ODBC Connectivity Guide

Informix Red Brick Decision Server

Version 2.0
November 1999
Part No. 000-6426

Published by Informix® Press

Informix Corporation
4100 Bohannon Drive
Menlo Park, CA 94025-1032

© 1999 Informix Corporation. All rights reserved. The following are trademarks of Informix Corporation or its affiliates, one or more of which may be registered in the United States or other jurisdictions:

Answers OnLine™; C-ISAM®; Client SDK™; DataBlade®; Data Director™; Decision Frontier™; Dynamic Scalable Architecture™; Dynamic Server™; Dynamic Server™, Developer Edition™; Dynamic Server™ with Advanced Decision Support Option™; Dynamic Server™ with Extended Parallel Option™; Dynamic Server™ with MetaCube®; Dynamic Server™ with Universal Data Option™; Dynamic Server™ with Web Integration Option™; Dynamic Server™, Workgroup Edition™; Dynamic Virtual Machine™; Enterprise Decision Server™; Formation™; Formation Architect™; Formation Flow Engine™; Gold Mine Data Access®; IIF.2000™; i.Reach™; i.Sell™; Illustra®; Informix®; Informix® 4GL; Informix® InquireSM; Informix® Internet Foundation.2000™; InformixLink®; Informix® Red Brick® Decision Server™; Informix Session Proxy™; Informix® Vista™; InfoShelf™; Interforum™; I-Spy™; Mediazation™; MetaCube®; NewEra™; ON-Bar™; OnLine Dynamic Server™; OnLine/Secure Dynamic Server™; OpenCase®; Orca™; PaVER™; Red Brick® and Design; Red Brick® Data Mine™; Red Brick® Mine Builder™; Red Brick® Decisionscape™; Red Brick® Ready™; Red Brick Systems®; Regency Support®; Rely on Red BrickSM; RISQL®; Solution DesignSM; STARindex™; STARjoin™; SuperView®; TARGETindex™; TARGETjoin™; The Data Warehouse Company®; The one with the smartest data wins.™; The world is being digitized. We're indexing it.SM; Universal Data Warehouse Blueprint™; Universal Database Components™; Universal Web Connect™; ViewPoint®; Visionary™; Web Integration Suite™. The Informix logo is registered with the United States Patent and Trademark Office. The DataBlade logo is registered with the United States Patent and Trademark Office.

Documentation Team: Kathy Eckardt, Jennifer Leland, Ann McKillop

GOVERNMENT LICENSE RIGHTS

Software and documentation acquired by or for the US Government are provided with rights as follows: (1) if for civilian agency use, with rights as restricted by vendor's standard license, as prescribed in FAR 12.212; (2) if for Dept. of Defense use, with rights as restricted by vendor's standard license, unless superseded by a negotiated vendor license, as prescribed in DFARS 227.7202. Any whole or partial reproduction of software or documentation marked with this legend must reproduce this legend.

Table of Contents

Introduction

In This Introduction

This Introduction provides an overview of the information in this document and describes the conventions it uses.

About This Guide

This guide provides Open Database Connectivity (ODBC) application developers with information about Informix Red Brick Decision Server that they can use to create compatible end-user database applications. This document is intended for use with other Red Brick Decision Server documentation and with the *Microsoft* ODBC *2.0 Programmer's Reference and* SDK *Guide*.

Types of Users

This guide is written for the following users:

- Database users
- Database server administrators
- Database-application programmers
- Database designers
- Database developers

This guide assumes that you have the following background:

- A working knowledge of your computer, your operating system, and the utilities that your operating system provides
- Some experience working with relational databases or exposure to database concepts
- Some experience with computer programming
- Some experience with database server administration, operating-system administration, or network administration

Software Dependencies

This guide assumes that you are using Informix Red Brick Decision Server, Version 6.0, as your database server.

Red Brick Decision Server includes the Aroma database, which contains sales data about a fictitious coffee and tea company. The database tracks daily retail sales in stores owned by the Aroma Coffee and Tea Company. The dimensional model for this database consists of a fact table and its dimensions.

For information about how to create and populate the demonstration database, see the *Administrator's Guide*. For a description of the database and its contents, see the *SQL Self-Study Guide*.

The scripts that you use to install the demonstration database reside in the *redbrick_dir/sample_input* directory, where *redbrick_dir* is the Red Brick Decision Server directory on your system.

New Features

The following section describes new database server features relevant to this document. For a comprehensive list of new features, see the release notes.

- Informix Red Brick JDBC Driver, which allows Java programs to access database management systems
- Support for the VARCHAR (variable-length character) data type
- Connectivity enhancements
- Level 2 support for the following functions:
  - **SQLPrimaryKeys**
  - **SQLForeignKeys**
  - **SQLTablePrivileges**
  - **SQLColumnPrivileges**
  - **SQLDescribeParam** (Red Brick Decision Server, Version 6.0 only)

Documentation Conventions

Informix Red Brick documentation uses the following notation and syntax conventions:

- Computer input and output, including commands, code, and examples, appear in `Courier`.
- Information that you enter or that is being emphasized in an example appears in **`Courier bold`** to help you distinguish it from other text.
- Filenames appear in bold. System-level commands and variables appear in *italic* or *`Courier italic`*, depending on the context.
- Document titles always appear in *Palatino italic*.
- Names of database tables and columns are capitalized (Sales table, Dollars column). Names of system tables and columns are in all uppercase (RBW_INDEXES table, TNAME column).

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 three types of information, as the following table describes. This information always appears in italics.

Icon	Label	Description
	***Warning:***	Identifies paragraphs that contain vital instructions, cautions, or critical information
	***Important:***	Identifies paragraphs that contain significant information about the feature or operation that is being described
	***Tip:***	Identifies paragraphs that offer additional details or shortcuts for the functionality that is being described

Platform Icons

Feature, product, and platform icons identify paragraphs that contain platform-specific information.

Icon	Description
	Identifies information that is specific to UNIX platforms
	Identifies information that is specific to Windows NT, Windows 95, and Windows 98 environments

These icons can apply to an entire section or to one or more paragraphs within a section. If an icon appears next to a section heading, the information that applies to the indicated feature, product, or platform ends at the next heading at the same or higher level. A ♦ symbol indicates the end of feature-, product-, or platform-specific information that appears within one or more paragraphs within a section.

Customer Support

Please review the following information before contacting Informix Customer Support.

If you have technical questions about Informix Red Brick Decision Server but cannot find the answer in the appropriate document, contact Informix Customer Support as follows:

Telephone	1-800-274-8184 or 1-913-492-2086 (7 A.M. to 7 P.M. CST, Monday through Friday)
Internet access	http://www.informix.com/techinfo

For nontechnical questions about Red Brick Decision Server, contact Informix Customer Support as follows:

Telephone	1-800-274-8184 (7 A.M. to 7 P.M. CST, Monday through Friday)
Internet access	http://www.informix.com/services

New Cases

To log a new case, you must provide the following information:

- Red Brick Decision Server version
- Platform and operating-system version
- Error messages returned by Red Brick Decision Server or the operating system
- Concise description of the problem, including any commands or operations performed before you received the error message
- List of Red Brick Decision Server or operating-system configuration changes made before you received the error message

For problems concerning client-server connectivity, you must provide the following additional information:

- Name and version of the client tool that you are using
- Version of Informix Red Brick ODBC Driver or Informix Red Brick JDBC Driver that you are using, if applicable
- Name and version of client network or TCP/IP stack in use
- Error messages returned by the client application
- Server and client locale specifications

Existing Cases

The support engineer who logs your case or first contacts you will always give you a case number. This number is used to keep track of all the activities performed during the resolution of each problem. To inquire about the status of an existing case, you must provide your case number.

Troubleshooting Tips

You can often reduce the time it takes to close your case by providing the smallest possible reproducible example of your problem. The more you can isolate the cause of the problem, the more quickly the support engineer can help you resolve it, as follows:

- For SQL query problems, try to remove columns or functions or to restate WHERE, ORDER BY, or GROUP BY clauses until you can isolate the part of the statement causing the problem.
- For Table Management Utility load problems, verify the data type mapping between the source file and the target table to ensure compatibility. Try to load a small test set of data to determine whether the problem concerns volume or data format.
- For connectivity problems, issue the *ping* command from the client to the host to verify that the network is up and running. If possible, try another client tool to see if the same problem arises.

Related Documentation

The standard documentation set for Red Brick Decision Server includes the following documents.

Document	Description
Administrator's Guide	Describes warehouse architecture, supported schemas, and other concepts relevant to databases. Procedural information for designing and implementing a database, maintaining a database, and tuning a database for performance. Includes a description of the system tables and the configuration file.
Installation and Configuration Guide	Provides installation and configuration information, as well as platform-specific material, about Red Brick Decision Server and related products. Customized for either UNIX or Windows NT.
Messages and Codes Reference Guide	Contains a complete listing of all informational, warning, and error messages generated by Informix Red Brick Decision Server products, including probable causes and recommended responses. Also includes event log messages that are written to the log files.
The release notes	Contains information pertinent to the current release that was unavailable when the documents were printed.
RISQL Entry Tool and RISQL Reporter User's Guide	Is a complete guide to the RISQL Entry Tool, a command-line tool used to enter SQL statements, and the RISQL Reporter, an enhanced version of the RISQL Entry Tool with report-formatting capabilities.
SQL Reference Guide	Is a complete language reference for the Informix Red Brick SQL implementation and RISQL extensions for warehouse databases.
SQL Self-Study Guide	Provides an example-based review of SQL and introduction to the RISQL extensions, the macro facility, and Aroma, the sample database.
Table Management Utility Reference Guide	Describes the Table Management Utility, including all activities related to loading and maintaining data. Also includes information about data replication and the rb_cm copy management utility.

In addition to the standard documentation set, the following documents are included for specific sites.

Document	Description
Client Connector Pack Installation Guide	Includes procedures for installing and configuring the Informix Red Brick ODBC Driver, the RISQL Entry Tool, and the RISQL Reporter on client systems. Included for sites that purchase the Client Connector Pack.
SQL-BackTrack User's Guide	Is a complete guide to SQL-BackTrack, a command-line interface for backing up and recovering warehouse databases. Includes procedures for defining backup configuration files, performing online and checkpoint backups, and recovering the database to a consistent state.
Informix Vista User's Guide	Describes the Informix Vista aggregate navigation and advisory system. Illustrates how Vista improves the performance of queries by automatically rewriting queries using aggregates, describes how the Advisor recommends the best set of aggregates based on data collected daily, and shows how the system operates in a versioned environment.
JDBC Connectivity Guide	Includes information about Informix Red Brick JDBC Driver and the JDBC API, which allow Java programs to access database management systems.
ODBC Connectivity Guide	Includes information about ODBC conformance levels and instructions for using the Informix Red Brick ODBClib SDK to compile and link an ODBC application.

Additional references you might find helpful include:

- An introductory-level book on SQL
- An introductory-level book on relational databases
- Documentation for your hardware platform and operating system

Additional Documentation

For additional information, you might want to refer to the following documents, which are available as online and printed manuals.

Online Manuals

An Answers OnLine CD that contains Informix manuals in electronic format is provided with your Informix products. You can install the documentation or access it directly from the CD. For information about how to install, read, and print online manuals, see the installation insert that accompanies Answers OnLine.

Printed Manuals

To order printed manuals, call 1-800-331-1763 or send email to moreinfo@informix.com. Please provide the following information when you place your order:

- The documentation that you need
- The quantity that you need
- Your name, address, and phone number

Informix Welcomes Your Comments

Let us know what you like or dislike about our manuals. To help us with future versions of our manuals, we want to know about any corrections or clarifications that you would find useful. 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

The **doc** alias is reserved exclusively for reporting errors and omissions in our documentation.

We appreciate your suggestions.

Chapter

1

Red Brick ODBC Driver Description and Implementation

In This Chapter

This chapter describes Informix Red Brick ODBC Driver for computers that run on UNIX and Windows platforms. Red Brick ODBC Driver is the Informix implementation of the Microsoft Open Database Connectivity (ODBC) interface, Version 2.0.

The driver allows a wide variety of Red Brick ODBC Driver-compliant database applications to work with Red Brick Decision Server. In addition, the driver acts as a translator between the Red Brick ODBC Driver interface that the client query applications use and the Red Brick Decision Server Red Brick ODBC Driver native application program interfaces (APIs).

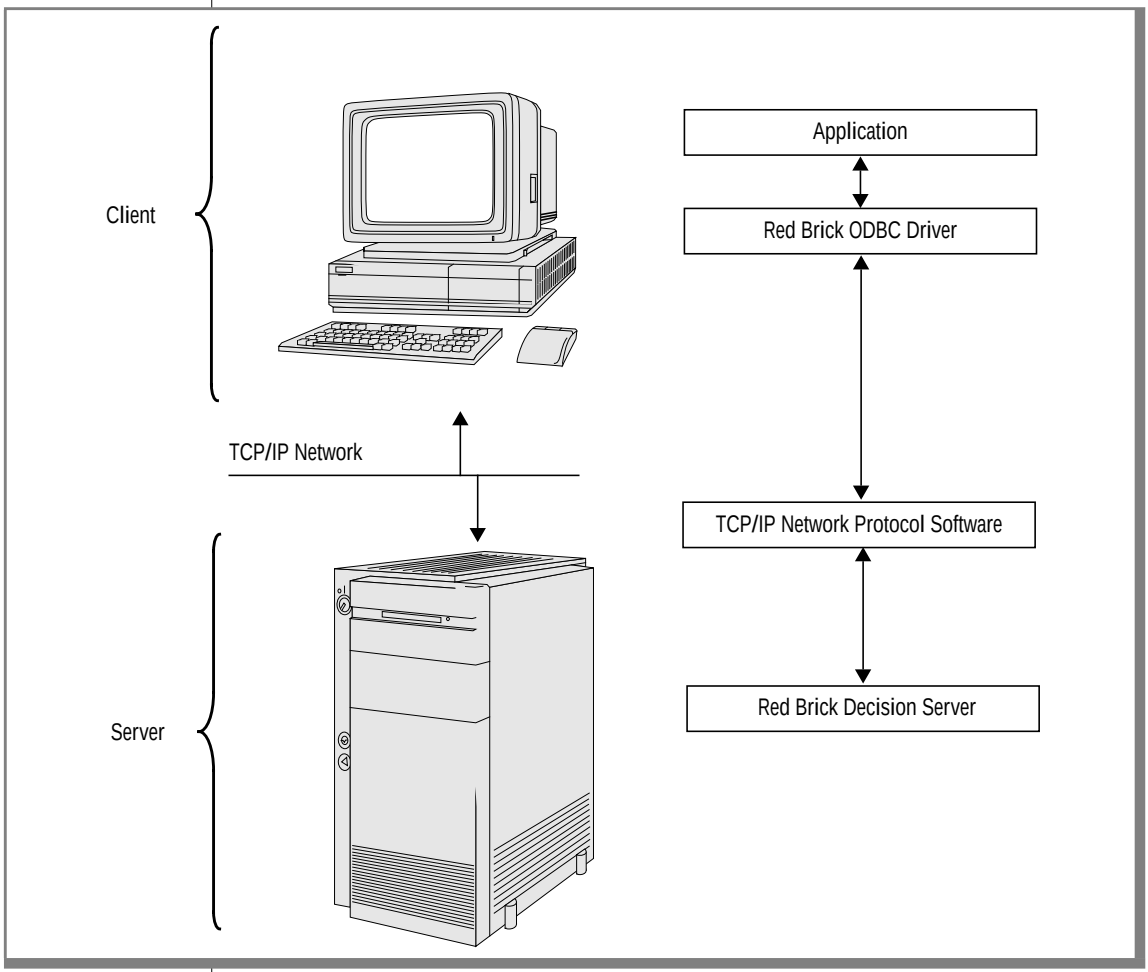
This chapter also describes the following features of Red Brick ODBC Driver:

For detailed information about the Red Brick ODBC Driver API, SQL grammar and syntax, Red Brick ODBC Driver data types, and Red Brick ODBC Driver scalar functions, see the *Microsoft* ODBC *2.0 Programmer's Reference*.

For more information about the functions and variables that Red Brick Decision Server supports, see the *SQL Reference Guide*.

Driver Architecture

Red Brick Decision Server is a relational database management system (RDBMS) designed for data warehouse, data mart, and online analytical processing (OLAP) applications. A Red Brick Decision Server database can be accessed with the RISQL Entry Tool or other client application through an Red Brick ODBC Driver interface. The following illustration describes the layers of communication software involved in Red Brick Decision Server access.



For more information about Red Brick Decision Server architecture and connectivity, see your *Administrator's Guide*.

Requirements

The following sections describe the system, stack size, and environment variable requirements for Red Brick ODBC Driver.

Software Requirements

Red Brick ODBC Driver requires the following system configuration.

Version of Red Brick Warehouse or Red Brick Decision Server	Driver to Use	Operating System to Use
Red Brick Warehouse Server, Version 5.1.2; Red Brick Decision Server, Version 6.0	Red Brick ODBC Driver (16 bit, V 4.01.x)	Recommended for Windows 3.1; supports Windows 95, Windows 98, and Windows NT
	Red Brick ODBC Driver (32 bit, V 5.01.x)	Recommended for Windows 95, Windows 98, and Windows NT
	Red Brick ODBC Driver (32 bit, V 6.01.x)	UNIX

The following table shows the recommended TCP/IP stacks for your operating system.

Client-Side Operating System	Recommended or Required TCP/IP Stack
Windows 3.1	Supports third-party, WinSock-compliant TCP/IP stacks. Contact Informix Customer Support for a complete list of approved stacks.
Windows 95, Windows 98, and Windows NT	Requires an integral WinSock-compliant TCP/IP stack.
UNIX	See the documentation notes or contact Informix Customer Support for a complete list of approved stacks.

Stack Size Requirements

Red Brick ODBC Driver shares stack space with the calling application and requires a minimum of 8 kilobytes of available space from the application.

Environment Variable Requirements

To make an application behave in conformance with different versions of the Red Brick ODBC Driver, change the **SQL_ATTR_ODBC_VERSION** environment variable, as follows:

- To make an application function at the Red Brick ODBC Driver, Version 2.x, level, set the **SQL_ATTR_ODBC_VERSION** environment variable to `SQL_OV_ODBC2`.
- To make an application function at the Red Brick ODBC Driver, Version 3.x, level, set the **SQL_ATTR_ODBC_VERSION** environment variable to `SQL_OV_ODBC3`.

  Only SQLSTATES; date, time, and timestamp data types; and catalog name arguments in SQL tables conform to Version 3.x.

ODBC API Conformance and Implementation

Red Brick ODBC Driver fully conforms to the Red Brick ODBC Driver Level 1 API and includes some Level 2 calls. The following table lists the Red Brick ODBC Driver API functions that Red Brick ODBC Driver supports and, if applicable, the Red Brick implementation and error status of those functions. If there is no corresponding error status, the driver returns the same value that Microsoft ODBC driver returns. Functions with more complex implementation are listed in the section that follows the table.

ODBC Function	Conformance Level	Red Brick Implementation/ Error Status
SQLAllocConnect	Core	
SQLAllocEnv	Core	
SQLAllocStmt	Core	
SQLBindCol	Core	
SQLBindParameter	Level 1	
SQLCancel	Core	
SQLColumnPrivileges	Level 2	
SQLColumns	Level 1	
SQLConnect	Core	
SQLDataSources	Level 2	
SQLDescribeCol	Core	
SQLDescribeParam	Level 2	If called after **SQLExecDirect**, will not return parameter description or number of parameters
SQLDisconnect	Core	
SQLDriverConnect	Level 1	
SQLDrivers	Level 2	

(1 of 3)

ODBC Function	Conformance Level	Red Brick Implementation/ Error Status
SQLError	Core	SQLSTATE S1000 (Generic Error)
SQLExecDirect	Core	
SQLExecute	Core	For each dynamic parameter in a statement, Red Brick ODBC Driver: ■ converts the value stored in the application's buffer to literal syntax with the correct prefix and suffix. ■ substitutes this literal for the parameter marker in the SQL statement that it executes.
SQLFetch	Core	
SQLForeignKeys	Level 2	
SQLFreeConnect	Core	
SQLFreeEnv	Core	
SQLFreeStmt	Core	
SQLGetConnectOption	Level 1	
SQLGetCursorName	Core	
SQLGetData	Level 1	
SQLGetFunctions	Level 1	
SQLGetInfo	Level 1	
SQLGetStmtOption	Level 1	
SQLGetTypeInfo	Level 1	
SQLMoreResults	Level 2	SQL_NO_DATA_FOUND
SQLNativeSql	Level 2	

(2 of 3)

ODBC Function	Conformance Level	Red Brick Implementation/ Error Status
SQLNumResultCols	Core	
SQLParamData	Level 1	
SQLPrepare	Core	
SQLPrimaryKeys	Level 2	
SQLPutData	Level 1	
SQLRowCount	Core	
SQLSetConnectOption	Level 1	
SQLSetCursorName	Core	
SQLSetParam	Core	
SQLSetStmtOption	Level 1	
SQLSpecialColumns	Level 1	
SQLStatistics	Level 1	
SQLTablePrivileges	Level 2	
SQLTables	Level 1	
SQLTransact	Core	S1C00 Driver not capable

(3 of 3)

The following sections provide additional Red Brick implementation information.

SQLGetConnectOption and SQLSetConnectOption

The following table lists the options that Red Brick ODBC Driver supports for the **SQLGetConnectOption** and **SQLSetConnectOption** functions.

fOption	Default Values
SQL_ACCESS_MODE	SQL_MODE_READ_WRITE
SQL_AUTOCOMMIT	1 (TRUE)
SQL_OPT_TRACE	The driver manager handles the value.
SQL_OPT_TRACEFILE	The driver manager handles the value.
SQL_TXN_ISOLATION *	0 (FALSE). You can read this option but not set it.

SQLGetCursorName and SQLSetCursorName

The **SQLGetCursorName** and the **SQLSetCursorName** functions support cursor names that are labels. These cursor names cannot be used for WHERE CURRENT OF clauses. Red Brick Decision Server does not support positioned updates or deletes.

SQLGetInfo

The following table lists the **SQLGetInfo** function options that Red Brick ODBC Driver returns.

fInfoType	Return Values
SQL_ACCESSIBLE_TABLES	"N"
SQL_ACCESSIBLE_PROCEDURES	"N"
SQL_ACTIVE_CONNECTIONS	0
SQL_ACTIVE_STATEMENTS	1
SQL_COFNCAT_NULL_BEHAVIOR	SQL_CB_NULL
SQL_CONVERT_BIGINT	0

(1 of 6)

fInfoType	Return Values
SQL_CONVERT_BINARY	0
SQL_CONVERT_BIT	0
SQL_CONVERT_CHAR	SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, SQL_CVT_TIME, SQL_CVT_DATE, SQL_CVT_FLOAT, or SQL_CVT_TIMESTAMP
SQL_CONVERT_DATE	SQL_CVT_CHAR, SQL_CVT_DATE
SQL_CONVERT_DECIMAL	SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT
SQL_CONVERT_DOUBLE	SQL_CVT_CHAR, SQL_CVT_FLOAT, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, or SQL_CVT_REAL
SQL_CONVERT_FLOAT	SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT
SQL_CONVERT_FUNCTIONS	SQL_FN_CVT_CONVERT
SQL_CONVERT_INTEGER	SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT
SQL_CONVERT_LONGVARBINARY	0
SQL_CONVERT_LONGVARCHAR	0

(2 of 6)

fInfoType	Return Values
SQL_CONVERT_NUMERIC	SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT
SQL_CONVERT_REAL	SQL_CVT_CHAR, SQL_CVT_DECIMAL, SQL_CVT_FLOAT, SQL_CVT_INTEGER, or SQL_CVT_REAL
SQL_CONVERT_SMALLINT	SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT
SQL_CONVERT_TIME	SQL_CVT_CHAR or SQL_CVT_TIME
SQL_CONVERT_TIMESTAMP	SQL_CVT_CHAR, SQL_CVT_TIME, or SQL_CVT_DATE
SQL_CONVERT_TINYINT	SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT
SQL_CONVERT_VARBINARY	0
SQL_CONVERT_VARCHAR	SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, SQL_CVT_FLOAT, SQL_CVT_TIMESTAMP
SQL_CORRELATION_NAME	SQL_CN_DIFFERENT
SQL_CURSOR_COMMIT_BEHAVIOR	SQL_CC_DELETE
SQL_CURSOR_ROLLBACK_BEHAVIOR	SQL_CR_DELETE

(3 of 6)

fInfoType	Return Values
SQL_DATA_SOURCE_NAME	Supplied at login
SQL_DATA_SOURCE_READ_ONLY	"N"
SQL_DATABASE_NAME	Supplied at login
SQL_DBMS_NAME	"Red Brick Decision Server"
SQL_DBMS_VER	Current server version
SQL_DEFAULT_TXN_ISOLATION	0
SQL_DRIVER_HDBC	The driver manager handles the value.
SQL_DRIVER_HENV	The driver manager handles the value.
SQL_DRIVER_HSTMT	The driver manager handles the value.
SQL_DRIVER_NAME	"RB16ODBC.DLL" or "RB32ODBC.DLL" or "RBODBC"
SQL_DRIVER_VER	Current driver version
SQL_EXPRESSIONS_IN_ORDERBY	"N"
SQL_FETCH_DIRECTION	SQL_FD_FETCH_NEXT
SQL_FILE_USAGE	SQL_FILE_NOT_SUPPORTED
SQL_IDENTIFIER_CASE	SQL_IC_UPPER
SQL_IDENTIFIER_QUOTE_CHAR	" " "
SQL_MAX_COLUMN_NAME_LEN	128
SQL_MAX_CURSOR_NAME_LEN	0
SQL_MAX_OWNER_NAME_LEN	0
SQL_MAX_PROCEDURE_NAME_LEN	0
SQL_MAX_QUALIFIER_NAME_LEN	0

(4 of 6)

fInfoType	Return Values
SQL_MAX_TABLE_NAME_LEN	128
SQL_MULT_RESULT_SETS	"N"
SQL_MULTIPLE_ACTIVE_TXN	"Y"
SQL_NON_NULLABLE_COLUMNS	SQL_NNC_NON_NULL
SQL_NUMERIC_FUNCTIONS	SQL_FN_NUM_ABS, SQL_FN_NUM_CEILING, SQL_FN_NUM_FLOOR, or SQL_FN_NUM_SIGN
SQL_ODBC_API_CONFORMANCE	SQL_OAC_LEVEL1
SQL_ODBC_SAG_CLI_CONFORMANCE	SQL_OSCC_COMPLIANT
SQL_ODBC_SQL_CONFORMANCE	SQL_OSC_MINIMUM
SQL_ODBC_SQL_OPT_IEF	"N"
SQL_ODBC_VER	The driver manager handles the value.
SQL_OUTER_JOINS	"Y"
SQL_OWNER_TERM	""
SQL_PROCEDURE_TERM	""
SQL_PROCEDURES	"N"
SQL_QUALIFIER_NAME_SEPARATOR	" "
SQL_QUALIFIER_TERM	""
SQL_ROW_UPDATES	"N"
SQL_SEARCH_PATTERN_ESCAPE	"\"
SQL_SERVER_NAME	Supplied at login
SQL_SCROLL_CONCURRENCY	SQL_SCCO_READ_ONLY
SQL_SCROLL_OPTIONS	SQL_SO_FORWARD_ONLY

(5 of 6)

fInfoType	Return Values
SQL_STRING_FUNCTIONS	SQL_FN_STR_CONCAT, SQL_FN_STR_LCASE, SQL_FN_STR_LEFT, SQL_FN_STR_LTRIM, SQL_FN_STR_RTRIM, SQL_FN_STR_SUBSTRING, or SQL_FN_STR_UCASE
SQL_SYSTEM_FUNCTIONS	SQL_FN_SYS_IFNULL or SQL_FN_SYS_USER
SQL_TABLE_TERM	"Table"
SQL_TIMEDATE_FUNCTIONS	SQL_FN_TD_CURDATE, SQL_FN_TD_CURTIME, SQL_FN_TD_DAYOFMONTH, SQL_FN_TD_DAYOFWEEK, SQL_FN_TD_DAYOFYEAR, SQL_FN_TD_HOUR, SQL_FN_TD_MINUTE, SQL_FN_TD_MONTH, SQL_FN_TD_NOW, SQL_FN_TD_QUARTER, SQL_FN_TD_SECOND, SQL_FN_TD_WEEK, or SQL_FN_TD_YEAR
SQL_TXN_CAPABLE	SQL_TC_NONE
SQL_TXN_ISOLATION_OPTION	0
SQL_USER_NAME	Supplied at login

(6 of 6)

SQLGetStmtOption and SQLSetStmtOption

The following table lists the options that Red Brick ODBC Driver supports for **SQLGetStmtOption** and **SQLSetStmtOption** functions.

fOption	Default Values
SQL_ASYNC_ENABLE	0 (FALSE)
SQL_BIND_TYPE	SQL_BIND_BY_COLUMN
SQL_MAX_LENGTH	0
SQL_MAX_ROWS	0

SQL Conformance

Red Brick ODBC Driver supports most Red Brick ODBC Driver Core SQL grammar for Microsoft ODBC Driver, Version 2.0. The following section lists exceptions to core conformance.

Minimum SQL Grammar

Red Brick ODBC Driver supports full Minimum SQL Grammar with the following exceptions.

Minimum SQL Grammar	Non-Supported Driver Options
CREATE TABLE	The driver does not support the IEF grammar CHECK() clause.
DROP TABLE	The driver does not support the IEF grammar CASCADE | RESTRICT clause.

Core SQL Grammar

Red Brick ODBC Driver supports the full Core SQL Grammar with the following exceptions.

Core SQL Grammar	Non-Supported Driver Options
ALTER TABLE	Red Brick SQL requires a DEFAULT clause when the application adds columns. Red Brick ODBC Driver grammar does not include this clause.
CREATE INDEX	UNIQUE and ASC / DESC
DELETE	Positioned deletes (WHERE CURRENT OF cursor-name)
DROP VIEW	IEF grammar CASCADE | RESTRICT
GRANT	IEF grammar REFERENCES()
REVOKE	IEF grammar CASCADE | RESTRICT
SELECT	FOR UPDATE OF
UPDATE	Positioned updates (WHERE CURRENT OF cursor-name)

Extended SQL Grammar

Red Brick ODBC Driver does not support Extended SQL Grammar for Microsoft Red Brick ODBC Driver, Version 2.0.

SQL Extensions

Red Brick ODBC Driver supports the following SQL extensions in the form of vendor strings (escape clauses) using both standard and extended syntax. Because Red Brick ODBC Driver is a pass-through technology, the driver accepts SQL extensions, converts them to native server syntax, and sends them to the server. The server then processes the extensions.

Datetime Extensions

Red Brick ODBC Driver supports all datetime extensions.

Outer-Join Extensions

Red Brick ODBC Driver supports outer-join extensions.

Scalar Function Extensions

Red Brick ODBC Driver supports scalar function extensions.

Tip: *You can store Red Brick Decision Server macros in a database to simulate the standard Red Brick ODBC Driver syntax for scalar and datetime functions. For more information on macros, see the "SQL Reference Guide."*

String Functions

Red Brick ODBC Driver supports the following string functions.

ODBC Function	Red Brick Equivalent
CONCAT(s1, s2)	**CONCAT(s1, s2)**
LCASE(s)	**LOWER(s)**
LEFT	**SUBSTR(*arg1*, 1, *arg2*)**
LTRIM(s)	**LTRIM(s)**

(1 of 2)

ODBC Function	Red Brick Equivalent
RTRIM(s)	**RTRIM(s)**
SUBSTRING(s, st, ln)	**SUBSTR(s, st, ln)**
UCASE(s)	**UPPER(s)**

(2 of 2)

Numeric Functions

Red Brick ODBC Driver supports the following Red Brick ODBC Driver numeric functions.

ODBC Function	Red Brick Equivalent
ABS(n)	**ABS(n)**
CEILING(n)	**CEIL(n)**
FLOOR(n)	**FLOOR(n)**
SIGN(n)	**SIGN(n)**

Date Functions

Red Brick ODBC Driver supports the following datetime functions.

ODBC Function	Red Brick Equivalent
CURDATE	**CURRENT_DATE**
CURTIME	**CURRENT_TIME**
DAYOFMONTH	**EXTRACT(DAY FROM *datetime_expression*)**
DAYOFWEEK	**EXTRACT(WEEKDAY FROM *datetime_expression*)**
DAYOFYEAR	**EXTRACT(DAYOFYEAR FROM *datetime_expression*)**
HOUR	**EXTRACT(HOUR FROM *datetime_expression*)**
MINUTE	**EXTRACT(MINUTE FROM *datetime_expression*)**

(1 of 2)

ODBC Function	Red Brick Equivalent
MONTH	**EXTRACT(MONTH FROM *datetime_expression*)**
NOW	**CURRENT_TIMESTAMP(6)**
QUARTER	**EXTRACT(QUARTER FROM *datetime_expression*)**
SECOND	**EXTRACT(SECOND FROM *datetime_expression*)**
WEEK	**EXTRACT(WEEK FROM *datetime_expression*)**
YEAR	**EXTRACT(YEAR FROM *datetime_expression*)**

(2 of 2)

System Functions

Red Brick ODBC Driver supports the following system functions.

ODBC Function	Red Brick Equivalent
IFNULL(exp, val)	**IFNULL(exp, val)**
USER( )	**CURRENT_USER**

Conversion Functions

Red Brick ODBC Driver supports the following Red Brick ODBC Driver conversion functions.

ODBC Function and Type	Red Brick Equivalent
CONVERT(*value*, SQL_CHAR)	**STRING(*value*)**
CONVERT(*value*, SQL_DATE)	**DATE(*value*)**
CONVERT(*value*, SQL_DECIMAL)	**DEC(*expression*)**
CONVERT(*value*, SQL_FLOAT)	**FLOAT(*value*)**
CONVERT(*value*, SQL_INTEGER)	**INT(*value*)**

(1 of 2)

ODBC Function and Type	Red Brick Equivalent
CONVERT(*value*, SQL_REAL)	**REAL(*value*)**
CONVERT(*value*, SQL_TIME)	**TIME(*value*)**
CONVERT(*value*, SQL_TIMESTAMP)	**TIMESTAMP(*value*)**

(2 of 2)

ODBC Data Types

The following table describes the data-type mapping that Red Brick ODBC Driver performs between Red Brick and Microsoft Red Brick ODBC Driver data types. If the user does not specify any other data-type conversion when the driver calls SQLGetData or SQLBindCol, then Red Brick ODBC Driver converts Red Brick data types to the indicated Red Brick ODBC Driver type.

Red Brick Data Types	ODBC SQL Data Type	Default ODBC C Data Type
CHAR	SQL_CHAR	SQL_C_CHAR
DATE	SQL_DATE	SQL_C_DATE
DOUBLE, FLOAT	SQL_DOUBLE, SQL_FLOAT ODBC	SQL_C_DOUBLE
INTEGER	SQL_INTEGER	SQL_C_SLONG
NUMERIC, DECIMAL	SQL_DECIMAL, SQL_NUMERIC	SQL_C_CHAR
REAL	SQL_REAL	SQL_C_FLOAT
SMALLINT	SQL_SMALLINT	SQL_C_SSHORT
TIME	SQL_TIME	SQL_C_TIME
TIMESTAMP	SQL_TIMESTAMP	SQL_C_TIMESTAMP
TINYINT	SQL_TINYINT	SQL_C_STINYINT
VARCHAR	SQL_VARCHAR	SQL_C_VARCHAR

Chapter

2

Building a Custom Client Application

UNIX

In This Chapter

This chapter explains how to use the Red Brick ODBClib software development kit (SDK) and MERANT DataDirect ODBC Driver to build a custom Red Brick ODBC Driver-based client application on UNIX. You can then use the client application, which can run on UNIX or Windows platforms, to access Informix Red Brick Decision Server.

Use MERANT DataDirect ODBC Driver to:

- develop custom Red Brick ODBC Driver applications that connect to Red Brick Decision Server.

  You must use Red Brick ODBC Driver with your custom applications.
- build a UNIX-based application.

For more information on MERANT DataDirect ODBC Driver, see MERANT documentation.

Important: *Red Brick does not support 16-bit platforms on UNIX*.

Use Microsoft Red Brick ODBC Driver SDK to build a Windows-based application. See Microsoft documentation for more information on Microsoft Red Brick ODBC Driver SDK and on how to develop applications to interface with Red Brick ODBC Driver.

The information in this chapter is divided into the following main sections:

For information on the sample Red Brick ODBC Driver program shipped with Red Brick ODBClib SDK on UNIX platforms, see Chapter 3, "Using the Sample ODBC Program."

Important: *Use Red Brick 32-bit Red Brick ODBC Driver driver even if you use a 64-bit platform.*

Using Red Brick ODBClib SDK

This section explains how to compile and link a custom Red Brick ODBC Driver program using Red Brick ODBClib SDK.

Red Brick ODBClib SDK uses libraries to link to an Red Brick ODBC Driver program. See the descriptions of Red Brick ODBC Driver-level conformance in Chapter 1, "Red Brick ODBC Driver Description and Implementation."

Red Brick ODBClib SDK files are located in the **redbrick_dir** directory.

C Compilers

Red Brick ODBClib SDK supports the following C compilers.

Platform	Compilers
Digital AlphaServer	Digital UNIX C Compiler
HP 9000	HP-UX C Compiler
IBM RISC System / 6000	AIX Compiler (XL C Compiler V1.3.0.19)
NCR UNIX SVR4 MP-RAS	High Performance C compiler
Sequent Symmetry DYNIX / ptx	PTX ANSI C Compiler
Silicon Graphics IRIX	MIPS Compiler
Sun SPARC-based System	Sun Compiler (SC V3.0.1 gcc V2.7)

Software Files

The **redbrick_dir** directory contains all the Red Brick software files. The following table list the files that MERANT DataDirect ODBC Driver and Red Brick ODBC Driver use for each platform that they support. The **xx** suffix designates platform-specific file extensions in the **lib** subdirectory.

Location	Filename	File Contents
lib subdirectory	**libodbc.xx**	MERANT ODBC driver manager
lib subdirectory	**libodbcinst.xx**	MERANT ODBC installer lib
lib subdirectory	**libivmback.xx**	SQLDriver Connect Motif lib
lib subdirectory	**libivmfront.xx**	SQLDriver Connect Motif lib
lib subdirectory	**vscnctdlg.so.1**	SQLDriver Connect Motif lib (SunOS only)
lib subdirectory	**odbctrac.xx**	MERANT ODBC tracing lib
lib subdirectory	**odbccurs.so.1**	MERANT ODBC cursors lib (SunOS only)
lib subdirectory	**rbodbcdrv.xx**	Red Brick ODBC Driver
redbrick_dir/messages/default/	**odbc.m**	MERANT error messages
redbrick_dir/locale/	***.nls**	Locale files for Red Brick ODBC Driver

ODBClib Libraries and Header Files

The following table list the Red Brick ODBC Driverlib library files and header files in the **redbrick_dir** directory. The **xx** suffix designates platform-specific file extensions in the **lib** subdirectory. Some platforms that have both 32-bit and 64-bit systems have multiple **lib** subdirectories. See "Compiling and Linking with Red Brick ODBClib."

Location	Filename	File Contents
lib subdirectory	**librbodbc.a**	Red Brick ODBClib static library
lib subdirectory	**librbodbc.xx**	Red Brick ODBClib shared library for non-AIX platforms
lib subdirectory	**librbodbcshared.a**	Red Brick ODBClib shared library for AIX only
include subdirectory	**rbsql.h**	Red Brick header file that contains all Red Brick ODBC Driver functions
include subdirectory	**rbsqlext.h**	Red Brick header file

Compiling and Linking with Red Brick ODBClib

This section provides information on the libraries and header files that you need to build an application with Red Brick ODBClib. Informix recommends that you use shared libraries because using them causes a decrease in the size of the executable file, increases link speed, and facilitates the use of newer versions of the libraries without the need to relink the application. The static library does not have to be available at runtime.

Platform-Specific Standard Libraries

The following table lists the platform-specific standard libraries that you should link with your application.

Platform	Libraries
Digital AlphaServer	**libpthreads.a, libm.a**
HP 9000 (32-bit compiler)	**libm.a, libBSD.a**
HP 9000 (64-bit compiler)	**libm.a**
IBM RISC System 6000 (32-bit and 64-bit compilers)	**libm.a**
NCR UNIX SVR4 MP-RAS	**libsocket.a, libnet.a, libnsl.a, libm.a**
Sequent Symmetry DYNIX ptx	**libmalloc.a, libsec.a, libseq.a, libsocket.a, libnsl.a, libm.a**
Silicon Graphics IRIX	**libm.a**
Sun SPARC-based System	**libsocket.a, libnsl.a, libm.a**

Compile and Link Lines

This section lists platform-specific definitions and sample commands that you should use to compile and link a file with Red Brick Red Brick ODBC Driver static libraries and header files. Sample commands use the following definitions:

```
# Directory where Red Brick ODBClib SDK is installed
REDBRICK_DIR=/redbrick_dir
#These compile examples assume the application C
#code is in 'rb_client.c' and executable will be 'rb_client'
```

Digital AlphaServer (Digital UNIX)

Definition:

```
CFLAGS=-I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lpthreads -lm
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lpthreads -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

HP 9000 (HP-UX 32-bit Compiler)

Definition:

```
CFLAGS=-Aa +z -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm -lBSD
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

HP 9000 (HP-UX 64-bit Compiler)

Definition:

```
CFLAGS=-Aa +DD64 -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

IBM RISC System/6000 (32-bit Compiler)

Definition:

```
CFLAGS=-I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbcshared.a -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

IBM RISC System/6000 (64-bit Compiler)

Definition:

```
CFLAGS= -q64 -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lm
```

Sample commands:

```
$ xlc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ xlc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

NCR WorldMark Server (NCR UNIX SVR4 MP-RAS)

Definition:

```
CFLAGS= -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lsocket -lnsl -lnet -lm #If
you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lsocket -lnsl -lnet -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

Sequent Symmetry System (DYNIX/ptx)

Definition:

```
CFLAGS= -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lmalloc -lsec \
-lseq -lsocket -lnsl -lm
# If you want to use Red Brick ODBClib shared library
# LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lmalloc -lsec \
#  -lseq -lsocket -lnsl -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

Silicon Graphics Server (IRIX 64-bit Compiler)

Definition:

```
CFLAGS= -mips3 -DSGI -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
# Define CFLAGS for 64-bit compilation
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

Silicon Graphics Server (IRIX 32-bit New-Format Compiler)

Definition:

```
CFLAGS= -n32 -mips3 -DSGI -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
# Define CFLAGS for 32-bit new-format compilation
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

Silicon Graphics Server (IRIX 32-bit Old-Format Compiler)

Definition:

```
CFLAGS=-32 -DSGI -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
# Define CFLAGS for 32-bit old-format compilation
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

Sun SPARC-Based System (Solaris)

Definition:

```
CFLAGS= -I$(REDBRICK_DIR)/include

# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lnsl -lm
#If you want to use Red Brick ODBClib shared library
#define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lnsl -lm
```

Sample commands:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```

Running an ODBC Application on UNIX

After you build your Red Brick ODBC Driver application, you must configure data sources (DSNs) and verify that platform-specific, shared-library environment variables are set correctly.

Important: *For information on how to run Red Brick ODBC Driver applications on UNIX, see MERANT documentation.*

Configuring Data Sources in the $HOME/.odbc.ini File

You can configure DSNs on UNIX in the $HOME**/.odbc.ini** file in the home directory of each user.

To configure DSNs in the $HOME**/.odbc.ini** file, you must create a symbolic link between a central **.odbc.ini** file and the **.odbc.ini** file in the home directory of each user to centralize DSN definitions. If you do not create these symbolic links, you will not be able to connect to a database unless the connection string in your application specifically lists all the components of the DSN specification.

There are different **.odbc.ini** files for applications that use Red Brick ODBC Driver on UNIX and for applications that are linked with Red Brick ODBClib SDK. The following sections describe these **.odbc.ini** files.

Important: *If you share your **$HOME** directory across multiple hardware platforms, your **.odbc.ini** file might specify platform-specific information. For example, the **.odbc.ini** file could have different InstallDir specifications, which contain platform-specific directory paths to the location of Red Brick ODBClib SDK files.*

Initialization file for applications that use Red Brick ODBC Driver

```
# Sample $HOME/.odbc.ini file if you are using
# the MERANT ODBC Driver Manager
# to load the Red Brick ODBC Driver for UNIX
#

# This section is to name your ODBC DSNs
# One entry per DSN
#
[ODBC Data Sources]
RBDSN=Red Brick Driver

# This section defines the ODBC environment
#
[ODBC]
Trace=<value>
TraceFile=<filename>
InstallDir=<redbrick_dir>/lib


# This section is to define your ODBC DSNs
# One set of entries per DSN
#
[RBDSN]
Driver=<redbrick_dir>/lib/rbodbcdrv.so
SERVER=<host>:<port>
RB_CONFIG=<redbrick_dir>
DATABASE=<database_name>
UID=<username>
PWD=<password (optional)>
```

Initialization file for applications that use Red Brick ODBClib SDK

```
# Sample $HOME/.odbc.ini file if you are using
# the Red Brick ODBClib SDK or if you are using
# the RISQL Entry Tool or RISQL Reporter
#
# This section is to name your ODBC DSNs
# One entry per DSN
#
[ODBC Data Sources]
RBDSN=Red Brick Driver


# This section defines the ODBC environment
#
[ODBC]
InstallDir=<redbrick_dir>/lib

# This section is to define your ODBC DSNs
# One set of entries per DSN
#
```

```
[RBDSN]
SERVER=<host>:<port>
RB_CONFIG=<redbrick_dir>
DATABASE=<database_name>
UID=<username>
PWD=<password (optional)>
```

This sample **.odbc.ini** file defines a DSN with the name RBDSN.

Setting and Configuring Platform-Specific Environment Variables

Set the **RB_CONFIG** environment variable before you run an Red Brick ODBC Driver application on UNIX. The Red Brick ODBC Driver application uses the information in this environment variable to find the **rbw.config** file, which is used to locate message files and to determine locale settings.

Set the **RB_CONFIG** environment variable as follows:

```
setenv RB_CONFIG redbrick_dir
```

The **redbrick_dir** directory is the directory in which Red Brick **ODBClib SDK** files are installed.

If you create an application that uses Red Brick **ODBClib** shared library, you must set platform-specific environment variables as described in the following table, where **redbrick_dir** is the directory in which Red Brick ODBClib SDK files are installed.

Platform	Environment Variable Settings
Digital AlphaServer	setenv LD_LIBRARY_PATH <*redbrick_dir*> / lib:$LD_LIBRARY_PATH
HP 9000	setenv SHLIB_PATH <*redbrick_dir*> / lib:$SHLIB_PATH
IBM RISC System / 6000	setenv LIBPATH <*redbrick_dir*> / lib:$LIB_PATH
NCR UNIX SVR4 MP-RAS	setenv LD_LIBRARY_PATH <*redbrick_dir*> / lib:$LD_LIBRARY_PATH
Sequent Symmetry DYNIX / ptx	setenv LIBRARY_PATH <*redbrick_dir*> / lib:$LD_LIBRARY_PATH
Silicon Graphics IRIX	setenv LD_LIBRARY_PATH <*redbrick_dir*> / lib:$LD_LIBRARY_PATH
Sun SPARC-based System	setenv LD_LIBRARY_PATH <*redbrick_dir*> / lib:$LD_LIBRARY_PATH

Functional Differences Between Red Brick and MERANT SDKs

This section describes functional differences between Red Brick ODBClib SDK and MERANT DataDirect Connect Red Brick ODBC Driver SDK. Use this information when you create applications that will be linked with the Red Brick ODBClib library.

Header Files

You must use the **rbsql.h** header file (included with Red Brick ODBClib SDK) if you write applications for Red Brick ODBClib SDK. You must use the **sql.h** header file (available from Microsoft or MERANT) if you write applications for Microsoft or MERANT DataDirect Connect Red Brick ODBC Driver SDKs.

DSN Connections

MERANT provides drivers for DSNs. If you build an Red Brick ODBC Driver application using MERANT DataDirect Connect Red Brick ODBC Driver SDK, that application can connect to any DSN that has an MERANT driver.

If you build an Red Brick ODBC Driver application using Red Brick ODBClib SDK, you can only connect to Red Brick Warehouse, Version 5.0.10 or later, or Red Brick Decision Server, Version 6.0.

Source-Code Portability

To port Red Brick ODBClib SDK applications to MERANT DataDirect Connect Red Brick ODBC Driver SDK, use MERANT header files to recompile the applications and use MERANT Driver Manager to link the files.

To port MERANT DataDirect Connect Red Brick ODBC Driver applications to Red Brick ODBClib SDK, use Red Brick header files to recompile the applications and use the Red Brick **ODBClib** library to link the files.

Driver Manager Function Emulation

Red Brick ODBClib SDK emulates the following functions that Microsoft Driver Manager or MERANT Driver Manager performs:

- **SQLDataSources**
- **SQLDrivers**
- **SQLGetInfo**

SQLDataSources

If you have an **.odbc.ini** file or symbolic link in your **$HOME** directory, Red Brick ODBClib SDK implements **SQLDataSources** in the same manner that MERANT Driver Manager implements **SQLDataSources** for an application created with MERANT DataDirect Connect Red Brick ODBC Driver SDK.

SQLDrivers

If you use Red Brick ODBClib SDK to create an application, that application can only use Red Brick ODBC Driver. Red Brick ODBClib SDK does not try to use any other drivers.

SQLGetInfo

MERANT SDK returns the same **SQLGetInfo** function values as Microsoft SDK.

Red Brick ODBClib SDK returns the same **SQLGetInfo** function values as the values documented in "SQLGetInfo" on page 1-10, with the exceptions that the following table shows.

fInfoType	Return Values
SQL_DRIVER_HDBC	Handle that is visible to the application. The driver manager is not involved.
SQL_DRIVER_HENV	Handle that is visible to the application. The driver manager is not involved.
SQL_DRIVER_HLIB	String value: "0" The application communicates directly with the driver.
SQL_DRIVER_HSTMT	Handle that is visible to the application. The driver manager is not involved.
SQL_DRIVER_VER	String value: "02.00" This value is derived from SQL_SPEC_MAJOR and SQL_SPEC_MINOR

Chapter

3

Using the Sample ODBC Program

UNIX

In This Chapter

Red Brick ships a sample Red Brick ODBC Driver C program called **rb_client** as part of the Red Brick Red Brick ODBC Driverlib SDK for UNIX platforms. This chapter contains information on the location of the sample program, as well as instructions to set up the program.

Location of the Sample C Program

The **redbrick_dir/lib/example** directory contains the source code for the sample C program. The **redbrick_dir** directory is the directory where the Red Brick Red Brick ODBC Driverlib SDK files are installed.

The **example** directory contains the following files:

- **README.TXT**, which includes an overview of how to build the sample program
- **rb_client.c**, which contains the sample C code
- **Makefile**, which is used with the **make** utility to define the environment in which the program is built

Important: *The 32-bit Silicon Graphics version of Red Brick ODBC Driverlib SDK does not have a sample program.*

Setting Up the Sample Program

The **rb_client** program is a simple Red Brick ODBC Driver client program that submits SQL statements to a Red Brick database. Modify **Makefile** for your platform-specific information in a text editor, such as **vi**, and then use **Makefile** to build your program.

Defining the Platform

Find the **RB_PLATFORM** environment variable that corresponds to your platform at the top of **Makefile** and remove the comment symbol (#) from the line. For example, if your platform is Solaris, remove the comment symbol from the following line in **Makefile**:

```
# RB_PLATFORM=solaris
```

Important: By default, a comment signal appears in front of all **RB_PLATFORM** environment variables in **Makefile**.

Defining the Red Brick ODBClib SDK

Make sure that **Makefile** contains the following line:

```
RB_ODBC=1
```

Defining the Installation Directory

Find the **RB_INSTALL_DIR** environment variable in **Makefile** and define it to reference the full directory path of **redbrick_dir**. For example, if you installed the SDK in the **/redbrick/odbc** directory, you should define the **RB_INSTALL_DIR** environment variable as follows:

```
RB_INSTALL_DIR= /redbrick/odbc
```

Specifying the Preprocessor, Archiver, and Compiler

Makefile allows you to specify the path to the executables for a C preprocessor, an object archiver, and a C compiler. The defaults will work correctly in most environments, but if you want to specify different executables for these tools, edit the **CPP**, **AR**, and **CC** variables for the C preprocessor, object archiver, and C compiler, respectively. For example, the default values for the **CPP**, **AR**, and **CC** variables are:

```
CPP= /odbc/cpp.exe
AR= /odbc/ar.exe
CC= /odbc/cc.exe
```

Replace the default values with the path to your executable.

Building the rb_client Program

After you edit the **Makefile** to reflect your platform-specific and environment-specific information, you can build the **rb_client** program. Before you build the program, make sure that you have done the following:

- Defined the data source name (DSN) in the **$HOME/.odbc.ini** file
- Set the **RB_CONFIG** environment variable

To build the **rb_client** program, change directories to the **$RB_CONFIG/lib/example** directory (the directory that contains Red Brick **Red Brick ODBC** Driver**lib SDK** files) and enter the following command:

```
% make
```

After you enter the **make** command, the **rb_client** program builds an executable called **rb_client** and asks you to enter a DSN and an SQL statement. The **rb_client** program then executes the SQL statement on the database that corresponds to the DSN that you specified.

Chapter

4

Adding, Configuring, and Deleting Data Sources

Windows

In This Chapter

This chapter describes how to add, configure, reconfigure, and delete a data source (DSN) on Windows. A DSN is a logical name for the combination of a specific server installed on a host system, the name of a database on that host, and a database username.

The chapter also describes how to use RBPing (Red Brick Ping) to test a DSN.

For information on how to add, configure, and delete a DSN on UNIX, see "Running an ODBC Application on UNIX" on page 2-12.

Adding and Configuring a New DSN

To add and configure a new DSN, use the following procedure.

To add and configure a new DSN

1. Choose **Start→Programs→ODBC** in **Client16** to display the Data Sources dialog box.

   Choose **Start→Setting→Control Panel→ODBC Data Sources** in **Client32** to display the ODBC Data Source Administrator dialog box.

   In **Client32,** choose an option to modify your DSN:

   - Choose the User DSN option to restrict access and migration privileges to one user.
   - Choose the System DSN option to allow access to all system users.
   - Choose the File DSN option to allow access to users who have write privileges to the file.

2. Click the tab that corresponds to the option you chose in Step 1.
3. Click **Add** to display the Add Data Source dialog box in **Client16** or the Create New Data Source dialog box in **Client32**.
4. Select a driver from the list and click **OK**.

   The Red Brick ODBC Driver Data Source dialog box appears. Type the following information in the dialog box:

   - **Data Source Name**

     A data source name is a logical identifier for a DSN and can be composed of any character string that does not include the following characters:

     ( ) [ ] = ; { } , ? * ! @

   - **Server (Host:Port)**

     This is the host name and port number of the server that runs Red Brick Decision Server (for example, `RBDecision:7923`).

   - **Database Name**

     A database name is the logical identifier for a valid database on the host computer, as defined by the database administrator (for example, `Aroma`).

   - **Default User Name**

     A default user name is a database username that allows access to the database, such as a system username.

5. Click **OK** to display the Data Sources dialog box in **Client16** or the ODBC Data Source Administrator dialog box in **Client32**.

   The new DSN you defined appears on the list.

   After you add a DSN, the name appears in the connectivity component of an ODBC-compliant tool.

6. In **Client16** only, click **ODBC Options** to select a tracing option and click OK.

If you install the CCP for the first time, the Setup Complete dialog box appears after you add a DSN. Click **Yes** to restart the computer now or **No** to restart the computer later.

Reconfiguring a DSN

To reconfigure a DSN, use the following procedure.

To reconfigure a DSN

1. Choose **Start→Programs→ODBC** in **Client16** to display the Data Sources dialog box.

   Choose **Start→Setting→Control Panel→ODBC Data Sources** in **Client32** to display the ODBC Data Source Administrator dialog box.
2. Select the DSN you want to change.
3. Click **Setup** in **Client16** or **Configure** in **Client32 to** display the Red Brick ODBC Driver Data Source dialog box.
4. Make changes to the DSN configuration information and click OK.

Deleting a DSN

To delete a DSN, use the following procedure.

To delete a DSN

1. Choose **Start→Programs→ODBC** in **Client16** to display the Data Sources dialog box.

   Choose **Start→Setting→Control Panel→ODBC Data Sources** in **Client32** to display the ODBC Data Source Administrator dialog box.
2. Select the DSN that you want to delete.
3. In **Client16,** click **Delete** to display the Microsoft ODBC Administrator question box.

   In **Client32,** click **Remove** to display the 32-bit ODBC Administrator question box.
4. Click **Yes** to confirm the deletion or **No** to cancel it.

Using RBPing to Test a DSN

To test a DSN with RBPing, use the following procedure.

To test a DSN

1. Choose **Start→Programs→Red Brick Client 32** or **Red Brick Client 16→RBPing32** or **RBPing16** in Windows 3.1 or Windows NT, Version 4.0.

   Choose **Start→Program Manager→Red Brick Client 32** or **Red Brick Client 16→RBPing32** or **RBPing16** in Windows 95 or Windows 98.

2. If you chose **Client16**, choose a DSN to test in the SQL Data Sources dialog box and click **OK**.

   If you chose **Client32**, choose a DSN to test in the RWPing dialog box and click **Ping**.

   The Status dialog box appears and displays information about the connection. If RBPing fails to connect with the DSN, the message explains the reason for the failure, such as a failure to load the driver.

Index

E

F

H

I

L

M

N

O

P

R

S

T

U

W

Symbols