ServerName=database_name
```

In this case, *database_name* identifies your Oracle database.

For remote database servers, the information required varies depending on the SQL*Net driver you are using.

To use SQL*Net, Version 1.x

The connection string for SQL*Net, Version 1.x has the following format:

```
ServerName=driver_prefix:computer_name[:sid]
```

The following table shows the meaning of each element.

Element	Description
driver_prefix	A string that indicates the network protocol The driver prefix can be as follows: `P` (named pipes), `X` (SPX), `B` (NetBIOS), `T` (TCP/IP), `D` (DECNet), `A` (Oracle Async), `AT` (AppleTalk), or `TNS` (SQL*Net 2.0). Check your Oracle documentation for other protocols.
computer_name	The name of the Oracle Listener on your network
sid	The Oracle System Identifier This string identifies the instance of Oracle running on the host and is required when an application connects to systems that support more than one instance of an Oracle database.

For example,

```
ServerName=T:mickey:employees
```

In this example, the SQL*Net string specifies that the remote database is called **employees** and that it resides on the host called **mickey**. The string specifies the **TCP/IP** network protocol.

To use SQL*Net, Version 2.x

The connection string for SQL*Net, Version 2.x has the following format:

```
ServerName=oracle_tsn
```

For example,

```
ServerName=bcc_tsn
```

In this example, the SQL*Net string specifies the name of the Oracle service (**bcc_tsn**) that points to the remote database. The database server checks the **TSNNAMES.ORA** file to determine the connection database.

For more information, see your Oracle SQL*Net documentation.

Optional Data-Source Configuration Information

When you add a data source, you can define additional connection defaults. The following table lists the configuration options that the Oracle driver supports. Some of the attributes have default values.

Attribute	Value
Description	A long description that identifies the data source
Default User Name	Your user ID as specified on the Oracle database server
Server List	The list of SQL*Net connection strings that will appear in the logon dialog box If more than one name is specified, the names must be separated by commas. If an SQL*Net connection string contains a comma, enclose it in quotation marks; for example, “Serv,1”, “Serv,2”, “Serv,3.”
Lock Time Out	A value that specifies whether Oracle should wait for a lock to be freed before raising an error when processing a SELECT...For UPDATE statement: ■ 0 = Do not wait. ■ 1 = Wait. This is the default setting.

(1 of 2)

Attribute	Value
Array Size	An integer value from 0 to 65536 that specifies the number of bytes the driver uses for fetching multiple rows The default is 60000. Larger values increase throughput by reducing the number of times the driver fetches data across the network. Smaller values increase response time, as there is less delay waiting for the database server to transmit data.
Packet Size	A value that controls the packet size for TCP/IP connections The following integers are the available values: ■ 1024 ■ 2048 ■ 4096 ■ 8192 All other values are ignored. Use this option only when the ServerName option designates T (TCP/IP) as the network protocol.
Catalog Comments	A value that determines whether the driver returns the contents of the COMMENTS column for catalog functions: ■ 0 = Don't return comments. This is the default setting. ■ 1 = Return comments. This setting might affect the performance of your data catalog operations. Check the box to change the setting to 1.

(2 of 2)

Warning: *Windows 3.1 users should never modify the* ***ODBC.INI*** *file directly. To modify the contents of this file, use the ODBC administrator. If you modify the* ***ODBC.INI*** *file directly, data-source configuration errors might result.*

To add a data source

1. Invoke the ODBC Administrator.

   The Data Sources dialog box appears, as Figure 5-1 illustrates.

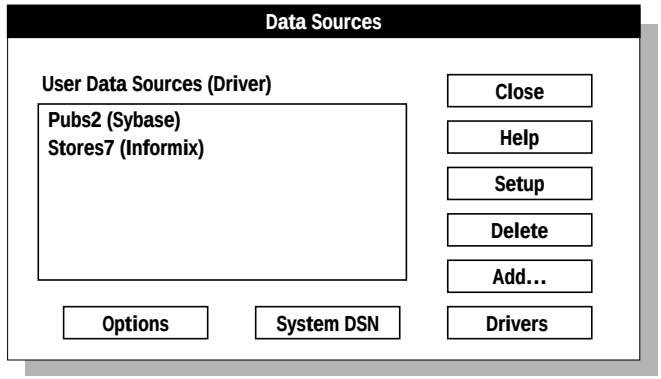


Figure 5-1
The Data Sources Dialog Box

2. Click **Add**.

   The Add Data Source dialog box appears, as Figure 5-2 illustrates.

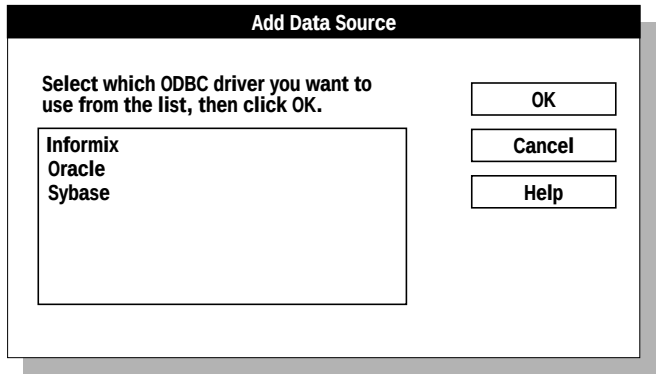


Figure 5-2
The Add Data Source Dialog Box

3. Select the Oracle driver from the Installed ODBC Drivers list.
4. Click **OK**.

   The ODBC Oracle Driver Setup dialog box appears, as Figure 5-3 illustrates.

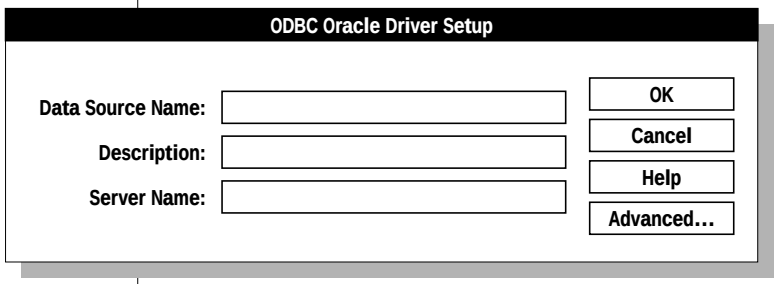


Figure 5-3
The ODBC Oracle Driver Setup Dialog Box

5. In the **Data Source Name** text box, type the name of the data source you want to access.

   You define the data-source name; that is, it can be any name that you choose.
6. In the **Description** text box, type a long description of your data source. This section is optional.
7. In the **Server Name** text box, type the SQL*Net connection string that designates the database server and database to which you want to connect by default.

   For information on the format of the SQL*Net connection string, see "The SQL*Net Connection String" on page 5-5.

   You now have entered enough information to be able to connect to the data source:

   - To add the data source, click **OK.**
     The Data Sources dialog box appears, as Figure 5-1 on page 5-9 illustrates. You can then click **Add** to add another data source or click **Close** to exit the Data Sources dialog box.
   - To add optional connection information about the data source, click **Advanced**.
     The ODBC Oracle Advanced Driver Setup dialog box appears, as Figure 5-4 illustrates.

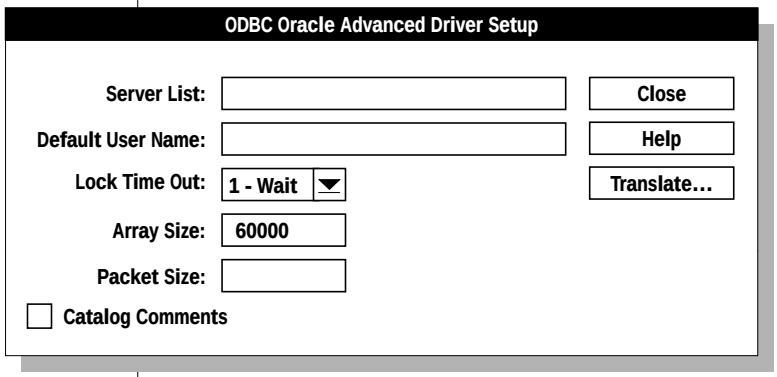


Figure 5-4 *The ODBC Oracle Advanced Driver Setup Dialog Box*

8. Enter the information you want to specify about this data source. You can also choose an option from the drop-down list.

   For a description of the advanced connection options that are available, see "Optional Data-Source Configuration Information" on page 5-7.

9. To specify a translator, click **Translate**.

   The Translator dialog box appears, as Figure 5-5 illustrates.

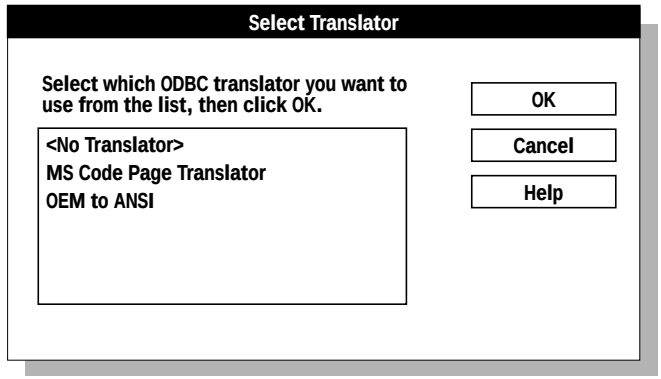


Figure 5-5 *The Translator Dialog Box*

10. Select the translator you want from the list.
11. Click **OK** to return to the ODBC Oracle Advanced Driver Setup dialog box.

12. Click **Close** to return to the ODBC Oracle Driver Setup dialog box.
13. To accept your specifications for this data source, click **OK**.

    When you click **OK** in the ODBC Oracle Driver Setup dialog box, the Data Sources dialog box appears, as shown in Figure 5-1 on page 5-9.
14. To add another data source, click **Add**. To exit the Data Sources dialog box, click **Close**.

After you click **OK** in the ODBC Oracle Driver Setup dialog box, the ODBC Administrator updates your **ODBC.INI** file (or Registry). The information that you entered in any of the setup dialog boxes becomes the new default data-source connection information for this data source.

Modifying a Data Source

Use the ODBC Administrator to make all modifications to your data source. To make basic modifications to the defaults for a data source, see "Using Dialog Boxes to Connect to a Data Source" on page 5-14. To make more detailed modifications to a data source, complete the following steps.

To modify a data source

1. Invoke the ODBC Administrator.

   The Data Sources dialog box appears, as Figure 5-6 illustrates.

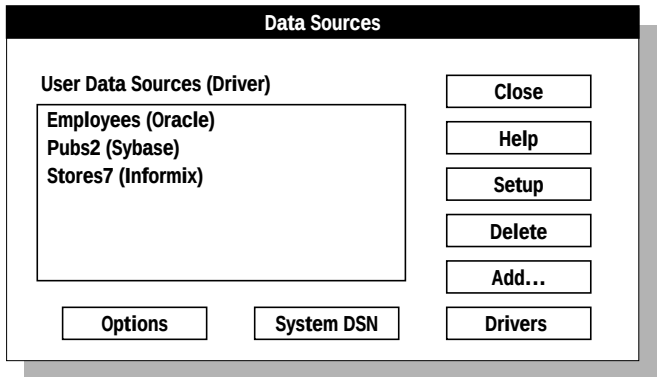


Figure 5-6
The Data Sources Dialog Box

2. In the Data Sources dialog box, select the Oracle data source that you want to modify and then click **Setup**.

   The ODBC Oracle Driver Setup dialog box appears, as Figure 5-7 illustrates. The values that appear are the default entries specified for this data-source connection.

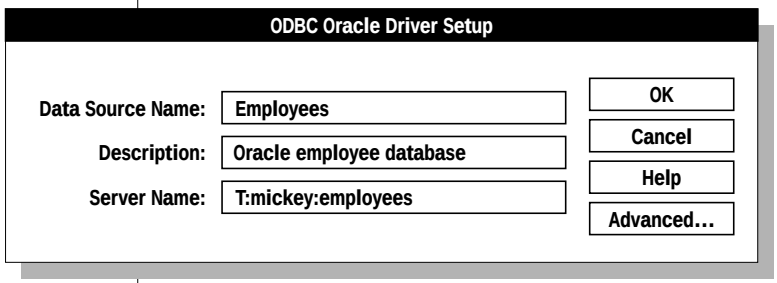


Figure 5-7
A Completed ODBC Oracle Driver Setup Dialog Box

3. Modify the applicable data-source text boxes in this dialog box and the ODBC Oracle Advanced Driver Setup dialog box if necessary.

   For more information regarding available options, see "Adding a Data Source" on page 5-5.

4. When you are finished, click **OK** in the ODBC Oracle Driver Setup dialog box.

   The ODBC Administrator updates your **ODBC.INI** file.

   When you connect to this data source using either a dialog box or connection string, the values that you entered appear as the new default entries for the data-source connection.

Connecting to a Data Source

An Informix ODBC-compliant application can transfer connection information in a number of ways. For example, the application might have the driver always prompt the user for connection information, or the application might expect a connection string that specifies the data-source connection. How you connect to a data source depends on the connection method your application uses.

Using Dialog Boxes to Connect to a Data Source

One common way of connecting to a data source is through the Data Source dialog box, as Figure 5-8 illustrates. If your application is set up to use a dialog box, the Data Source dialog box is displayed and prompts you for the appropriate data-source connection information.

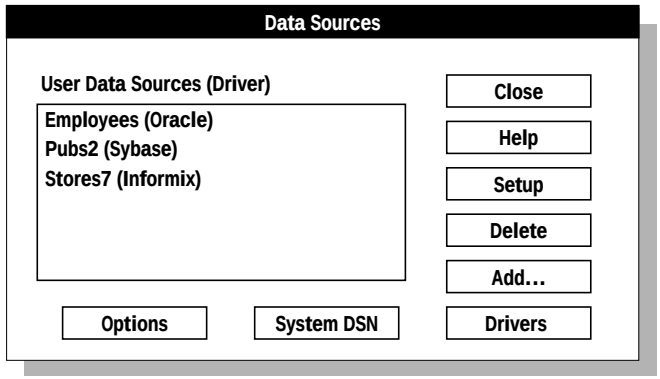


Figure 5-8
The Data Sources Dialog Box

To connect to a data source

1. In the Data Source dialog box, select an Oracle data source.
2. Click **OK**.

   The Logon to Oracle dialog box appears, as Figure 5-9 illustrates.The connection information that appears is the default information listed in your **ODBC.INI**.

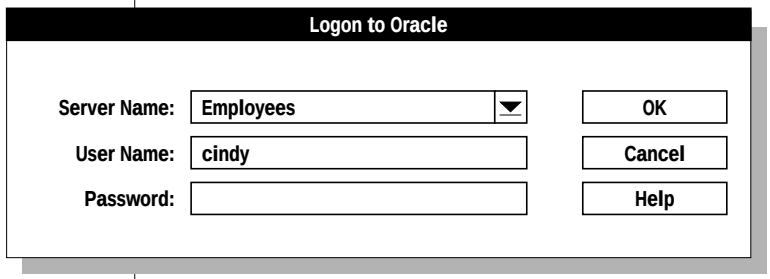


Figure 5-9
The Logon to Oracle Dialog Box

3. To accept the default values and connect to the data source, click **OK**.
   or
   To modify the default values for this data source, continue to step 4.
4. In the **Server Name** text box, type the SQL*Net connection string for the Oracle database that contains the tables that you want to access.

   You can also click the down arrow to choose a name from a drop-down list. For information on the format of the SQL*Net connection string, see "Required Data-Source Configuration Information" on page 5-5.
5. In the **User Name** text box, type your user name as specified on the database server.
6. In the **Password** text box, type your password for the database server to which you want to connect.
7. Click **OK** to connect to the data source and to update these values in **ODBC.INI**.

   Once the connection information is verified, your application can use the Oracle driver to access the information that the data source contains.

Using a Connection-String to Connect to a Data Source

Some applications require that you connect to a data source by typing a connection string. A connection string consists of a number of attributes that specify how a driver connects to a data source. An attribute identifies a specific piece of information that the driver must know before it can make the appropriate data-source connection.

A connection string has the following format:

```
"DSN=data_source_name[;attribute=value[;attribute=value]...]"
```

You must specify the *data_source_name*; however, all other attributes are optional. If you do not specify an attribute, that attribute defaults to the one that is specified in the Data Source Specification section (for the data source specified in the connection string) of your **ODBC.INI** file in Windows 3.1 or your Registry in Windows NT and Windows 95.

The following table lists the long and short names of the attributes that you can include in the Oracle driver connection string. With the exception of the data-source name (which must be specified by `DSN`), you can use either the long or short names in the connection string. For a description of these attributes, see "Adding a Data Source" on page 5-5.

Attribute Long Name	Attribute Short Name
DataSourceName	DSN
ServerName	SRVR
LogonID	UID
Password	PWD
LockTimeOut	LTO
ArraySize	AS
PacketSize	PS
Catalog Comments	CC

For example, a connection string that connects to the **Employees** data source using the **employees** database server might have the following attributes in the connection string:

```
"DSN=Employees;SRVR=T:mickey:employees;UID=cindy;PWD=secret"
```

This example uses SQL*Net, Version 1 syntax. For more information on SQL*Net, Version 2 Syntax, see "The SQL*Net Connection String" on page 5-5.

ODBC Conformance Levels

ODBC defines the API and the SQL conformance standards for drivers. *API conformance* refers to the functions that a driver supports. *SQL conformance* refers to the SQL statements, expressions, and data types that the driver supports. Each conformance standard consists of three levels. For more information on these standards, see "ODBC Conformance Levels" on page 1-11.

API Conformance Level

The Oracle driver supports the core and level-1 API functions listed in "API Conformance Levels" on page 1-11.

- The Oracle driver supports the following options for the **SQLGetConnectOption** and **SQLSetConnectOption** level-1 functions.

SQL_ACCESS_MODE	SQL_QUIETMODE
SQL_AUTOCOMMIT	SQL_TRANSLATE_DLL
SQL_ODBC_CURSORS	SQL_TRANSLATE_OPTION
SQL_OPT_TRACE	SQL_TXN_ISOLATION
SQL_OPT_TRACEFILE	

  The driver supports SQL_ACCESS_MODE for only the **SQLGetConnectOption** function.

- The Oracle driver supports the following options for the **SQLSetStmtOption** and **SQLGetStmtOption** level-1 functions.

SQL_BIND_TYPE	SQL_NOSCAN
SQL_CONCURRENCY	SQL_RETRIEVE_DATA
SQL_CURSOR_TYPE	SQL_ROWSET_SIZE
SQL_KEYSET_SIZE	SQL_ROW_NUMBER
SQL_MAX_LENGTH	SQL_SIMULATE_CURSOR
SQL_MAX_ROWS	

In addition, the Oracle driver supports the following level-2 functions.

SQLBrowseConnect	SQLNumParams
SQLDataSources	SQLParamOptions
SQLDrivers	SQLPrimaryKeys
SQLExtendedFetch (forward)	SQLProcedureColumns
SQLForeignKeys	SQLProcedures
SQLMoreResults	SQLSetScrollOptions
SQLNativeSql	

SQL Conformance Level

The Oracle driver meets the core level of SQL conformance that is listed in "SQL Conformance Levels" on page 1-12. In addition to the ODBC extensions supported by the ORACLE7 Server, the Oracle driver supports the following ODBC extensions to SQL:

- Date, time, and time-stamp data
- Outer joins
- The following numeric functions.

abs	floor	power	sqrt
ceiling	log	round	tan
cos	log10	sign	truncate
exp	mod	sin	

- The following date functions.

curdate	dayofyear	monthname	week
dayname	hour	now	year
dayofmonth	minute	quarter	
dayofweek	month	second	

- The following string functions.

ascii	left	repeat	space
char	length	replace	substring
concat	locate	right	ucase
insert	locate _2	rtrim	
lcase	ltrim	soundex	

- The following system functions.

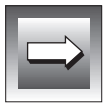
ifnull	user

For more information on ODBC numeric, date, string, and system functions, see Appendix A, "Scalar Functions."

Mapping Data Types

The Oracle driver maps ORACLE7 Server data types to their appropriate ODBC SQL data types. The following table lists the ORACLE7 Server data type and its corresponding ODBC SQL data type.

ORACLE7 Server Data Type	ODBC SQL Data Type
CHAR	SQL_CHAR
DATE	SQL_TIMESTAMP
LONG	SQL_LONGVARCHAR
LONG RAW	SQL_LONGVARBINARY
NUMBER	SQL_DOUBLE
NUMBER (p,s)	SQL_DECIMAL
RAW	SQL_VARBINARY
VARCHAR2	SQL_VARCHAR



Important: *The Oracle driver does not support the FLOAT or the MLSLABEL ORACLE7 data types.*

Supported Isolation and Lock Levels

The Oracle driver supports isolation-level 2 (repeatable read) only. The driver supports record-level locking. For more information on isolation and locking levels, see the *Informix Guide to SQL: Tutorial* and your Oracle database server documentation.

Chapter

6

Sybase Driver for UNIX

This chapter describes the Sybase driver for UNIX, and explains how to complete the following tasks:

- Set up the Sybase driver
- Add or modify a Sybase data source
- Connect to a Sybase data source

In addition, the chapter includes information on the ODBC API and SQL conformance levels, data-type mappings, and isolation and locking levels that the Sybase driver supports.

Setting Up the Sybase Driver

The Sybase driver is installed during the driver-installation procedure. The name of the current driver is **IXsyb08**. The driver might have a platform-specific extension; for example, the driver for Solaris is **IXsyb08.so**.

You must perform the following setup tasks before you can use the driver:

- Confirm that your system has the appropriate software installed.
- Set environment variables to reflect the appropriate directory paths.
- Build the Required Sybase Open Client Net Libraries

System Requirements

To use the Sybase driver, you must have the following Sybase software installed on your system or network:

- SYBASE Open Client (Version 10.x)
- SYBASE System 10 or SYBASE System 11 (database server)

For information on these Sybase products, refer to your Sybase documentation set.

Setting Environment Variables

Specify the directory that contains the shared libraries of your database server in a platform-specific environment variable. For information on setting platform-specific environment variables, refer to the INFORMIX-Open machine notes.

In addition, set the **SYBASE** environment variable to the fully qualified path of the directory where your Sybase product is installed. You can set this variable at the command line. For example, if your Sybase directory path is **/apps/sybase10**, in the C shell, you can use the following command:

```
setenv SYBASE /apps/sybase10
```

In the Bourne or Korn shells, you can use the following command:

```
SYBASE=/apps/sybase10; export SYBASE
```

If you set these variables at the command line, you must reset them whenever you log on to your system. To set these variables automatically, add the lines to your **.login** or **.cshrc** files (C shell) or your **.login** or **.profile** (Bourne or Korn shells). As an alternative, you can set these variables in the **.open.csh** (C shell) or **.open.sh** (Bourne or Korn shells) file that is installed with INFORMIX-Open. Source or execute this file to set the environment variables.

For additional setting information, refer to your Sybase client-software documentation.

Important: *All drivers included in INFORMIX-Open use the Informix GLS environment variables. If you want to use a different locale (for example, French), set the DB_LOCALE environment variable to a valid locale name. For more information on valid locales, refer to the "Guide to GLS Functionality." If the data appears to be in a locale that is different from the one that you have set, check that IV_LOCALE, IV_GLS_LCDIR, IV_GLS_REGISTRY, and GL_PATH are not set.*

Building the Required Sybase Open Client Net-Libraries

For AIX, the Sybase driver requires a one-time site link to build the Sybase Open Client Net-Libraries. For specific instructions, refer to the INFORMIX-Open machine notes.

Adding and Modifying Data Sources

To connect to a data source, the driver manager looks at your **.odbc.ini** file for specific connection information. This file contains information about each data source. You must modify your **.odbc.ini** file using a text editor.

Adding a Data Source

Before you can connect to a data source, you must add an entry for that data source in your **.odbc.ini** file. For complete information on the format and contents of the **.odbc.ini** file, refer to "Understanding the odbc.ini File" on page 1-5.

Required Data-Source Configuration Information

When you add a data source, you must provide two pieces of information in its Data Source Specification section.

Attribute	Value
Driver	The full path to your driver shared library
ServerName	The name of the database server to which you want to connect by default If you do not supply a database server name, the driver uses the name of a database server from your **$SYBASE/interfaces** file.

All other connection information is optional.

Optional Data-Source Configuration Information

When you add a data source, you can define additional connection defaults. You can specify two types of connection options in the Data-Source Specification section:

- Options that define names
- Options that define performance settings

Options That Define Names

The following table lists the names that you can set as default connection options for a data source. Some of the attributes have default values.

Attribute	Value
Description	A long description that identifies the data source
Database	The name of the database to which you want to connect by default If you do not specify a value, the default is the database defined by the system administrator for each user.
LogonID	Your user name as specified on the Sybase database server

(1 of 2)

Attribute	Value
InterfacesFile	The pathname of the interfaces file The default is the normal Sybase interfaces file.
PasswordEncryption	A value that determines whether password encryption can be performed from the Open Client Library to the database server: ■ 0 = Do not enable password encryption. This is the default setting. ■ 1 = Enable password encryption.
Charset	The name of a character set corresponding to a subdirectory in $SYBASE/charsets The default is the setting on the database server.
WorkstationID	The workstation ID used by the client
Language	The national language corresponding to a subdirectory in $SYBASE/locales The default is English.
ApplicationName	The name used by the database server to identify your application

(2 of 2)

Options That Define Performance Settings

The following table lists the types of operation behavior that you can set as default connection options for a data source. The table lists the default values.

Attribute	Value
OptimizePrepare	A value that determines whether stored procedures are created on the database server for every call to SQLPrepare: ■ 0 = Create stored procedures for every call to SQLPrepare. This setting can result in bad performance. ■ 1 = Create stored procedures only if the statement contains parameters. Otherwise, cache the statement and execute directly at SQLExecute time. This is the default setting. ■ 2 = Do not create stored procedures.
ArraySize	A value that specifies the number of rows the driver retrieves when fetching from the database server This value is not the number of rows given to the user. The default is 10.
EnableQuotedIdentifier	A value that determines whether quoted identifier strings are allowed ■ 0 = No quoted identifier strings. This is the default setting. ■ 1 = Enable quoted identifier strings.
SelectMethod	A value that determines whether database cursors are used for SELECT statements: ■ 0 = Use database cursors. This is the default setting. ■ 1 = Execute SELECT statements directly without using database cursors. This setting limits the data source to one active statement and one active connection.

(1 of 2)

Attribute	Value

PacketSize	A value of `-1`, `0`, or *x* that determines the number of bytes per network packet transferred from the database server to the client:
	■ `0` = Use the default packet size as specified in the Sybase database server configuration. This is the default setting.
	■ `1` = Compute the maximum-allowable packet size on the first connect to the data source and save the value in the **.odbc.ini** file.
	■ *x* = Set the packet size to *x* `* 512` bytes where *x* is an integer from `1` to `10`. (For example, if the value were `6`, the packet size would be set to `6 * 512 or 3072` bytes.)
	To take advantage of this connection attribute, configure the Sybase database server for a maximum-network packet size greater than or equal to the value you specified for PacketSize. For example,
	`sp_configure "maximum network packet size", 5120`
	`reconfigure`
	`Restart System 10 Server`
	The ODBC specification identifies a connect option, SQL_PACKET_SIZE, that offers this same functionality. To avoid conflicts with applications that might set both this connection string attribute and the ODBC connect option, the two have been defined as mutually exclusive. If PacketSize is specified, you will receive a message "Driver Not Capable" if you attempt to call SQL_PACKET_SIZE. If you do not set PacketSize, then application calls to SQL_PACKET_SIZE are accepted by the driver.

CursorCacheSize	A value that determines the number of connections that the connection cache can hold The default setting is `1`. To set this option, you must set the SelectMethod option to `1`. Increasing the Cursor-CacheSize might increase the performance of some applications, but requires additional database resources.

(2 of 2)

To add a data source

1. Edit your **.odbc.ini** file using a text editor such as the **vi** editor.

   If you do not have this file in your home directory, copy the default **.odbc.ini** file from the **$INFORMIXDIR/cli** directory into your home directory.

2. Under the ODBC Data Sources section, add an entry for your data source.

   Each entry in this section lists the data source and a description of the driver that the data source uses. Use the following format for data-source entries:

   ```
   [ODBC Data Sources]
   data_source_name=driver_description
   .
   .
   .
   ```

   The following table shows the meaning of each element.

Element	Description
data_source_name	identifies the data source that the Sybase driver accesses. You define the *data_source_name*. The name can be any name that you choose.
driver_description	describes the driver that accesses the data source. This field is optional. Set this field to `Sybase Driver`.

   For identification purposes, you might want make the data-source name the same as the name of the database to which you want to connect by default. Then, when you connect to a data source, you know the specific database to which you are connecting. The following example uses the name **Pubs2**.

   For example, to associate the **Pubs2** data source with the Sybase driver, you would make the following entry in the Data Sources section of your **.odbc.ini** file:

   ```
   [ODBC Data Sources]
   Pubs2=Sybase Driver
   ```

3. After the ODBC Data Sources section, add an entry for each data source that you specified.

   Each data source listed in the ODBC Data Sources section of your **.odbc.ini** file requires a Data Source Specification. Use the following format for Data Source Specification entries:

```
[data_source_name]
Driver=driver_path
ServerName=server_name
DatabaseName=database_name
attribute=value
.
.
.
```

   The following table shows the meaning of each element.

Element	Description
data_source_name	is the name of the data source, as specified in the ODBC Data Sources section of your **.odbc.ini** file.
driver_path	is the full path to the Sybase shared library.
server_name	is the name of the Sybase database server that contains the tables that you want to access.
database_name	is the name of the database that the data source accesses.
attribute	specifies optional configuration information. For a list of the attributes that the Sybase driver supports, see "Optional Data-Source Configuration Information" on page 6-6.
value	specifies the value for the attribute.

   The required fields in this section are *data_source_name*, *driver_path* and *server_name*."DatabaseName" is an attribute that is used to define optional connection information.

   The attributes that you enter in this section become the default connection values for this data-source.

Important: *The section name for the Data Source Specification must match the data-source name listed in the ODBC Data Sources section of your* ***.odbc.ini*** *file.*

For example, an entry for the **Pubs2** data source might look like the following example:

```
[Pubs2]
Driver=/usr/informix/cli/dlls/IXsyb08.so
ServerName=Cyber10
Database=pubs2
LogonID=mary
```

In this example, the data source and database have the same name, **pubs2**.

Tip: *Data-source names are case insensitive; that is,* ***pubs2*** *and* ***Pubs2*** *refer to the same data source.*

.odbc.ini Sample Entry for the Sybase Driver

A complete **.odbc.ini** data-source entry for the **Pubs2** data source described in "To add a data source" on page 6-10 would look like the following example:

```
[ODBC Data Sources]
Pubs2=Sybase Driver

[Pubs2]
Driver=/usr/informix/cli/dlls/IXsyb08.so
ServerName=Cyber10
DatabaseName=pubs2
LogonID=mary
```

Modifying a Data Source

To edit a data source, use a text editor such as the **vi** editor. Open your **.odbc.ini** file and modify the appropriate lines in that file. The sections that make up this file are described in "To add a data source" on page 6-10.

Connecting to a Data Source

On UNIX, you must supply a connection string to connect to a data source. A connection string consists of a number of attributes that specify how a driver connects to a data source. An attribute identifies a specific piece of information that the driver must know before it can make the appropriate data-source connection.

A connection string has the following format:

```
"DSN=data_source_name[;attribute=value[;attribute=value]...]"
```

You must specify the *data_source_name*; however, all other attributes are optional. If you do not specify an attribute, that attribute defaults to the one that is specified in the Data Source Specification section (for the data source specified in the connection string) of your **.odbc.ini** file.

The following table lists the long and short names of the attributes that you can include in the Sybase driver connection string. With the exception of the data-source name (which must be specified by DSN), you can use either the long or short names in the connection string. For a description of these attributes, see "Optional Data-Source Configuration Information" on page 6-6.

Attribute Long Name	Attribute Short Name
DataSourceName	DSN
ServerName	SRVR
LogonID	UID
Password	PWD
Database	DB
Language	LANG
Charset	CS

(1 of 2)

Attribute Long Name	Attribute Short Name
WorkstationID	WKID
ApplicationName	APP
InterfacesFile	IFILE
OptimizePrepare	OP
SelectMethod	SM
ArraySize	AS
PasswordEncryption	PE
PacketSize	PS
CursorCacheSize	CCS

(2 of 2)

The following example shows a valid connection string for connecting to the **Pubs2** data source. The string specifically requests a connection to the **pubs2** database that resides on the **Cyber10** database server. The user ID is **mary**.

```
"DSN=Pubs2;SRVR=Cyber10;DB=pubs2;UID=mary;PWD=secret"
```

ODBC Conformance Levels

ODBC defines the API and the SQL conformance standards for drivers. *API conformance* refers to the functions that a driver supports. *SQL conformance* refers to the SQL statements, expressions, and data types that the driver supports. Each conformance standard consists of three levels. For more information on these standards, see "ODBC Conformance Levels" on page 1-11.

API Conformance Level

The Sybase driver supports the core and level-1 API functions listed in "API Conformance Levels" on page 1-11.

- The Sybase driver supports the following options for the **SQLGetConnectOption** and **SQLSetConnectOption** level-1 functions.

SQL_ACCESS_MODE	SQL_OPT_TRACEFILE
SQL_AUTOCOMMIT	SQL_TXN_ISOLATION
SQL_CURRENT_QUALIFIER	SQL_TRANSLATE_DLL
SQL_LOGIN_TIMEOUT	SQL_TRANSLATE_OPTION
SQL_OPT_TRACE	

  The driver supports SQL_ACCESS_MODE for only the **SQLGetConnectOption** function.

- The Sybase driver supports the following options for the **SQLSetStmtOption** and **SQLGetStmtOption** level-1 functions.

SQL_ASYNC_ENABLE	SQL_NOSCAN
SQL_BIND_TYPE	SQL_QUERY_TIMEOUT
SQL_CONCURRENCY	SQL_RETRIEVE_DATA
SQL_CURSOR_TYPE	SQL_ROWSET_SIZE
SQL_KEYSET_SIZE	SQL_ROW_NUMBER
SQL_MAX_LENGTH	SQL_SIMULATE_CURSOR
SQL_MAX_ROWS	

In addition, the Sybase driver supports the following level-2 functions.

SQLBrowseConnect	SQLNumParams
SQLColumnPrivileges	SQLParamOptions
SQLDataSources	SQLPrimaryKeys
SQLDrivers	SQLProcedureColumns
SQLExtendedFetch (forward)	SQLProcedures
SQLForeignKeys	SQLSetScrollOptions
SQLMoreResults	SQLTablePrivileges
SQLNativeSql	

SQL Conformance Level

The Sybase driver meets the minimum level of SQL conformance that is listed in "SQL Conformance Levels" on page 1-12. The driver also supports all core level statements, expressions, and data types with the exception of the ODBC ALTER TABLE syntax and a DOUBLE PRECISION data type. In addition to the ODBC extensions supported by the SYBASE SQL Server, the Sybase driver supports the following ODBC extensions to SQL:

- Date, time, and time-stamp data
- Outer joins
- The convert function
- The following numeric functions.

abs	ceiling	floor	power	sin
acos	cos	log	radians	sqrt
asin	cot	log10	rand	tan
atan	degrees	mod	round	
atan2	exp	pi	sign	

- The following date functions.

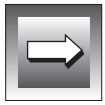
dayname	dayofyear	month	quarter	year
dayofmonth	hour	monthname	second	
dayofweek	minute	now	week	

- The following string functions.

ascii	lcase	repeat	soundex
char	left	right	substring
concat	length	rtrim	ucase
difference	ltrim	space	

- The following system functions.

database	ifnull	user



Important: *With the convert function, the data-type conversions must be supported by the underlying SYBASE database server.*

For more information on ODBC numeric, date, string, and system functions, see Appendix A, "Scalar Functions."

Mapping Data Types

The Sybase driver maps Sybase data types to their appropriate ODBC SQL data types. The following table lists the Sybase data type and its corresponding ODBC SQL data type.

Sybase Data Type	ODBC SQL Data Type
binary	SQL_BINARY
bit	SQL_BIT
char	SQL_CHAR
datetime	SQL_TIMESTAMP
decimal	SQL_DECIMAL
float	SQL_FLOAT
image	SQL_LONGVARBINARY
int	SQL_INTEGER
money	SQL_DECIMAL
numeric	SQL_NUMERIC

(1 of 2)

Sybase Data Type	ODBC SQL Data Type
real	SQL_REAL
smalldatetime	SQL_TIMESTAMP
smallint	SQL_SMALLLINT
smallmoney	SQL_DECIMAL
sysname	SQL_VARCHAR
text	SQL_LONGVARCHAR
timestamp	SQL_VARBINARY
tinyint	SQL_TINYINT
varbinary	SQL_VARBINARY
varchar	SQL_VARCHAR

(2 of 2)

Important: *The Sybase driver does not support the NCHAR, NVARCHAR, SENSITIVITY, and SENSITIVITY_BOUNDARY Sybase data types.*

Supported Isolation and Lock Levels

The Sybase driver supports isolation-levels 1 (read committed) and 3 (serializable). Level 1, read committed, is the default. The driver supports page-level locking.

For more information on isolation and locking levels, see the *Informix Guide to SQL: Tutorial* and your Sybase database server documentation.

Chapter

7

Sybase Driver for Windows

This chapter describes the Sybase driver for Windows environments, and explains how to complete the following tasks:

- Set up the Sybase driver
- Add or modify a Sybase data source
- Connect to a Sybase data source

In addition, the chapter includes information on the ODBC API and SQL conformance levels, data-type mappings, and isolation and locking levels that the Sybase driver supports.

Setting Up the Sybase Driver

The Sybase driver is installed during the driver-installation procedure. For Windows 3.1, the current driver is **IVINF509.DLL**. For Windows NT and Windows 95, the current driver is **IVSYB09.DLL**.

You must perform the following setup tasks before you can use the driver:

- Confirm that your system has the appropriate software installed.
- Confirm that your environment-variable settings reflect the appropriate directory paths.

System Requirements

To use the Sybase driver, you must have the following Sybase software installed on your system or network:

- SYBASE Open Client (Version 10.x)
- SYBASE System 10 or Sybase System 11 (database server)

To test that your Sybase network is working, use the WSYBPING (Windows 3.1) or SYBPING (Windows 95 or Windows NT) tool that is provided with the Sybase Net-Library. For information on these tools and other Sybase products, refer to your Sybase documentation set.

Setting Environment Variables

INFORMIX-Open provides double-byte character support through standard language support in Windows environments. (You must have a double-byte, localized Microsoft Windows environment.) Before you use the Informix driver, ensure that you configure the Windows environment for the language and cultural conventions that you require. For Windows 3.1 and Windows NT, in the **Program Manager**, choose **Main→Control Panel→International** to set these conventions. For Windows 95, set the appropriate language in the Control Panel, Regional Settings dialog box.

Check that the **SYBASE** environment variable is set to the full path of the directory where your System 10 client is installed.

```
Set SYBASE=C:\SQL10
```

For Windows 3.1 and Windows 95, this information is set in the **AUTOEXEC.BAT** file. For Windows NT, this information is set in the Registry.

For additional setting information, refer to your Sybase client-software documentation.

Adding and Modifying Data Sources

To add and configure data sources, use the ODBC Administrator. The ODBC Administrator then updates your **ODBC.INI** file in Windows 3.1 or your Registry in Windows 95 and Windows NT to reflect your data-source connection information.

Adding a Data Source

Windows3.1

In Windows 3.1, the **ODBC.INI** file is an initialization file used by the ODBC drivers and is located in the **WINDOWS** directory. This file contains information about each data source and its associated driver. Before you can connect to a data source, you must add the connection information for the data source to the **ODBC.INI** file. The ODBC Administrator adds the information to this file for you; never modify the **ODBC.INI** file directly. For complete information on the format and contents of this file, refer to "Understanding the odbc.ini File" on page 1-5. ♦

Windows NT

Windows 95

In Windows NT and Windows 95, **ODBC.INI** is a subkey of the HKEY_CURRENT_USER key within the Registry. When you access the Registry using this subkey, the ODBC structure is the same as the structure described in the "odbc.ini File Format" on page 1-6. Before you can connect to a data source, you must use the ODBC Administrator to add the connection information for the data source to the Registry. ♦

Required Data-Source Configuration Information

When you add a data source, you must provide two pieces of information.

Attribute	Value
Data Source Name	The name of the data source you want to access
Server Name	The name of the database server to which you want to connect by default If you do not supply a database server name, the driver uses the name of the database server specified by the **DSQUERY** environment variable.

All other connection information is optional.

Optional Data-Source Configuration Information

When you add a data source, you can define additional connection defaults. You can specify two types of connection options in your data-source setup:

- Options that define names
- Options that define performance settings

Options That Define Names

The following table lists the names that you can set as default connection options for a data source. Some of the attributes have default values.

Attribute	Value
Description	A long description that identifies the data source
Server List	The list of database servers that will be displayed in the logon dialog box If more than one name is specified, the names must be separated by commas.
Database Name	The name of the database to which you want to connect by default If you do not specify a value, the default is the database defined by the system administrator for each user.
Database List	The list of databases that appears in the logon dialog box If more than one name is specified, the names must be separated by commas.
Default Logon ID	Your user name as specified on the Sybase database server
Interfaces File	The pathname of the interfaces file The default is the normal Sybase interfaces file.

(1 of 2)

Attribute	Value
Password Encryption	A value that determines whether password encryption can be performed from the Open Client Library to the database server: ■ 1 = Enable password encryption. ■ 0 = Do not enable password encryption. This is the default setting.
Charset	The name of a character set corresponding to a subdirectory in $SYBASE/charsets The default is the setting on the database server.
Workstation ID	The workstation ID used by the client
Language	The national language corresponding to a subdirectory in $SYBASE/locales The default is English.
Application Name	The name used by the database server to identify your application
Yield Proc (For Windows 3.1 only)	A value of 0, 1, or 3 that determines whether you can work in other Windows applications when the Sybase driver is busy: ■ 0 = Peek and dispatch. This setting causes the driver to check the Windows message queue and send any messages to the appropriate Windows application. ■ 1 = No yielding. This setting does not let you work in other applications. This is the default setting. ■ 3 = Dispatch with the Windows Yield function. This setting turns control over to the Windows kernel. The Windows kernel checks the message queue and sends any messages to the appropriate application window. The recommended value is 1. ♦

(2 of 2)

Options That Define Performance Settings

The following table lists the types of operation behavior that you can set as default connection options for a data source. The table lists the default values.

Attribute	Value
Prepare Method	A value that determines whether stored procedures are created on the database server for every call to **SQLPrepare**: ■ 0 = Create stored procedures for every call to **SQLPrepare**. This setting can result in bad performance. ■ 1 = Create stored procedures only if the statement contains parameters. Otherwise, cache the statement and execute directly at **SQLExecute** time. This is the default setting. ■ 2= Do not create stored procedures.
Fetch Array Size	The number of rows the driver retrieves when fetching from the database server This is not the number of rows given to the user. The default is 10 rows.
Select Method	A value that determines whether database cursors are used for SELECT statements: ■ 0 = Use database cursors. This is the default setting. ■ 1 = Execute SELECT statements directly without using database cursors. This setting limits the data source to one active statement and one active connection.

(1 of 2)

Attribute	Value

Packet Size	A value of `-1`, `0`, or *x* that determines the number of bytes per network packet transferred from the database server to the client:
	■ `0` = Use the default packet size as specified in the Sybase database server configuration. This is the default setting.
	■ `1` = Compute the maximum allowable packet size on the first connect to the data source and save the value in **ODBC.INI**.
	■ *x* = Set the packet size to *x* `* 512` bytes where *x* is an integer from `1` to `10`. (For example, if the value were `6`, the packet size would be set to `6 * 512 or 3072` bytes.)
	To take advantage of this connection attribute, configure the Sybase database server for a maximum-network packet size greater than or equal to the value you specified for PacketSize. For example,
	`sp_configure "maximum network packet size", 5120`
	`reconfigure`
	`Restart System 10 Server`
	The ODBC specification identifies a connect option, SQL_PACKET_SIZE, that offers this same functionality. To avoid conflicts with applications that might set both this connection string attribute and the ODBC connect option, the two have been defined as mutually exclusive. If PacketSize is specified, you will receive a message "Driver Not Capable" if you attempt to call SQL_PACKET_SIZE. If you do not set PacketSize, then application calls to SQL_PACKET_SIZE are accepted by the driver.

Connection Cache	A value that determines the number of connections that the connection cache can hold The default setting is `1`. To set this option, you must set the Select Method option to `1` (Direct). Increasing the Connection Cache might increase the performance of some applications, but requires additional database resources.

(2 of 2)

Warning: *Windows 3.1 users should never modify the* ***ODBC.INI*** *file directly. To modify the contents of this file, use the ODBC administrator. If you modify the* ***ODBC.INI*** *file directly, data-source configuration errors might result.*

To add a data source

1. Invoke the ODBC Administrator.

   The Data Sources dialog box appears, as Figure 7-1 illustrates.

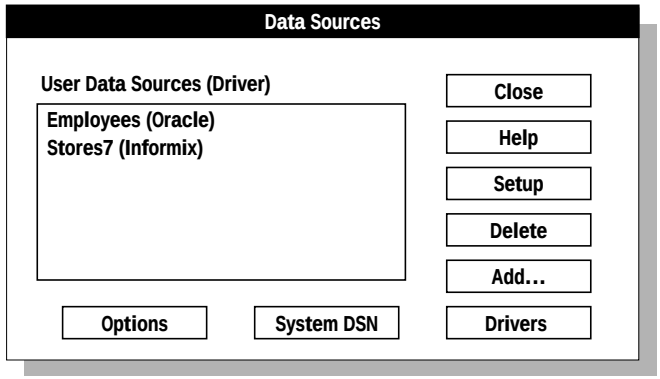


Figure 7-1
The Data Sources Dialog Box

2. Click **Add**.

   The Add Data Source dialog box appears, as Figure 7-2 illustrates.

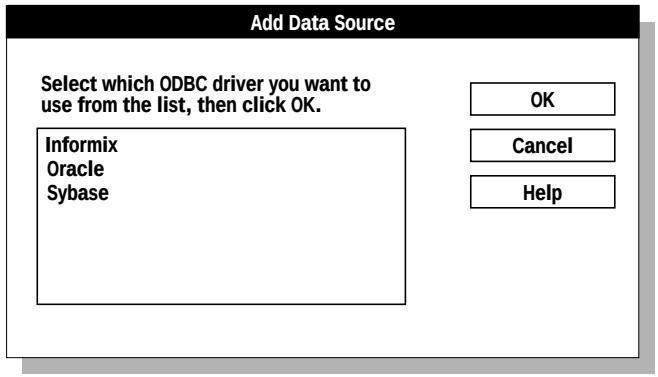


Figure 7-2
The Add Data Source Dialog Box

3. Select the Sybase driver from the Installed ODBC Drivers list.
4. Click **OK**.

   The ODBC System 10 Setup dialog box appears, as Figure 7-3 illustrates.

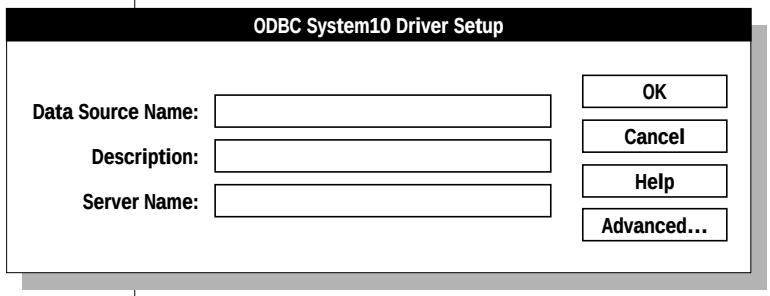


Figure 7-3
The ODBC System 10 Driver Setup Dialog Box

5. In the **Data Source Name** text box, enter the name of the data source you want to access.

   You define the data-source name; that is, it can be any name that you choose.
6. In the **Description** text box, type a long description of your data source. This section is optional.
7. In the **Server Name** text box, enter the name of the database server that contains the tables to which you want to connect by default. If you do not supply a name, the default is the database server name specified by the **DSQUERY** environment variable.

   You now have entered enough information to be able to connect to the data source:

   - To add the data source, click **OK.**
     The Data Sources dialog box appears, as Figure 7-1 on page 7-10 illustrates. You can then click **Add** to add another data source or click **Close** to exit the Data Sources dialog box.
   - To add optional connection information about the data source, click **Advanced**.
     The ODBC System 10 Advanced Driver Setup dialog box appears, as Figure 7-4 illustrates.

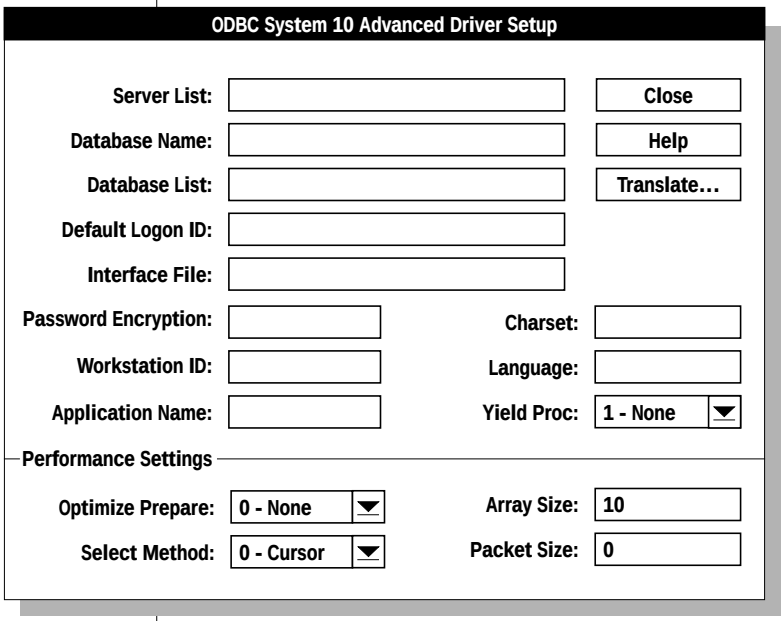


Figure 7-4
The ODBC System 10 Advanced Driver Setup Dialog Box

8. Enter the information you want to specify about this data source. You can also choose options from the drop-down lists. For a description of the advanced connection options that are available, see "Optional Data-Source Configuration Information" on page 7-6.

9. To specify a translator, click **Translate**.

   The Translator dialog box appears, as Figure 7-5 illustrates.

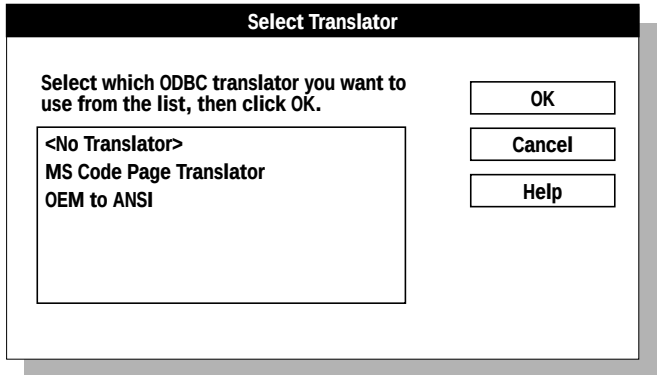


Figure 7-5
The Translator Dialog Box

10. Select the translator you want from the list.
11. Click **OK** to return to the ODBC System 10 Advanced Driver Setup dialog box.
12. Click **Close** to return to the ODBC System 10 Driver Setup dialog box.
13. To accept your specifications for this data source, click **OK**.

    When you click **OK** in the ODBC System 10 Driver Setup dialog box, the Data Sources dialog box appears, as shown in Figure 7-1 on page 7-10.
14. To add another data source, click **Add**. To exit the Data Sources dialog box, click **Close**.

After you click **OK** in the ODBC System 10 Driver Setup dialog box, the ODBC Administrator updates your **ODBC.INI** file (or Registry). The information that you entered in any of the setup dialog boxes becomes the new default data-source connection information for this data source.

Modifying a Data Source

Use the ODBC Administrator to make all modifications to your data source. To make basic modifications to the defaults for a data source, see "Using Dialog Boxes to Connect to a Data Source" on page 16. To make more detailed modifications to a data source, complete the following steps.

To modify a data source

1. Invoke the ODBC Administrator.

   The Data Sources dialog box appears, as Figure 7-6 illustrates.

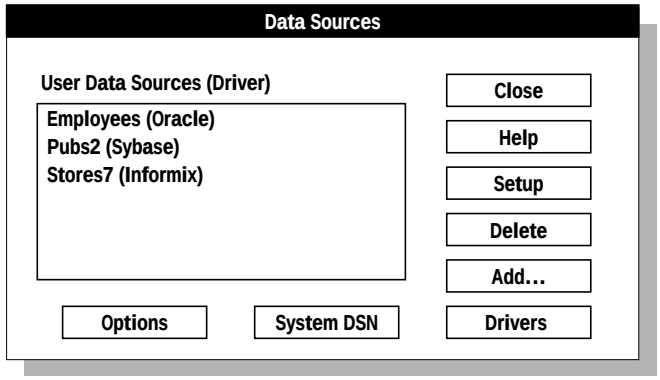


Figure 7-6
The Data Sources Dialog Box

2. In the Data Sources dialog box, select the Sybase data source that you want to modify and then click **Setup**.

   The ODBC System 10 Driver Setup dialog box appears, as page 7-15 illustrates. The values that appear are the default entries specified for this data-source connection.

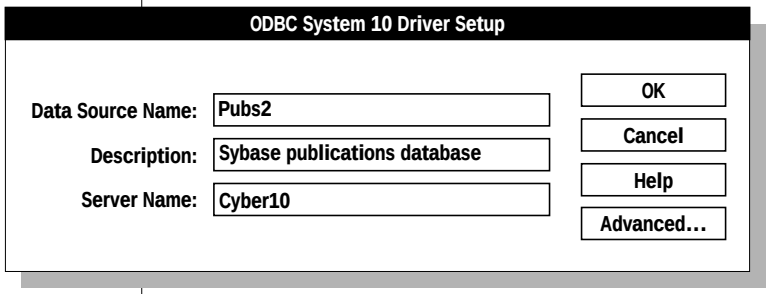


Figure 7-7
A Completed ODBC System 10 Driver Setup Dialog Box

3. Modify the applicable data-source text boxes in this dialog box and the ODBC System 10 Advanced Driver Setup dialog box if necessary.

   For more information regarding available options, see "Adding a Data Source" on page 7-5.

4. When you are finished, click **OK** in the ODBC System 10 Driver Setup dialog box.

   The ODBC Administrator updates the **ODBC.INI**.

   When you connect to this data source using either a dialog box or connection string, the values that you entered appear as the new default entries for the data-source connection.

Connecting to a Data Source

An Informix ODBC-compliant application can transfer connection information in a number of ways. For example, the application might have the driver always prompt the user for connection information, or the application might expect a connection string that specifies the data-source connection. How you connect to a data source depends on the connection method your application uses.

Using Dialog Boxes to Connect to a Data Source

One common way of connecting to a data source is through the Data Source dialog box, as Figure 7-8 illustrates. If your application is set up to use a dialog box, the Data Source dialog box is displayed and prompts you for the appropriate data-source connection information.

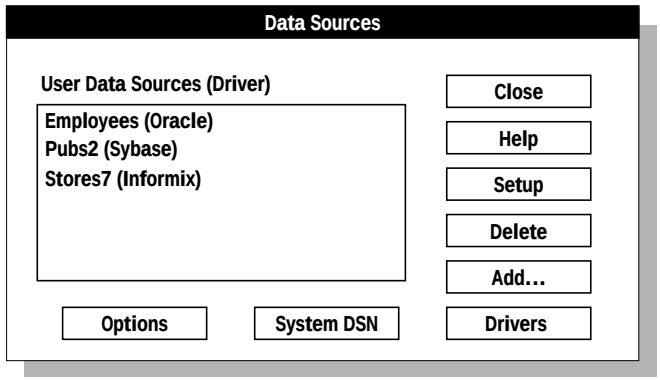


Figure 7-8
The Data Sources Dialog Box

To connect to a data source

1. In the Data Source dialog box, select a Sybase data source.
2. Click **OK**.

   The Logon to System 10 dialog box appears, as Figure 7-9 illustrates. The connection information that appears is the default information for the data source that is listed in your **ODBC.INI**.

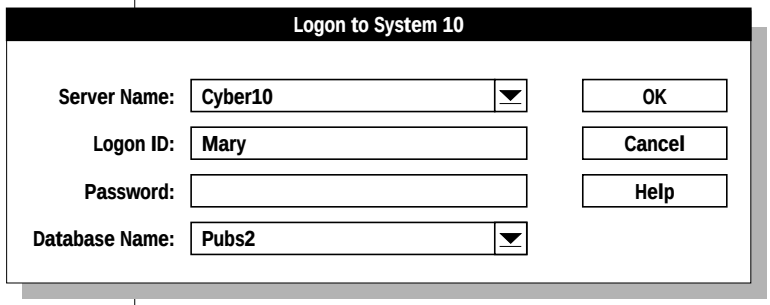


Figure 7-9
The Logon to System 10 Dialog Box

3. To accept the default values and connect to the data source, click **OK**.
   or
   To modify the default values for this data source, continue to step 4.
4. In the **Server Name** text box, type the case-sensitive name of the Sybase database server that contains the tables that you want to access.

   You can also click the down arrow to choose a name from a drop-down list.
5. In the **Logon ID** text box, type your case-sensitive user name as specified on the database server.
6. In the **Password** text box, type your case-sensitive password for the Sybase database server to which you want to connect.
7. In the **Database Name** text box, type the case-sensitive name of the database that contains the tables that you want to access.

   You can also click the down arrow to choose a name from a drop-down list.
8. Click **OK** to connect to the data source to update these values in **ODBC.INI**.

   Once the connection information is verified, your application can use the Sybase driver to access the information that the data source contains.

Using a Connection-String to Connect to a Data Source

Some applications require that you connect to a data source by typing a connection string. A connection string consists of a number of *attributes* that specify how a driver connects to a data source. An attribute identifies a specific piece of information that the driver must know before it can make the appropriate data-source connection.

A connection string has the following format:

```
"DSN=data_source_name[;attribute=value[;attribute=value]...]"
```

You must specify the *data_source_name*; however, all other attributes are optional. If you do not specify an attribute, it defaults to the one that is specified in the Data Source Specification section (for the data source specified in the connection string) of your **ODBC.INI** file in Windows 3.1 or your Registry in Windows NT and Windows 95.

The following table lists the long and short names of the attributes that you can include in the Sybase driver connection string. With the exception of the data-source name (which must be specified by `DSN`), you can use either the long or short names in the connection string. For a description of these attributes, see "Optional Data-Source Configuration Information" on page 7-6.

Attribute Long Name	Attribute Short Name
DataSourceName	DSN
ServerName	SRVR
LogonID	UID
Password	PWD
Database	DB
Language	LANG
Charset	CS
WorkstationID	WKID
ApplicationName	APP

(1 of 2)

Attribute Long Name	Attribute Short Name
InterfacesFile	IFILE
OptimizePrepare	OP
SelectMethod	SM
ArraySize	AS
YieldProc (Windows 3.1)	YLD
PasswordEncryption	PE
PacketSize	PS
CursorCacheSize	CCS

(2 of 2)

The following example shows a valid connection string for connecting to the **Pubs2** data source. The string specifically requests a connection to the **pubs2** database that resides on the **Cyber10** database server. The user ID is **mary**.

```
"DSN=Pubs2;SRVR=Cyber10;DB=pubs2;UID=mary;PWD=secret"
```

ODBC Conformance Levels

ODBC defines the API and the SQL conformance standards for drivers. *API conformance* refers to the functions that a driver supports. *SQL conformance* refers to the SQL statements, expressions, and data types that the driver supports. Each conformance standard consists of three levels. For more information on these standards, see "ODBC Conformance Levels" on page 1-11.

API Conformance Level

The Sybase driver supports the core and level-1 API functions listed in "API Conformance Levels" on page 1-11.

- The Sybase driver supports the following options for the **SQLGetConnectOption** and **SQLSetConnectOption** level-1 functions.

SQL_ACCESS_MODE	SQL_OPT_TRACEFILE
SQL_AUTOCOMMIT	SQL_PACKET_SIZE
SQL_CURRENT_QUALIFIER	SQL_QUIET_MODE
SQL_LOGIN_TIMEOUT	SQL_TRANSLATE_DLL
SQL_ODBC_CURSORS	SQL_TRANSLATE_OPTION
SQL_OPT_TRACE	SQL_TXN_ISOLATION

  The driver supports SQL_ACCESS_MODE for only the **SQLGetConnectOption** function.

- The Sybase driver supports the following options for the **SQLSetStmtOption** and **SQLGetStmtOption** level-1 functions.

SQL_ASYNC_ENABLE	SQL_NOSCAN
SQL_BIND_TYPE	SQL_QUERY_TIMEOUT
SQL_CONCURRENCY	SQL_RETRIEVE_DATA
SQL_CURSOR_TYPE	SQL_ROWSET_SIZE
SQL_KEYSET_SIZE	SQL_ROW_NUMBER
SQL_MAX_LENGTH	SQL_SIMULATE_CURSOR
SQL_MAX_ROWS	

In addition, the Sybase driver supports the following level-2 functions.

SQLBrowseConnect	SQLNumParams
SQLColumnPrivileges	SQLParamOptions
SQLDataSources	SQLPrimaryKeys
SQLDrivers	SQLProcedureColumns
SQLExtendedFetch (forward)	SQLProcedures
SQLForeignKeys	SQLSetScrollOptions
SQLMoreResults	SQLTablePrivileges
SQLNativeSql	

SQL Conformance Level

The Sybase driver meets the minimum level of SQL conformance that is listed in "SQL Conformance Levels" on page 1-12. The driver also supports all core level statements, expressions, and data types with the exception of the ODBC ALTER TABLE syntax and a DOUBLE PRECISION data type. In addition to the ODBC extensions supported by the SYBASE SQL Server, the Sybase driver supports the following ODBC extensions to SQL:

- Date, time, and time-stamp data
- Outer joins
- The convert function
- The following numeric functions.

abs	ceiling	floor	power	sin
acos	cos	log	radians	sqrt
asin	cot	log10	rand	tan
atan	degrees	mod	round	
atan2	exp	pi	sign	

- The following date functions.

dayname	dayofyear	month	quarter	year
dayofmonth	hour	monthname	second	
dayofweek	minute	now	week	

- The following string functions.

ascii	lcase	repeat	soundex
char	left	right	substring
concat	length	rtrim	ucase
difference	ltrim	space	

- The following system functions.

database	ifnull	user

Important: *With the convert function, the data-type conversions must be supported by the underlying SYBASE database server.*

For more information on ODBC numeric, date, string, and system functions, see Appendix A, "Scalar Functions."

Mapping Data Types

The Sybase driver maps Sybase data types to their appropriate ODBC SQL data types. The following table lists the Sybase data type and its corresponding ODBC SQL data type.

Sybase Data Type	ODBC SQL Data Type
binary	SQL_BINARY
bit	SQL_BIT
char	SQL_CHAR
datetime	SQL_TIMESTAMP
decimal	SQL_DECIMAL
float	SQL_FLOAT
image	SQL_LONGVARBINARY
int	SQL_INTEGER
money	SQL_DECIMAL
numeric	SQL_NUMERIC
real	SQL_REAL
smalldatetime	SQL_TIMESTAMP
smallint	SQL_SMALLLINT

(1 of 2)

Sybase Data Type	ODBC SQL Data Type
smallmoney	SQL_DECIMAL
sysname	SQL_VARCHAR
text	SQL_LONGVARCHAR
timestamp	SQL_VARBINARY
tinyint	SQL_TINYINT
varbinary	SQL_VARBINARY
varchar	SQL_VARCHAR

(2 of 2)

Important: *The Sybase driver does not support the NCHAR, NVARCHAR, SENSITIVITY, and SENSITIVITY_BOUNDARY Sybase data types.*

Supported Isolation and Lock Levels

The Sybase driver supports isolation-levels 1 (read committed) and 3 (serializable). Level 1, read committed, is the default. The driver supports page-level locking.

For more information on isolation and locking levels, see the *Informix Guide to SQL: Tutorial* and your Sybase database server documentation.

Appendix

A

Scalar Functions

ODBC specifies five types of scalar functions: string, numeric, time and data, system, and data-type conversion functions. The following sections list functions by function type. Descriptions include the associated syntax.

- String Functions
- Numeric Functions
- Time and Date Functions
- System Functions
- Explicit Data-Type Conversion

String Functions

Figure A-1 on page A-2 lists string-manipulation functions. Character-string literals used as arguments to scalar functions must be bounded by single quotes.

Arguments denoted as *string_exp* can be the name of a column, a string literal, or the result of another scalar function, where the underlying data type can be represented as SQL_CHAR, SQL_VARCHAR, or SQL_LONGVARCHAR.

Arguments denoted as *start*, *length*, *code*, or *count* can be a numeric literal or the result of another scalar function, where the underlying data type can be represented as SQL_TINYINT, SQL_SMALLINT, or SQL_INTEGER.

The string functions listed here are 1-based; that is, the first character in the string is the character 1.

Figure A-1
String Functions

Function	Description
ASCII(*string_exp*)	Returns the ASCII code value of the left-most character of *string_exp* as an integer.
CHAR(*code*)	Returns the character that has the ASCII code value specified by *code*. The value of *code* should be between 0 and 255; otherwise, the return value is data-source dependent.
CONCAT(*string_exp1*, *string_exp2*)	Returns a character string that is the result of concatenating *string_exp2* to *string_exp1*. The resulting string is DBMS dependent. For example, if the column represented by *string_exp1* contained a null value, DB2 would return null, but SQL Server would return the non-null string.
DIFFERENCE (*string_exp1*, *string_exp2*)	Returns an integer value that indicates the difference between the values returned by the SOUNDEX function for *string_exp1* and *string_exp2*.
INSERT(*string_exp1*, *start*,*length*, *string_exp2*)	Returns a character string where *length* characters were deleted from *string_exp1* beginning at *start* and where *string_exp2* was inserted into *string_exp*, beginning at *start*.
LCASE(*string_exp*)	Converts all uppercase characters in *string_exp* to lowercase characters.
LEFT(*string_exp*, *count*)	Returns the left-most *count* of characters of *string_exp*.
LENGTH(*string_exp*)	Returns the number of characters in *string_exp*, excluding trailing blanks and the string-termination character.
LOCATE(*string_exp1*, *string_exp2*[, *start*])	Returns the starting position of the first occurrence of *string_exp1* within *string_exp2*. The search for the first occurrence of *string_exp1* begins with the first character position in *string_exp2* unless the optional argument, *start*, is specified. If *start* is specified, the search begins with the character position indicated by the value of *start*. The first character position in *string_exp2* is indicated by the value 1. If *string_exp1* is not found within *string_exp2*, the value 0 is returned.

(1 of 2)

Function	Description
LTRIM(*string_exp*)	Returns the characters of *string_exp*, with leading blanks removed.
REPEAT (*string_exp*,*count*)	Returns a character string composed of *string_exp* repeated *count* times.
REPLACE(*string_exp1*, *string_exp2*, *string_exp3*)	Replaces all occurrences of *string_exp2* in *string_exp1* with *string_exp3*.
RIGHT(*string_exp*, *count*)	Returns the right-most count of characters of *string_exp*.
RTRIM(*string_exp*)	Returns the characters of *string_exp* with trailing blanks removed.
SOUNDEX(*string_exp*)	Returns a data-source dependent character string representing the sound of the words in *string_exp*. For example, SQL Server returns a four-digit SOUNDEX code; Oracle returns a phonetic representation of each word.
SPACE(*count*)	Returns a character string consisting of *count* spaces.
SUBSTRING (*string_exp*, *start*, *length*)	Returns a character string that is derived from *string_exp* beginning at the character position specified by *start* for *length* characters.
UCASE(*string_exp*)	Converts all lowercase characters in *string_exp* to uppercase characters.

(2 of 2)

Numeric Functions

Figure A-2 on page A-4 describes numeric functions that are included in the ODBC scalar function set.

Arguments denoted as *numeric_exp* can be the name of a column, the result of another scalar function, or a numeric literal, where the underlying data type could be represented as SQL_NUMERIC, SQL_DECIMAL, SQL_TINYINT, SQL_SMALLINT, SQL_INTEGER, SQL_BIGINT, SQL_FLOAT, SQL_REAL, or SQL_DOUBLE.

Arguments denoted as *float_exp* can be the name of a column, the result of another scalar function, or a numeric literal, where the underlying data type can be represented as SQL_FLOAT.

Arguments denoted as *integer_exp* can be the name of a column, the result of another scalar function, or a numeric literal, where the underlying data type can be represented as SQL_TINYINT, SQL_SMALLINT, SQL_INTEGER, or SQL_BIGINT.

Figure A-2
Numeric Functions

Function	Description
ABS(*numeric_exp*)	Returns the absolute value of *numeric_exp*.
ACOS(*float_exp*)	Returns the arccosine of *float_exp* as an angle, expressed in radians.
ASIN(*float_exp*)	Returns the arcsine of *float_exp* as an angle, expressed in radians.
ATAN(*float_exp*)	Returns the arctangent of *float_exp* as an angle, expressed in radians.
ATAN2(*float_exp1*, *float_exp2*)	Returns the arctangent of the *x* and *y* coordinates, specified by *float_exp1* and *float_exp2*, respectively, as an angle, expressed in radians.
CEILING (*numeric_exp*)	Returns the smallest integer greater than or equal to *numeric_exp*.
COS(*float_exp*)	Returns the cosine of *float_exp*, where *float_exp* is an angle expressed in radians.
COT(*float_exp*)	Returns the cotangent of *float_exp*, where *float_exp* is an angle expressed in radians.
DEGREES (*numeric_exp*)	Returns the number of degrees converted from *numeric_exp* radians.
EXP(*float_exp*)	Returns the exponential value of *float_exp*.
FLOOR(*numeric_exp*)	Returns largest integer less than or equal to *numeric_exp*.
LOG(*float_exp*)	Returns the natural logarithm of *float_exp*.

(1 of 2)

Function	Description
LOG10(*float_exp*)	Returns the base 10 logarithm of *float_exp*.
MOD(*integer_exp1*, *integer_exp2*)	Returns the remainder (modulus) of *integer_exp1* divided by *integer_exp2*.
PI( )	Returns the constant value of pi as a floating-point value.
POWER(*numeric_exp*, *integer_exp*)	Returns the value of *numeric_exp* to the power of *integer_exp*.
RADIANS (*numeric_exp*)	Returns the number of radians converted from *numeric_exp* degrees.
RAND([*integer_exp*])	Returns a random floating-point value using *integer_exp* as the optional seed value.
ROUND(*numeric_exp*, *integer_exp*)	Returns *numeric_exp* rounded to *integer_exp* places right of the decimal point. If *integer_exp* is negative, *numeric_exp* is rounded to |*integer_exp*| places to the left of the decimal point.
SIGN(*numeric_exp*)	Returns an indicator or the sign of *numeric_exp*. If *numeric_exp* is less than zero, `-1` is returned. If *numeric_exp* equals zero, `0` is returned. If *numeric_exp* is greater than zero, `1` is returned.
SIN(*float_exp*)	Returns the sine of *float_exp*, where *float_exp* is an angle expressed in radians.
SQRT(*float_exp*)	Returns the square root of *float_exp*.
TAN(*float_exp*)	Returns the tangent of *float_exp*, where *float_exp* is an angle expressed in radians.
TRUNCATE (*numeric_exp*, *integer_exp*)	Returns *numeric_exp* truncated to *integer_exp* places right of the decimal point. If *integer_exp* is negative, *numeric_exp* is truncated to |*integer_exp*| places to the left of the decimal point.

(2 of 2)

Time and Date Functions

Figure A-3 lists time and date functions that are included in the ODBC scalar function set. Date, time, and time-stamp literals must be bounded by single quotes.

Arguments denoted as *timestamp_exp* can be the name of a column, the result of another scalar function, or a time, date, or time-stamp literal, where the underlying data type could be represented as SQL_CHAR, SQL_VARCHAR, SQL_TIME, SQL_DATE, or SQL_TIMESTAMP.

Arguments denoted as *date_exp* can be the name of a column, the result of another scalar function, or a date or time-stamp literal, where the underlying data type could be represented as SQL_CHAR, SQL_VARCHAR, SQL_DATE, or SQL_TIMESTAMP.

Arguments denoted as *time_exp* can be the name of a column, the result of another scalar function, or a time or time-stamp literal, where the underlying data type could be represented as SQL_CHAR, SQL_VARCHAR, SQL_TIME, or SQL_TIMESTAMP.

Values returned are represented as ODBC data types.

Figure A-3
Time and date functions

Function	Description
CURDATE( )	Returns the current date as a date value.
CURTIME( )	Returns the current local time as a time value.
DAYNAME(*date_exp*)	Returns a character string containing the data-source-specific name of the day (for example, Sunday through Saturday, or Sun. through Sat. for a data source that uses English, or Sonntag through Samstag for a data source that uses German) for the day portion of *date_exp*.
DAYOFMONTH (*date_exp*)	Returns the day of the month in *date_exp* as an integer value in the range of 1 through 31.
DAYOFWEEK (*date_exp*)	Returns the day of the week in *date_exp* as an integer value in the range of 1 through 7, where 1 represents Sunday.

(1 of 4)

Function	Description
DAYOFYEAR (*date_exp*)	Returns the day of the year in *date_exp* as an integer value in the range of 1 through 366.
HOUR(*time_exp*)	Returns the hour in *time_exp* as an integer value in the range of 0 through 23.
MINUTE(*time_exp*)	Returns the minute in *time_exp* as an integer value in the range of 0 through 59.
MONTH(*date_exp*)	Returns the month in *date_exp* as an integer value in the range of 1 through 12.
MONTHNAME (*date_exp*)	Returns a character string containing the data source–specific name of the month (for example, January through December, or Jan. through Dec. for a data source that uses English, or Januar through Dezember for a data source that uses German) for the month portion of *date_exp*.
NOW( )	Returns current date and time as a time-stamp value.
QUARTER(*date_exp*)	Returns the quarter in *date_exp* as an integer value in the range of 1 through 4, where 1 represents January 1 through March 31.
SECOND(*time_exp*)	Returns the second in *time_exp* as an integer value in the range of 0 through 59.

(2 of 4)

Function	Description

TIMESTAMPADD (*interval*, *integer_exp*, *timestamp_exp*)	Returns the time stamp calculated by adding *integer_exp* intervals of type *interval* to *timestamp_exp*. Valid values of interval are the following keywords:
	SQL_TSI_FRAC_SECOND
	SQL_TSI_SECOND
	SQL_TSI_MINUTE
	SQL_TSI_HOUR
	SQL_TSI_DAY
	SQL_TSI_WEEK
	SQL_TSI_MONTH
	SQL_TSI_QUARTER
	SQL_TSI_YEAR
	where fractional seconds are expressed in billionths of a second. For example, the following SQL statement returns the name of each employee and their one-year anniversary dates:
	`SELECT NAME,`
	`{fn TIMESTAMPADD(SQL_TSI_YEAR,`
	`1, HIRE_DATE)} FROM EMPLOYEES`
	If *timestamp_exp* is a time value and interval specifies days, weeks, months, quarters, or years, the date portion of *timestamp_exp* is set to the current date before calculating the resulting time stamp.
	If *timestamp_exp* is a date value and *interval* specifies fractional seconds, seconds, minutes, or hours, the time portion of *timestamp_exp* is set to `0` before calculating the resulting time stamp.
	An application determines which intervals a data source supports by calling **SQLGetInfo** with the SQL_TIMEDATE_ADD_INTERVALS option.

(3 of 4)

Function	Description

TIMESTAMPDIFF (*interval*, *timestamp_exp1*, *timestamp_exp2*)	Returns the integer number of intervals of type *interval* by which *timestamp_exp2* is greater than *timestamp_exp1*. Valid values of interval are the following keywords:
	SQL_TSI_FRAC_SECOND
	SQL_TSI_SECOND
	SQL_TSI_MINUTE
	SQL_TSI_HOUR
	SQL_TSI_DAY
	SQL_TSI_WEEK
	SQL_TSI_MONTH
	SQL_TSI_QUARTER
	SQL_TSI_YEAR
	where fractional seconds are expressed in billionths of a second. For example, the following SQL statement returns the name of each employee and the number of years they have been employed.
	`SELECT NAME,`
	`fn TIMESTAMPDIFF(SQL_TSI_YEAR,`
	`{fn CURDATE()}, HIRE_DATE)}`
	`FROM EMPLOYEES`
	If either time-stamp expression is a time value and *interval* specifies days, weeks, months, quarters, or years, the date portion of that time stamp is set to the current date before calculating the difference between the time stamps.
	If either time stamp expression is a date value and *interval* specifies fractional seconds, seconds, minutes, or hours, the time portion of that time stamp is set to 0 before calculating the difference between the time stamps.
	An application determines which intervals a data source supports by calling **SQLGetInfo** with the SQL_TIMEDATE_DIFF_INTERVALS option.

WEEK(*date_exp*)	Returns the week of the year in *date_exp* as an integer value in the range of 1 through 53.
YEAR(*date_exp*)	Returns the year in *date_exp* as an integer value. The range is data-source dependent.

(4 of 4)

System Functions

Figure A-4 lists system functions that are included in the ODBC scalar function set.

Arguments denoted as *exp* can be the name of a column, the result of another scalar function, or a literal, where the underlying data type could be represented as SQL_NUMERIC, SQL_DECIMAL, SQL_TINYINT, SQL_SMALLINT, SQL_INTEGER, SQL_BIGINT, SQL_FLOAT, SQL_REAL, SQL_DOUBLE, SQL_DATE, SQL_TIME, or SQL_TIMESTAMP.

Arguments denoted as *value* can be a literal constant, where the underlying data type can be represented as SQL_NUMERIC, SQL_DECIMAL, SQL_TINYINT, SQL_SMALLINT, SQL_INTEGER, SQL_BIGINT, SQL_FLOAT, SQL_REAL, SQL_DOUBLE, SQL_DATE, SQL_TIME, or SQL_TIMESTAMP.

Values returned are represented as ODBC data types.

Figure A-4
System functions

Function	Description
DATABASE( )	Returns the name of the database corresponding to the connection handle (*hdbc*). (The name of the database is also available by calling **SQLGetConnectOption** with the SQL_CURRENT_QUALIFIER connection option.)
IFNULL(*exp*,*value*)	If *exp* is null, *value* is returned. If *exp* is not null, *exp* is returned. The possible data type(s) of *value* must be compatible with the data type of *exp*.
USER( )	Returns the authorization name of the user. (The authorization name of the user is also available by way of **SQLGetInfo** by specifying the information type: SQL_USER_NAME.)

Explicit Data-Type Conversion

Explicit data-type conversion is specified in terms of ODBC SQL data-type definitions.

The ODBC syntax for the explicit data-type conversion function does not restrict conversions. The validity of specific conversions of one data type to another data type are determined by each driver-specific implementation. As the driver translates the ODBC syntax into the native syntax, it rejects conversions that, although legal in the ODBC syntax, are not supported by the data source. The ODBC function **SQLGetInfo** provides a way to inquire about conversions supported by the data source.

The format of the **CONVERT** function is as follows:

```
CONVERT(value_exp, data_type)
```

The function returns the value specified by *value_exp* converted to the specified *data_type*, where *data_type* is one of the following keywords:

- SQL_BIGINT
- SQL_BINARY
- SQL_BIT
- SQL_CHAR
- SQL_DATE
- SQL_DECIMAL
- SQL_DOUBLE
- SQL_FLOAT
- SQL_INTEGER
- SQL_LONGVARBINARY
- SQL_LONGVARCHAR
- SQL_REAL
- SQL_SMALLINT
- SQL_TIME
- SQL_TIMESTAMP
- SQL_TINYINT

- SQL_VARBINARY
- SQL_VARCHAR

The ODBC syntax for the explicit data-type conversion function does not support specification of conversion format. If specification of explicit formats is supported by the underlying data source, a driver must specify a default value or implement format specification.

The argument *value_exp* can be a column name, the result of another scalar function, or a numeric or string literal. The following example converts the output of the CURDATE scalar function to a character string:

```
{ CONVERT( { CURDATE() }, SQL_CHAR) }
```

The following two examples illustrate the use of the CONVERT function. These examples assume the existence of a table called **employees**, with an **empno** column of type SQL_SMALLINT and an **empname** column of type SQL_CHAR.

If an application specifies the following query:

```
SELECT empno FROM employees WHERE
--(*vendor(Microsoft),product(ODBC) CONVERT(empno,SQL_CHAR)*)--
LIKE '1%'
```

or its equivalent in shorthand form:

```
SELECT empno FROM employees WHERE
    { CONVERT(empno,SQL_CHAR)} LIKE '1%'
```

A driver that supports an Oracle DBMS would translate the request to:

```
SELECT empno FROM employees WHERE to_char(empno) LIKE '1%'
```

A driver that supports a Sybase SQL Server DBMS would translate the request to:

```
SELECT empno FROM employees WHERE convert(char,empno) LIKE
'1%'
```

If an application specifies the following query:

```
SELECT
--(*vendor(Microsoft),product(ODBC) ABS(empno)*)--,
--(*vendor(Microsoft),product(ODBC) CONVERT(empname,SQL_SMALLINT)*)--
   FROM employees WHERE empno <> 0
```

or its equivalent in shorthand form:

```
SELECT {ABS(empno)}, {CONVERT(empname,SQL_SMALLINT)}
    FROM employees WHERE empno <> 0
```

A driver that supports an Oracle DBMS would translate the request to:

```
SELECT abs(empno), to_number(empname) FROM employees
    WHERE empno <> 0
```

A driver that supports a Sybase SQL Server DBMS would translate the request to:

```
SELECT abs(empno), convert(smallint, empname) FROM employees
    WHERE empno != 0
```

Index

A

C

D

E

F

G

H

I

P

R

S

T

U

W

Y