Contents

About This Document
   Purpose v
   Audience v
   Organization vi
   Related Documentation vii
   Conventions ix
      Syntax Notation ix
   Customer Support x
      Support Solutions Warehouse x
      General and Technical Questions x
      Troubleshooting Tips xii
      Documentation Questions and Comments xii

1 Introduction to Red Brick Warehouse

2 Programmer's Notes for the Red Brick ODBC Driver
   System Requirements 2-2
   ODBC API Conformance 2-3
      SQL Error 2-4
      SQL Execute 2-5
      SQL Fetch 2-5
      SQL Get Connect Option, SQL Set Connect Option 2-5
      SQL Get Cursor Name, SQL Set Cursor Name 2-5
      SQL Get Info 2-6
      SQL Get Stmt Option, SQL Set Stmt Option 2-11
      SQL More Results 2-11
      SQL Prepare 2-11
      SQL Transact 2-11
   SQL Conformance 2-12
      Minimum SQL Grammar 2-12
      Core SQL Grammar 2-12
      Extended SQL Grammar 2-13
   SQL Extensions 2-13
      Datetime Extensions 2-13
      Outer-Join Extensions 2-13
      Scalar Function Extensions 2-13
   ODBC Datatypes 2-17
   Stack Size Requirements 2-17
Contents

3 Creating and Using a Custom ODBC Program
   Using the Red Brick ODBClib Interface 3-2
      Supported C Compilers 3-2
      ODBClib Libraries and Header Files 3-3
      Compiling and Linking with Red Brick ODBClib 3-3
   Running an ODBC Application on UNIX 3-9
      Setting the RB_CONFIG Environment Variable 3-9
      Configuring Data Sources in the $HOME/.odbc.ini File 3-9
      Configuring the Environment for Shared Libraries 3-11

4 Using the Sample ODBC Program
   Sample C Program 4-2
   Setting Up the Sample Program 4-2
      Defining the Platform 4-2
      Defining the Red Brick ODBClib SDK 4-2
      Defining the Installation Directory 4-3
      Specifying the Pre-Processor, Archiver, and Compiler 4-3
      Building the rb_client Application 4-3

Index
About This Document

Purpose

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

Audience

The intended users of this guide are software developers who are developing applications for use with Red Brick Warehouse. Knowledge of ODBC interfaces is assumed.
This guide contains the following information:

Chapter 1, “Introduction to Red Brick Warehouse,” describes Red Brick Warehouse and the connectivity architecture used for Red Brick Warehouse access.


Chapter 3, “Creating and Using a Custom ODBC Program,” describes how to compile and link a custom ODBC program.

Chapter 4, “Using the Sample ODBC Program,” describes how to use the sample ODBC program shipped with the Red Brick ODBClib SDK.
## Related Documentation

The standard documentation set for Red Brick Warehouse includes the following documents:

<table>
<thead>
<tr>
<th>Document</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Installation and Configuration Guide</strong></td>
<td>Installation and configuration information, as well as platform-specific material, about Red Brick Warehouse and related products. Customized for either UNIX-based or Windows NT systems.</td>
</tr>
<tr>
<td><strong>Warehouse Administrator’s Guide</strong></td>
<td>Description of warehouse architecture, supported schemas, and other concepts relevant to warehouse databases. Procedural information for designing and implementing a warehouse database, maintaining a database, and tuning a database for performance. Includes a description of the system tables and the configuration file (rbw.config). Customized for UNIX-based or Windows NT systems.</td>
</tr>
<tr>
<td><strong>Table Management Utility Reference Guide</strong></td>
<td>Description of 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.</td>
</tr>
<tr>
<td><strong>SQL Reference Guide</strong></td>
<td>Complete language reference for the Red Brick Systems SQL implementation and RISQL® extensions for warehouse databases.</td>
</tr>
<tr>
<td><strong>SQL Self-Study Guide</strong></td>
<td>Example-based review of SQL and introduction to the RISQL extensions, the macro facility, and Aroma, the sample database.</td>
</tr>
<tr>
<td><strong>RISQL Entry Tool and RISQL Reporter User’s Guide</strong></td>
<td>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-formating capabilities.</td>
</tr>
<tr>
<td><strong>Messages and Codes Reference Guide</strong></td>
<td>Complete listing of all informational, warning, and error messages generated by warehouse products, including probable causes and recommended responses. Also includes event log messages that are written to the log files.</td>
</tr>
<tr>
<td><strong>Release Notes</strong></td>
<td>Information pertinent to the current release that was unavailable when the documents were printed.</td>
</tr>
</tbody>
</table>
In addition to the standard documentation set, the following documents are included for specific sites:

**Red Brick Vista User’s Guide**
Description of the Red Brick Vista™ aggregate navigation and advice system, including procedures for rewriting queries and getting advice on the best set of aggregate tables and views to create. Includes detailed examples of queries whose performance can be dramatically increased by using aggregate navigation.

**SQL-BackTrack for Red Brick Warehouse User’s Guide**
Complete guide to SQL-BackTrack™ for Red Brick Warehouse, 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.

**Client Connector Pack Installation Guide**
Procedures to install and configure the Red Brick ODBC Driver, the RISQL Entry Tool, and the RISQL Reporter on client systems. Included for those sites that purchase the Client Connector Pack.

**ODBC Connectivity Guide**
Information about ODBC conformance levels as well as instructions on compiling and linking an ODBC application using the Red Brick ODBClib SDK.

**Red Brick Data Mine User’s Guide**
Description of the data mining process, and procedural information for using Red Brick Data Mine’s SQL-based interface to find hidden or unpredictable relationships among the data in a data set. Included for those sites that purchase the Red Brick Data Mine™ option.

**Red Brick Data Mine Builder™ User’s Guide**
Description of the data mining process, and procedural information for performing data mining using Red Brick’s GUI-based product in a Microsoft Windows environment.

Additional reference material you will find useful is the documentation for the ODBC Application Programming Interface (API), available from Microsoft Corporation (for developing Windows applications):

- *Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide*

**Online Documentation**

The English version of the Red Brick Warehouse documentation is also available in Adobe Acrobat format (PDF) on a separate CD-ROM.
Conventions

Throughout Red Brick Systems technical publications, the following notation and syntax conventions are used:

- 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, system-level commands, and variables appear in Palatino 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).

Syntax Notation

This guide uses the following conventions to describe the syntax of operating-system commands:

<table>
<thead>
<tr>
<th>Command Element</th>
<th>Example</th>
<th>Convention</th>
</tr>
</thead>
<tbody>
<tr>
<td>Values and parameters</td>
<td>table_name</td>
<td>Items that you replace with an appropriate name, value, or expression are in italic type style.</td>
</tr>
<tr>
<td>Optional items</td>
<td>[ ]</td>
<td>Optional items are enclosed by square brackets. Do not type the brackets.</td>
</tr>
<tr>
<td>Choices</td>
<td>ONE</td>
<td>TWO</td>
</tr>
<tr>
<td>Required choices</td>
<td>{ONE</td>
<td>TWO}</td>
</tr>
<tr>
<td>Default values</td>
<td>ONE</td>
<td>TWO</td>
</tr>
<tr>
<td>Repeating items</td>
<td>name, ...</td>
<td>Items that can be repeated are followed by a comma and an ellipsis. Separate the items with commas.</td>
</tr>
<tr>
<td>Language elements</td>
<td>( ) ; .</td>
<td>Parentheses, commas, semicolons, and periods are language elements. Use them exactly as shown.</td>
</tr>
</tbody>
</table>
Customer Support

Please review the following information before contacting the Customer Support Center at Red Brick Systems.

Support Solutions Warehouse

The Support Solutions Warehouse is the Customer Support Center’s external web site, an online resource that registered Red Brick customers can use to:

- Submit new cases.
- Read release notes.
- Find answers to frequently asked questions (FAQs).
- Search the Problems and Solutions database.

To use the Support Solutions Warehouse, point your web browser to the following URL and enter your registered username and password:

http://www.redbrick.com/RBCustomer/index.htm

If you do not have a registered username and password, contact the Customer Support Center by telephone, fax, or e-mail.

General and Technical Questions

If you have general sales-related questions or technical questions about Red Brick products or services, contact Red Brick Systems as follows:

Telephone
General Questions  (408) 399-3200 or 1 (800) 777-2585
Technical Questions (408) 399-7100 or 1 (800) 727-1866

FAX
General Questions  (408) 399-3277
Technical Questions (408) 399-3297

Internet e-mail
General Questions  info@redbrick.com
Technical Questions support@redbrick.com

World Wide Web  www.redbrick.com
**Existing Cases**

If you want to inquire about the status of an existing case, please have the case number ready. The case number will always be given to you by the support engineer who logs the case or first contacts you. This number is used to keep track of all the activities performed during the resolution of each problem.

**New Cases**

If you want to log a new case, please have the following information ready:

- Red Brick Warehouse version
- Platform and operating-system version
- Error messages returned by Red Brick Warehouse or the operating system
- Concise description of the problem, including any commands or operations performed prior to the occurrence of the error message
- List of Red Brick Warehouse and/or operating-system configuration changes made prior to the occurrence of the error message

If you think the problem concerns client-server connectivity, please have the following additional information ready:

- Name and version of the client tool in use
- Version of Red Brick ODBC Driver in use (if applicable)
- Name and version of client network and/or TCP/IP stack in use
- Error messages returned by the client application
- Warehouse and client locale specifications
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.

- For SQL query problems, try removing columns or functions, or restating WHERE, ORDER BY, or GROUP BY clauses until you can isolate the part of the statement causing the problem.
- For TMU load problems, verify the datatype mapping between the source file and the target table to ensure compatibility. Try loading a small test set of data to determine whether the problem concerns volume or data format.
- For connectivity problems, verify that the network is up and running by running the rbping utility from the client to the host. If possible, try another client tool to see if the same problem arises.

Documentation Questions and Comments

If you have questions or comments about the Red Brick Warehouse documentation, please contact the Technical Publications Department at Red Brick Systems as follows:

Telephone
+1 408 399 3200
+1 800 727 1866 (USA only)

Internet e-mail
docs@redbrick.com
Red Brick® Warehouse is a relational database management system (RDBMS) designed for data warehouse, data mart, and online analytical processing (OLAP) applications. Compared to online transaction processing (OLTP) or “universal” database products, Red Brick Warehouse delivers higher query-processing and data-loading performance, greater ease of administration, and a richer set of specialized features for applications that range from a few gigabytes to well over a terabyte, and from a few users to thousands of users.

Red Brick Warehouse can scale from the workgroup to the enterprise, is built for an open client/server environment, and is accessed using industry-standard SQL. The server’s RISQL® extensions simplify analyses that require ranks, ratios, and other commonly used business calculations, while the STARjoin™, STARindex™, TARGETjoin™, and TARGETindex™ technologies provide unparalleled ad hoc query and analysis performance against very large databases with various schema designs. Managers and analysts can pose numerous and creative queries to quickly receive the information they need, and make good business decisions with similar speed and confidence.
Introduction to Red Brick Warehouse

A Red Brick Warehouse database can be accessed with the RISQL Entry Tool, RISQL Reporter, or other client applications through an ODBC interface. Layers of communication software are shown in the following illustration:

For more information about Red Brick Warehouse architecture and connectivity, refer to the Warehouse Administrator’s Guide.
This chapter describes the Red Brick ODBC Driver with respect to:

- System Requirements
- ODBC API Conformance
- SQL Conformance
- SQL Extensions
- ODBC Datatypes
- Stack Size Requirements

For detailed information about the ODBC API, SQL grammar, ODBC datatypes, and ODBC scalar functions, refer to the Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide available from Microsoft Corporation.
System Requirements

The Red Brick ODBC Driver requires the following system configuration for operation.

<table>
<thead>
<tr>
<th>Red Brick Warehouse Version</th>
<th>Driver to Use</th>
<th>Operating System to Use</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1.2 or later</td>
<td>Red Brick ODBC-16 V 4.00.x</td>
<td>Windows 3.1 (primarily), Windows 95, or Windows NT</td>
</tr>
<tr>
<td></td>
<td>Red Brick ODBC-32 V 5.00.x</td>
<td>Windows 95 or Windows NT</td>
</tr>
</tbody>
</table>

The Red Brick ODBC Driver is layered on a supported TCP/IP stack.
- For Windows 95 and Windows NT, the integral WinSock-compliant TCP/IP stack is required.
- For Windows 3.1, several third-party WinSock-compliant TCP/IP stacks are supported. Contact Red Brick Systems for a list of supported TCP/IP stacks.
ODBC API Conformance

The Red Brick ODBC Driver fully conforms to the ODBC Level 1 API and includes some Level 2 calls. The following table lists the ODBC API functions that are supported.

<table>
<thead>
<tr>
<th>ODBC Function</th>
<th>Conformance Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQLAllocConnect</td>
<td>Core</td>
</tr>
<tr>
<td>SQLAllocEnv</td>
<td>Core</td>
</tr>
<tr>
<td>SQLAllocStmt</td>
<td>Core</td>
</tr>
<tr>
<td>SQLBindCol</td>
<td>Core</td>
</tr>
<tr>
<td>SQLBindParameter</td>
<td>Level 1</td>
</tr>
<tr>
<td>SQLCancel</td>
<td>Core</td>
</tr>
<tr>
<td>SQLColAttributes</td>
<td>Core</td>
</tr>
<tr>
<td>SQLColumns</td>
<td>Level 1</td>
</tr>
<tr>
<td>SQLConnect</td>
<td>Core</td>
</tr>
<tr>
<td>SQLDataSources</td>
<td>Level 2</td>
</tr>
<tr>
<td>SQLDescribeCol</td>
<td>Core</td>
</tr>
<tr>
<td>SQLDisconnect</td>
<td>Core</td>
</tr>
<tr>
<td>SQLDriverConnect</td>
<td>Level 1</td>
</tr>
<tr>
<td>SQLDrivers</td>
<td>Level 2</td>
</tr>
<tr>
<td>SQLError</td>
<td>Core</td>
</tr>
<tr>
<td>SQLExecDirect</td>
<td>Core</td>
</tr>
<tr>
<td>SQLExecute</td>
<td>Core</td>
</tr>
<tr>
<td>SQLFetch</td>
<td>Core</td>
</tr>
<tr>
<td>SQLFreeConnect</td>
<td>Core</td>
</tr>
<tr>
<td>SQLFreeEnv</td>
<td>Core</td>
</tr>
<tr>
<td>SQLFreeStmt</td>
<td>Core</td>
</tr>
<tr>
<td>SQLGetConnectOption</td>
<td>Level 1</td>
</tr>
<tr>
<td>SQLGetCursorName</td>
<td>Core</td>
</tr>
<tr>
<td>SQLGetData</td>
<td>Level 1</td>
</tr>
</tbody>
</table>
The following sections provide additional information about some of the functions listed in the ODBC conformance table.

**SQLError**

As many Red Brick-specific errors as possible are mapped to ODBC-defined SQLSTATE codes. If an error cannot be mapped directly, the SQLSTATE S1000 (Generic Error) is returned and error text is provided by Red Brick Warehouse.
**SQLExecute**

If a statement contains dynamic parameters, for each parameter the Red Brick ODBC Driver:

1. Converts the value stored in the application’s buffer to literal syntax with the correct prefix and suffix.
2. Substitutes this literal for the parameter marker in the SQL statement being executed.

**SQLFetch**

Only SQL_FETCH_NEXT is supported, which is consistent with Level 1 conformance.

**SQLGetConnectOption, SQLSetConnectOption**

The following table describes the options supported by the Red Brick ODBC Driver for SQLSetConnectOption and SQLGetConnectOption. Default values are shown where applicable.

<table>
<thead>
<tr>
<th>fOption</th>
<th>Default Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_ACCESS_MODE</td>
<td>SQL_MODE_READ_WRITE</td>
</tr>
<tr>
<td>SQL_AUTOCOMMIT</td>
<td>1 (TRUE)</td>
</tr>
<tr>
<td>SQL_OPT_TRACE</td>
<td>Handled by the ODBC Driver Manager</td>
</tr>
<tr>
<td>SQL_OPT_TRACEFILE</td>
<td>Handled by the ODBC Driver Manager</td>
</tr>
<tr>
<td>SQL_TXN_ISOLATION *</td>
<td>0 (FALSE)</td>
</tr>
</tbody>
</table>

Options marked with an asterisk (*) can be read but not set.

**SQLGetCursorName, SQLSetCursorName**

The cursor names supported by these functions are labels; they cannot be used for WHERE CURRENT OF clauses. Red Brick Warehouse does not support positioned updates or deletes.
### SQLGetInfo

The following table lists the values returned by the Red Brick ODBC Driver for the SQLGetInfo options. Default values are shown where applicable.

<table>
<thead>
<tr>
<th>fInfoType</th>
<th>Return Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_ACCESSIBLE_TABLES</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_ACCESSIBLE_PROCEDURES</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_ACTIVE_CONNECTIONS</td>
<td>0</td>
</tr>
<tr>
<td>SQL_ACTIVE_STATEMENTS</td>
<td>1</td>
</tr>
<tr>
<td>SQL_CONCAT_NULL_BEHAVIOR</td>
<td>SQL_CB_NULL</td>
</tr>
<tr>
<td>SQL_CONVERT_BIGINT</td>
<td>0</td>
</tr>
<tr>
<td>SQL_CONVERT_BINARY</td>
<td>0</td>
</tr>
<tr>
<td>SQL_CONVERT_BIT</td>
<td>0</td>
</tr>
<tr>
<td>SQL_CONVERT_CHAR</td>
<td>SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, SQL_CVT_TIME, SQL_CVT_DATE, SQL_CVT_FLOAT, or SQL_CVT_TIMESTAMP</td>
</tr>
<tr>
<td>SQL_CONVERT_DATE</td>
<td>SQL_CVT_CHAR, SQL_CVT_DATE</td>
</tr>
<tr>
<td>SQL_CONVERT_DECIMAL</td>
<td>SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT</td>
</tr>
<tr>
<td>SQL_CONVERT_DOUBLE</td>
<td>SQL_CVT_CHAR, SQL_CVT_FLOAT, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, or SQL_CVT_REAL</td>
</tr>
<tr>
<td>SQL_CONVERT_FLOAT</td>
<td>SQL_CVT_CHAR, SQL_CVT_FLOAT, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, or SQL_CVT_REAL</td>
</tr>
<tr>
<td>fInfoType</td>
<td>Return Values</td>
</tr>
<tr>
<td>---------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SQL_CONVERT_FUNCTIONS</td>
<td>SQL_FN_CVT_CONVERT</td>
</tr>
<tr>
<td>SQL_CONVERT_INTEGER</td>
<td>SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT</td>
</tr>
<tr>
<td>SQL_CONVERT_LONGVARBINARY</td>
<td>0</td>
</tr>
<tr>
<td>SQL_CONVERT_LONGVARCHAR</td>
<td>0</td>
</tr>
<tr>
<td>SQL_CONVERT_NUMERIC</td>
<td>SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT</td>
</tr>
<tr>
<td>SQL_CONVERT_REAL</td>
<td>SQL_CVT_CHAR, SQL_CVT_DECIMAL, SQL_CVT_FLOAT, SQL_CVT_INTEGER, or SQL_CVT_REAL</td>
</tr>
<tr>
<td>SQL_CONVERT_SMALLINT</td>
<td>SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT</td>
</tr>
<tr>
<td>SQL_CONVERT_TIME</td>
<td>SQL_CVT_CHAR or SQL_CVT_TIME</td>
</tr>
<tr>
<td>SQL_CONVERT_TIMESTAMP</td>
<td>SQL_CVT_CHAR, SQL_CVT_TIME, or SQL_CVT_DATE</td>
</tr>
<tr>
<td>SQL_CONVERT_TINYINT</td>
<td>SQL_CVT_CHAR, SQL_CVT_INTEGER, SQL_CVT_DECIMAL, SQL_CVT_REAL, or SQL_CVT_FLOAT</td>
</tr>
<tr>
<td>SQL_CONVERT_VARBINARY</td>
<td>0</td>
</tr>
<tr>
<td>SQL_CONVERT_VARCHAR</td>
<td>0</td>
</tr>
<tr>
<td>SQL_CORRELATION_NAME</td>
<td>SQL_CN_DIFFERENT</td>
</tr>
<tr>
<td>SQL_CURSOR_COMMIT_BEHAVIOR</td>
<td>SQL_CC_DELETE</td>
</tr>
</tbody>
</table>
Public's Notes for the Red Brick ODBC Driver
ODBC API Conformance

<table>
<thead>
<tr>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_CURSOR_ROLLBACK_BEHAVIOR</td>
<td>SQL_CR_DELETE</td>
</tr>
<tr>
<td>SQL_DATA_SOURCE_NAME</td>
<td>Supplied at login</td>
</tr>
<tr>
<td>SQL_DATA_SOURCE_READ_ONLY</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_DATABASE_NAME</td>
<td>Supplied at login</td>
</tr>
<tr>
<td>SQL_DBMS_NAME</td>
<td>&quot;Red Brick Warehouse&quot;</td>
</tr>
<tr>
<td>SQL_DBMS_VER</td>
<td>Current server version</td>
</tr>
<tr>
<td>SQL_DEFAULT_TXN_ISOLATION</td>
<td>0</td>
</tr>
<tr>
<td>SQL_DRIVER_HDBC</td>
<td>Handled by the ODBC Driver Manager</td>
</tr>
<tr>
<td>SQL_DRIVER_HENV</td>
<td>Handled by the ODBC Driver Manager</td>
</tr>
<tr>
<td>SQL_DRIVER_HSTMT</td>
<td>Handled by the ODBC Driver Manager</td>
</tr>
<tr>
<td>SQL_DRIVER_NAME</td>
<td>&quot;RB16ODBC.DLL&quot; or &quot;RB32ODBC.DLL&quot; or &quot;RBODBC&quot;</td>
</tr>
<tr>
<td>SQL_DRIVER_VER</td>
<td>Current driver version</td>
</tr>
<tr>
<td>SQL_EXPRESSIONS_IN_ORDERBY</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_FETCH_DIRECTION</td>
<td>SQL_FD_FETCH_NEXT</td>
</tr>
<tr>
<td>SQL_FILE_USAGE</td>
<td>SQL_FILE_NOT_SUPPORTED</td>
</tr>
<tr>
<td>SQL_IDENTIFIER_CASE</td>
<td>SQL_IC_UPPER</td>
</tr>
<tr>
<td>SQL_IDENTIFIER_QUOTE_CHAR</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>SQL_MAX_COLUMN_NAME_LEN</td>
<td>128</td>
</tr>
<tr>
<td>SQL_MAX_CURSOR_NAME_LEN</td>
<td>0</td>
</tr>
<tr>
<td>SQL_MAX.Owner_NAME_LEN</td>
<td>0</td>
</tr>
<tr>
<td>SQL_MAX_PROCEDURE_NAME_LEN</td>
<td>0</td>
</tr>
<tr>
<td>SQL_MAX_QUALIFIER_NAME_LEN</td>
<td>0</td>
</tr>
<tr>
<td>SQL_MAX_TABLE_NAME_LEN</td>
<td>128</td>
</tr>
<tr>
<td>SQL_MULT_RESULT_SETS</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_MULTIPLE_ACTIVE_TXN</td>
<td>&quot;Y&quot;</td>
</tr>
</tbody>
</table>
### Programmer’s Notes for the Red Brick ODBC Driver

**ODBC API Conformance**

<table>
<thead>
<tr>
<th>fInfoType</th>
<th>Return Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_NON_NULLABLE_COLUMNS</td>
<td>SQL_NNC_NON_NULL</td>
</tr>
<tr>
<td>SQL_NUMERIC_FUNCTIONS</td>
<td>SQL_FN_NUM_ABS, SQL_FN_NUM_CEILING, SQL_FN_NUM_FLOOR, or SQL_FN_NUM_SIGN</td>
</tr>
<tr>
<td>SQL_ODBC_API_CONFORMANCE</td>
<td>SQL_OAC_LEVEL1</td>
</tr>
<tr>
<td>SQL_ODBC_SAG_CLI_CONFORMANCE</td>
<td>SQL_OSCC_COMPLIANT</td>
</tr>
<tr>
<td>SQL_ODBC_SQL_CONFORMANCE</td>
<td>SQL_OSC_MINIMUM</td>
</tr>
<tr>
<td>SQL_ODBC_SQL_OPT_IEF</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_ODBC_VER</td>
<td>Handled by the ODBC Driver Manager</td>
</tr>
<tr>
<td>SQL_OUTER_JOINS</td>
<td>&quot;Y&quot;</td>
</tr>
<tr>
<td>SQL_OWNER_TERM</td>
<td>&quot;&quot;</td>
</tr>
<tr>
<td>SQL_PROCEDURE_TERM</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_PROCEDURES</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_QUALIFIER_NAME_SEPARATOR</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>SQL_QUALIFIER_TERM</td>
<td>&quot; &quot;</td>
</tr>
<tr>
<td>SQL_ROW_UPDATES</td>
<td>&quot;N&quot;</td>
</tr>
<tr>
<td>SQL_SEARCH_PATTERN_ESCAPE</td>
<td>&quot;\ &quot;</td>
</tr>
<tr>
<td>SQL_SERVER_NAME</td>
<td>Supplied at login</td>
</tr>
<tr>
<td>SQL_SCROLL_CONCURRENCY</td>
<td>SQL_SCCO_READ_ONLY</td>
</tr>
<tr>
<td>SQL_SCROLL_OPTIONS</td>
<td>SQL_SO_FORWARD_ONLY</td>
</tr>
<tr>
<td>SQL_STRING_FUNCTIONS</td>
<td>SQL_FN_STR_CONCAT, SQL_FN_STR_LCASE, SQL_FN_STR_LEFT, SQL_FN_STR_LTRIM,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_STR_RTRIM, SQL_FN_STR_SUBSTRING, or SQL_FN_STR_UCASE</td>
</tr>
<tr>
<td>SQL_SYSTEM_FUNCTIONS</td>
<td>SQL_FN_SYS_IFNULL or SQL_FN_SYS_USER</td>
</tr>
<tr>
<td>SQL_TABLE_TERM</td>
<td>&quot;Table&quot;</td>
</tr>
</tbody>
</table>
Programmer’s Notes for the Red Brick ODBC Driver

ODBC API Conformance

<table>
<thead>
<tr>
<th>fInfoType</th>
<th>Return Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_TIMEDATE_FUNCTIONS</td>
<td>SQL_FN_TD_CURDATE,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_CURTIME,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_DAYOFMONTH,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_DAYOFWEEK,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_DAYOFYEAR,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_HOUR,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_MINUTE,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_MONTH,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_NOW,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_QUARTER,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_SECOND,</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_WEEK,</td>
</tr>
<tr>
<td></td>
<td>or</td>
</tr>
<tr>
<td></td>
<td>SQL_FN_TD_YEAR</td>
</tr>
<tr>
<td>SQL_TXN_CAPABLE</td>
<td>SQL_TC_NONE</td>
</tr>
<tr>
<td>SQL_TXN_ISOLATION_OPTION</td>
<td>0</td>
</tr>
<tr>
<td>SQL_USER_NAME</td>
<td>Supplied at login</td>
</tr>
</tbody>
</table>
**SQLGetStmtOption, SQLSetStmtOption**

The following table lists the options supported by the Red Brick ODBC Driver for the SQLSetStmtOption and the SQLGetStmtOption. Default values are shown where applicable.

<table>
<thead>
<tr>
<th>fOption</th>
<th>Default Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>SQL_ASYNC_ENABLE</td>
<td>0 (FALSE)</td>
</tr>
<tr>
<td>SQL_BIND_TYPE</td>
<td>SQL_BIND_BY_COLUMN</td>
</tr>
<tr>
<td>SQL_MAX_LENGTH</td>
<td>0</td>
</tr>
<tr>
<td>SQL_MAX_ROWS</td>
<td>0</td>
</tr>
</tbody>
</table>

**SQLMoreResults**

Because Red Brick Warehouse does not support multiple result sets, the Red Brick ODBC Driver always returns SQL_NO_DATA_FOUND from the SQLMoreResults function. (The Red Brick ODBC Driver supports this function with this return value because it is required by Microsoft Access.)

**SQLPrepare**

During the prepare phase (SQLPrepare), the SQL statement submitted to the driver is prefixed by the PREPARE keyword. This statement is then executed for syntax/semantic checking. The original statement is stored and then executed during the execute phase (SQLExecute) after dynamic parameter substitution, if any.

**SQLTransact**

Because Red Brick Warehouse is designed for decision support, transactions are not supported and the driver is essentially always in AUTOCOMMIT mode. Therefore, SQL_COMMIT returns SQL_SUCCESS as if a commit were performed successfully; SQL_ROLLBACK returns SQL_ERROR and registers the following error:

S1C00 Driver not capable
SQL Conformance

The Red Brick ODBC Driver supports ODBC Core SQL grammar, with a few exceptions as listed in this section.

Minimum SQL Grammar

The Red Brick ODBC Driver supports the full Minimum SQL Grammar as documented in Appendix C of the Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide with the following exceptions:

<table>
<thead>
<tr>
<th>Minimum SQL Grammar</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE TABLE</td>
<td>IEF grammar CHECK() clause is not supported.</td>
</tr>
<tr>
<td>DROP TABLE</td>
<td>IEF grammar CASCADE</td>
</tr>
</tbody>
</table>

Core SQL Grammar

The Red Brick ODBC Driver supports the full Core SQL Grammar as documented in Appendix C of the Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide with the following exceptions:

<table>
<thead>
<tr>
<th>Core SQL Grammar</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ALTER TABLE</td>
<td>Red Brick SQL requires a DEFAULT clause when columns are added, which is not included in ODBC grammar.</td>
</tr>
<tr>
<td>CREATE INDEX</td>
<td>UNIQUE and ASC/DESC are not supported.</td>
</tr>
<tr>
<td>DELETE</td>
<td>Positioned deletes are not supported. (WHERE CURRENT OF cursor-name)</td>
</tr>
<tr>
<td>DROP VIEW</td>
<td>IEF grammar CASCADE</td>
</tr>
<tr>
<td>GRANT</td>
<td>IEF grammar REFERENCES() is not supported.</td>
</tr>
<tr>
<td>REVOKE</td>
<td>IEF grammar CASCADE</td>
</tr>
<tr>
<td>SELECT</td>
<td>FOR UPDATE OF is not supported.</td>
</tr>
<tr>
<td>UPDATE</td>
<td>Positioned updates are not supported. (WHERE CURRENT OF cursor-name)</td>
</tr>
</tbody>
</table>

For more information about Red Brick SQL grammar, refer to the SQL Reference Guide.
**Extended SQL Grammar**

The Red Brick ODBC Driver does not support Extended SQL Grammar, as documented in Appendix C of the *Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide*.

**SQL Extensions**

The Red Brick ODBC Driver supports the following SQL extensions in the form of vendor strings (escape clauses) using both standard and extended syntax. Because ODBC is a “pass-through” technology, SQL extensions are accepted by the driver, converted to the server’s native syntax, and then sent to the server where they are processed.

**Datetime Extensions**

The Red Brick ODBC Driver supports all of the datetime extensions. For syntax descriptions, refer to Appendix C of the *Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide*.

**Outer-Join Extensions**

The Red Brick ODBC Driver supports outer-join syntax. For more information about warehouse-supported SQL, refer to the *SQL Reference Guide*.

**Scalar Function Extensions**

The Red Brick ODBC Driver supports scalar function extensions, which can be included as primary expressions in SQL statements. For a description of the SQL syntax for these functions, refer to Appendix G of the *Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide*.

**Note:** Red Brick Warehouse macros can be stored in a warehouse database to simulate the standard ODBC syntax for scalar and datetime functions. For information about macros, refer to the *SQL Reference Guide*. 
**String Functions**

The Red Brick ODBC Driver supports the following ODBC string functions.

<table>
<thead>
<tr>
<th>ODBC Function</th>
<th>Red Brick Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONCAT(s1, s2)</td>
<td>CONCAT(s1, s2)</td>
</tr>
<tr>
<td>LCASE(s)</td>
<td>LOWER(s)</td>
</tr>
<tr>
<td>LEFT</td>
<td>SUBSTR(arg1, 1, arg2)</td>
</tr>
<tr>
<td>LTRIM(s)</td>
<td>LTRIM(s)</td>
</tr>
<tr>
<td>RTRIM(s)</td>
<td>RTRIM(s)</td>
</tr>
<tr>
<td>SUBSTRING(s, st, ln)</td>
<td>SUBSTR(s, st, ln)</td>
</tr>
<tr>
<td>UCASE(s)</td>
<td>UPPER(s)</td>
</tr>
</tbody>
</table>

For information about additional string functions supported by Red Brick Warehouse, refer to the *SQL Reference Guide*.

**Numeric Functions**

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

<table>
<thead>
<tr>
<th>ODBC Function</th>
<th>Red Brick Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABS(n)</td>
<td>ABS(n)</td>
</tr>
<tr>
<td>CEILING(n)</td>
<td>CEIL(n)</td>
</tr>
<tr>
<td>FLOOR(n)</td>
<td>FLOOR(n)</td>
</tr>
<tr>
<td>SIGN(n)</td>
<td>SIGN(n)</td>
</tr>
</tbody>
</table>

For information about additional numeric functions supported by Red Brick Warehouse, refer to the *SQL Reference Guide*. 
### Date Functions

The Red Brick ODBC Driver supports the following datetime functions.

<table>
<thead>
<tr>
<th>ODBC Function</th>
<th>Red Brick Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CURDATE</td>
<td>CURRENT_DATE</td>
</tr>
<tr>
<td>CURTIME</td>
<td>CURRENT_TIME</td>
</tr>
<tr>
<td>DAYOFMONTH</td>
<td>EXTRACT(DAY FROM datetime_expression)</td>
</tr>
<tr>
<td>DAYOFWEEK</td>
<td>EXTRACT(WEEKDAY FROM datetime_expression)</td>
</tr>
<tr>
<td>DAYOFYEAR</td>
<td>EXTRACT(DAYOFYEAR FROM datetime_expression)</td>
</tr>
<tr>
<td>HOUR</td>
<td>EXTRACT(HOUR FROM datetime_expression)</td>
</tr>
<tr>
<td>MINUTE</td>
<td>EXTRACT(MINUTE FROM datetime_expression)</td>
</tr>
<tr>
<td>MONTH</td>
<td>EXTRACT(MONTH FROM datetime_expression)</td>
</tr>
<tr>
<td>NOW</td>
<td>CURRENT_TIMESTAMP(6)</td>
</tr>
<tr>
<td>QUARTER</td>
<td>EXTRACT(QUARTER FROM datetime_expression)</td>
</tr>
<tr>
<td>SECOND</td>
<td>EXTRACT(SECOND FROM datetime_expression)</td>
</tr>
<tr>
<td>WEEK</td>
<td>EXTRACT(WEEK FROM datetime_expression)</td>
</tr>
<tr>
<td>YEAR</td>
<td>EXTRACT(YEAR FROM datetime_expression)</td>
</tr>
</tbody>
</table>

For more information about Red Brick datetime scalar functions and the `datetime_expression` variables listed in this table, refer to the SQL Reference Guide.

### System Functions

The Red Brick ODBC Driver supports the following system functions.

<table>
<thead>
<tr>
<th>ODBC Function</th>
<th>Red Brick Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>IFNULL(exp, val)</td>
<td>IFNULL(exp, val)</td>
</tr>
<tr>
<td>USER( )</td>
<td>CURRENT_USER</td>
</tr>
</tbody>
</table>

For more information about system functions supported by Red Brick Warehouse, refer to the SQL Reference Guide.
Conversion Functions

The Red Brick ODBC Driver supports the following conversion functions.

<table>
<thead>
<tr>
<th>ODBC Function and Type</th>
<th>Red Brick Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONVERT(value, SQL_CHAR)</td>
<td>STRING(value)</td>
</tr>
<tr>
<td>CONVERT(value, SQL_DATE)</td>
<td>DATE(value)</td>
</tr>
<tr>
<td>CONVERT(value, SQL_DECIMAL)</td>
<td>DEC(expression)</td>
</tr>
<tr>
<td>CONVERT(value, SQL_FLOAT)</td>
<td>FLOAT(value)</td>
</tr>
<tr>
<td>CONVERT(value, SQL_INTEGER)</td>
<td>INT(value)</td>
</tr>
<tr>
<td>CONVERT(value, SQL_REAL)</td>
<td>REAL(value)</td>
</tr>
<tr>
<td>CONVERT(value, SQL_TIME)</td>
<td>TIME(value)</td>
</tr>
<tr>
<td>CONVERT(value, SQL_TIMESTAMP)</td>
<td>TIMESTAMP(value)</td>
</tr>
</tbody>
</table>

For more information about conversion functions supported by Red Brick Warehouse, refer to the SQL Reference Guide.
**ODBC Datatypes**

This table shows datatype mapping that the Red Brick ODBC Driver performs between Red Brick and ODBC datatypes. If no other datatype conversion is specified by the user when SQLGetData or SQLBindCol is called, then the Red Brick ODBC Driver converts Red Brick datatypes to the indicated ODBC type. For more information about ODBC datatype conversions, refer to Appendix D of the *Microsoft ODBC 2.0 Programmer’s Reference and SDK Guide*.

<table>
<thead>
<tr>
<th>Warehouse Datatype (Name used in a CREATE TABLE statement to create a specified Red Brick datatype)</th>
<th>ODBC SQL Datatype</th>
<th>Default ODBC C Datatype (ODBC datatype to which the Red Brick datatype is logically mapped)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CHAR</td>
<td>SQL_CHAR</td>
<td>SQL_C_CHAR</td>
</tr>
<tr>
<td>TINYINT</td>
<td>SQL_TINYINT</td>
<td>SQL_C_STINYINT</td>
</tr>
<tr>
<td>SMALLINT</td>
<td>SQL_SMALLINT</td>
<td>SQL_C_SSHORT</td>
</tr>
<tr>
<td>INTEGER</td>
<td>SQL_INTEGER</td>
<td>SQL_C_SLONG</td>
</tr>
<tr>
<td>NUMERIC, DECIMAL</td>
<td>SQL_DECIMAL</td>
<td>SQL_C_CHAR</td>
</tr>
<tr>
<td>REAL</td>
<td>SQL_REAL</td>
<td>SQL_C_FLOAT</td>
</tr>
<tr>
<td>DOUBLE, FLOAT</td>
<td>SQL_DOUBLE</td>
<td>SQL_C_DOUBLE</td>
</tr>
<tr>
<td>DATE</td>
<td>SQL_DATE</td>
<td>SQL_C_DATE</td>
</tr>
<tr>
<td>TIME</td>
<td>SQL_TIME</td>
<td>SQL_C_TIME</td>
</tr>
<tr>
<td>TIMESTAMP</td>
<td>SQL_TIMESTAMP</td>
<td>SQL_C_TIMESTAMP</td>
</tr>
</tbody>
</table>

The SQL_NUMERIC and SQL_FLOAT ODBC SQL datatypes are also supported.

**Stack Size Requirements**

The Red Brick ODBC Driver shares stack space with the calling application and requires a minimum of 8 kilobytes of available space from the application.
Programmer’s Notes for the Red Brick ODBC Driver

Stack Size Requirements
Creating and Using a Custom ODBC Program

This chapter contains information about using a software development kit (SDK) to build a custom ODBC-based client application to access Red Brick Warehouse; such client applications can be created to run on either Microsoft Windows or UNIX-based systems.

- To build a Windows-based application, you should purchase and use the Microsoft ODBC SDK.
- To build a UNIX-based application, you should use the Red Brick ODBClib SDK, which is available on all supported UNIX platforms.

For information about creating and using an ODBC application under Microsoft Windows, refer to the documentation for the Microsoft ODBC SDK.

This chapter contains the information you need to compile and link a UNIX-based custom application, as well as information about running an ODBC application on UNIX platforms. This information is divided into the following main sections:

- Using the Red Brick ODBClib Interface
- Running an ODBC Application on UNIX

For information on the sample ODBC program shipped with the Red Brick ODBClib SDK on UNIX platforms, refer to Chapter 4, “Using the Sample ODBC Program.”
Creating and Using a Custom ODBC Program
Using the Red Brick ODBClib Interface

Using the Red Brick ODBClib Interface

This section provides the following information needed to compile and link a custom ODBC program using the Red Brick ODBClib interface:

- Supported C Compilers
- ODBClib Libraries and Header Files
- Compiling and Linking with Red Brick ODBClib

The Red Brick ODBClib SDK contains libraries to link with an ODBC program. The level of ODBC conformance is described in Chapter 2, “Programmer’s Notes for the Red Brick ODBC Driver.”

For additional information about developing applications to interface with the Red Brick ODBC Driver, refer to the ODBC 2.0 Programmer’s Reference and SDK Guide available from Microsoft Corporation.

Supported C Compilers

The following C compilers are supported:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Compilers</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM RISC System/6000</td>
<td>AIX Compiler (XL C Compiler V1.3.0.19)</td>
</tr>
<tr>
<td>HP 9000</td>
<td>HP-UX C Compiler</td>
</tr>
<tr>
<td>Sun SPARC-based System</td>
<td>Sun Compiler (SC V3.0.1) gcc V2.7</td>
</tr>
<tr>
<td>Digital AlphaServer</td>
<td>Digital UNIX C Compiler</td>
</tr>
<tr>
<td>Silicon Graphics IRIX</td>
<td>MIPS Compiler</td>
</tr>
<tr>
<td>Sequent Symmetry DYNIX/ptx</td>
<td>PTX ANSI C Compiler</td>
</tr>
<tr>
<td>NCR UNIX SVR4 MP-RAS</td>
<td>High Performance C compiler</td>
</tr>
<tr>
<td>Unisys System V</td>
<td>C compiler</td>
</tr>
</tbody>
</table>
**ODBClib Libraries and Header Files**

The following libraries and header files are installed in the `redbrick_dir` directory, where `redbrick_dir` is the directory in which the Red Brick software is installed:

```
redbrick_dir
  |-- lib
     |-- librbodbc.a (Red Brick ODBClib static library)
     |-- librbodbc.so (Red Brick ODBClib shared library)
  |-- include
     |-- rbsql.h (Red Brick header file with all ODBC functions)
     |-- rbsqlext.h (Red Brick header file)
```

**Note:** The Red Brick ODBClib shared library is not available on the Sequent DYNIX/ptx platform.

**Compiling and Linking with Red Brick ODBClib**

This section provides information about the libraries and header files you will need to build an application with Red Brick ODBClib.

**Note:** The advantages to linking with shared libraries are a smaller executable, faster linking, and the ability to use a newer version of the shared library without re-linking the application. Executables linked with static libraries, however, might start up more quickly; furthermore, the static library does not have to be available at run time.
Platform-Specific Libraries

Platform-specific standard libraries that should be linked with the application are as follows:

<table>
<thead>
<tr>
<th>Platform</th>
<th>Libraries</th>
</tr>
</thead>
<tbody>
<tr>
<td>IBM RISC System/6000</td>
<td>libm.a</td>
</tr>
<tr>
<td>HP 9000</td>
<td>libm.a</td>
</tr>
<tr>
<td>Sun SPARC-based System</td>
<td>libsocket.a</td>
</tr>
<tr>
<td></td>
<td>libnsl.a</td>
</tr>
<tr>
<td></td>
<td>libm.a</td>
</tr>
<tr>
<td>Digital AlphaServer</td>
<td>libpthread.a</td>
</tr>
<tr>
<td></td>
<td>libm.a</td>
</tr>
<tr>
<td>NCR UNIX SVR4 MP-RAS</td>
<td>libsocket.a</td>
</tr>
<tr>
<td></td>
<td>libnsl.a</td>
</tr>
<tr>
<td></td>
<td>libm.a</td>
</tr>
<tr>
<td>Sequent Symmetry DYNIX/ptx</td>
<td>libmalloc.a</td>
</tr>
<tr>
<td></td>
<td>libsec.a</td>
</tr>
<tr>
<td></td>
<td>libseq.a</td>
</tr>
<tr>
<td></td>
<td>libsocket.a</td>
</tr>
<tr>
<td></td>
<td>libinet.a</td>
</tr>
<tr>
<td></td>
<td>libnsl.a</td>
</tr>
<tr>
<td></td>
<td>libm.a</td>
</tr>
<tr>
<td>Silicon Graphics IRIX</td>
<td>libm.a</td>
</tr>
<tr>
<td>Unisys System V</td>
<td>libsocket.a</td>
</tr>
<tr>
<td></td>
<td>libnsl.a</td>
</tr>
<tr>
<td></td>
<td>libm.a</td>
</tr>
</tbody>
</table>

Compile and Link Lines

This section lists sample commands for each platform to compile and link a file with the Red Brick ODBC static libraries and header files. The 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'
```
IBM RISC System/6000—AIX

The following definitions apply to the compile and link commands for the AIX operating system:

```bash
# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
CFLAGS=

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

Use commands in the following form to compile and link an application on the AIX operating system:

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

HP 9000—HP-UX

The following definitions apply to the compile and link commands for the HP-UX operating system:

```bash
# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
CFLAGS=-Aa +z

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

Use commands in the following form to compile and link an application on the HP-UX operating system:

```bash
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```
Sun SPARC-Based System—Solaris

The following definitions apply to the compile and link commands for the Solaris operating system:

```bash
# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lsocket -lnsl -lm
CFLAGS=

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

Use commands in the following form to compile and link an application on the Solaris operating system:

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

Digital AlphaServer—Digital UNIX

The following definitions apply to the compile and link commands for the Digital UNIX operating system:

```bash
# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lpthreads -lm
CFLAGS=

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

Use commands in the following form to compile and link an application on the Digital UNIX operating system:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```
Creating and Using a Custom ODBC Program
Using the Red Brick ODBClib Interface

NCR WorldMark Server—NCR UNIX SVR4 MP-RAS

The following definitions apply to the compile and link commands for the NCR UNIX SVR4 MP-RAS operating system:

```
# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lsocket -lnsl -lm \
    -L/usr/ucblib -lucb
CFLAGS=

# If you want to use Red Brick ODBClib shared library
# define LIBS as follows
#LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lsocket -lnsl -lm \
#    -L/usr/ucblib -lucb
```

Use commands in the following form to compile and link an application on the NCR UNIX SVR4 MP-RAS operating system:

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

Sequent Symmetry System—DYNIX/ptx

The following definitions apply to the compile and link commands for the DYNIX/ptx operating system:

```
# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lmalloc -lsec -lsocket \ 
    -linet -lnsl -lm
CFLAGS=

# Red Brick ODBClib shared library not supported on DYNIX/ptx
```

Use commands in the following form to compile and link an application on the DYNIX/ptx operating system:

```
$ cc $(CFLAGS) -I$(REDBRICK_DIR)/include -c rb_client.c
$ cc $(CFLAGS) -o rb_client rb_client.o $(LIBS)
```
Creating and Using a Custom ODBC Program
Using the Red Brick ODBClib Interface

Silicon Graphics Server—IRIX

The following definitions apply to the compile and link commands for the IRIX operating system:

```
# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lm
# Define CFLAGS for 64-bit compilation
CFLAGS=-mips3

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

Use commands in the following form to compile and link an application on the IRIX operating system:

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

Unisys U6000—Unisys UNIX System V

The following definitions apply to the compile and link commands for the Unisys UNIX System V operating system:

```
# Define libraries to link
LIBS=$(REDBRICK_DIR)/lib/librbodbc.a -lsocket -lnsl -lm -L/usr/ucblib -lucb
CFLAGS=

# If you want to use Red Brick ODBClib shared library
# define LIBS as follows
# LIBS=$(REDBRICK_DIR)/lib/librbodbc.so -lsocket -lnsl -lm -L/usr/ucblib -lucb
```

Use commands in the following form to compile and link an application on the Unisys UNIX System V operating system:

```
$ 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

To run an ODBC application on a UNIX platform where the ODBC application connects to a Red Brick Warehouse database, you must have your UNIX environment configured such that the RB_CONFIG environment variable is set, a valid $HOME/odbc.ini file exists in each user’s home directory, and the shared library environment variable specific to your hardware platform is set correctly.

Setting the RB_CONFIG Environment Variable

You must set the RB_CONFIG environment variable before you run an ODBC application on UNIX. This allows the application to locate the rbw.config file, which is used to locate the message files and to determine the LOCALE setting. Set the RB_CONFIG environment variable as in the following C shell example:

```bash
setenv RB_CONFIG <redbrick_dir>
```

where `redbrick_dir` is the directory in which the Red Brick ODBClib SDK is installed.

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

You can configure data source names (DSNs) on UNIX in the $HOME/odbc.ini file in each user’s home directory. It is often convenient for an administrator to place in each user’s home directory a .odbc.ini file that is a symbolic link to a central .odbc.ini file. This allows simpler administration by centralizing the DSN definitions, rather than repeating them for each user.

You must have a .odbc.ini file (or symbolic link) in each user’s home directory ($HOME). Otherwise you will not be able to connect to a Red Brick Warehouse database unless the connection string in your application specifically lists all of the components of the DSN specification.

Note: If your $HOME directory is shared across multiple hardware platforms, your .odbc.ini file might be platform-specific. This is because the .odbc.ini file includes the InstallDir specification that specifies the location of the Red Brick ODBClib SDK, and this location may be a different directory path on different platforms in your environment.
Initialization File for Applications That Use the Red Brick ODBClib SDK

The following is a sample .odbc.ini file for an application that uses the 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 defines the ODBC environment
#
[ODBC]
InstallDir=<redbrick_dir>/lib

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

# 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)>

Note: This sample .odbc.ini file defines a DSN with the name RBDSN.
Configuring the Environment for Shared Libraries

If you create an application that uses the Red Brick ODBClib shared library, you must configure the UNIX environment in which the program runs such that it can locate the shared library. The following sections show the environment variable to set for each platform.

Note: The Red Brick ODBClib shared library is not available on the Sequent DYNIX/ptx platform.

**IBM RISC System/6000—AIX**

On AIX, set the LIBPATH environment variable for the Red Brick ODBClib shared library as in the following C shell example:

```
setenv LIBPATH <redbrick_dir>/lib:$LIB_PATH
```

**HP 9000—HP-UX**

On HP-UX, set the SHLIB_PATH environment variable for the Red Brick ODBClib shared library as in the following C shell example:

```
setenv SHLIB_PATH <redbrick_dir>/lib:$SHLIB_PATH
```

**Sun SPARC-Based System—Solaris**

On Solaris, set the LD_LIBRARY_PATH environment variable for the Red Brick ODBClib shared library as in the following C shell example:

```
setenv LD_LIBRARY_PATH <redbrick_dir>/lib:$LD_LIBRARY_PATH
```

**Digital AlphaServer—Digital UNIX**

On Digital UNIX, set the LD_LIBRARY_PATH environment variable for the Red Brick ODBClib shared library as in the following C shell example:

```
setenv LD_LIBRARY_PATH <redbrick_dir>/lib:$LD_LIBRARY_PATH
```

**NCR WorldMark Server—NCR UNIX SVR4 MP-RAS**

On NCR UNIX SVR4 MP-RAS, set the LD_LIBRARY_PATH environment variable for the Red Brick ODBClib shared library as in the following C shell example:

```
setenv LD_LIBRARY_PATH <redbrick_dir>/lib:$LD_LIBRARY_PATH
```
Creating and Using a Custom ODBC Program
Running an ODBC Application on UNIX

Silicon Graphics Server—IRIX

On IRIX, set the LD_LIBRARY_PATH environment variable for the Red Brick ODBClib shared library as in the following C shell example:

```
setenv LD_LIBRARY_PATH <redbrick_dir>/lib:$LD_LIBRARY_PATH
```

Unisys U6000—Unisys UNIX System V

On Unisys UNIX System V, set the LD_LIBRARY_PATH environment variable for the Red Brick ODBClib shared library as in the following C shell example:

```
setenv LD_LIBRARY_PATH <redbrick_dir>/lib:$LD_LIBRARY_PATH
```
This chapter contains the information you need to compile and link the sample ODBC C program shipped with the Red Brick ODBClib SDK on UNIX platforms. This information is divided into the following sections:

• Sample C Program
• Setting Up the Sample Program
Using the Sample ODBC Program

Sample C Program

A sample ODBC C program is shipped as part of the Red Brick ODBClib SDK. The source code for the program is installed in the $RB_CONFIG/lib/example directory, where $RB_CONFIG is the directory in which the Red Brick ODBClib SDK is 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 application is built.

Setting Up the Sample Program

The sample program, named rb_client, is a simple ODBC client application that submits SQL statements to a Red Brick Warehouse database. To build this application, use an editor such as vi to modify the Makefile for your platform-specific information and then compile and link it.

Defining the Platform

In section 1 of the Makefile, find the RB_PLATFORM variable corresponding to your platform and un-comment the appropriate one. For example, if your platform is Solaris, the following line should appear in the Makefile:

```
RB_PLATFORM=solaris
```

Defining the Red Brick ODBClib SDK

In section 2 of the Makefile, make sure the following line is present to define to the application that the Red Brick ODBClib SDK is being used:

```
RB_ODBC=1
```
Defining the Installation Directory

In section 3 of the Makefile, find the RB_INSTALL_DIR variable and define it to reference the full directory path of the directory in which the Red Brick ODBClib SDK is installed. For example, if you have installed the SDK in a directory named /redbrick/odbc, the following line should appear in your Makefile:

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

Specifying the Pre-Processor, Archiver, and Compiler

The Makefile included with the sample program allows you to specify the path to the executables for a C pre-processor, 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 pre-processor, object archiver, and C compiler, respectively.

Building the rb_client Application

After you have edited the Makefile with all of your platform- and environment-specific information, build the rb_client application by entering the following command from the $RB_CONFIG/lib/example directory, where $RB_CONFIG is the directory in which the Red Brick ODBClib SDK is installed:

```
% make
```

This builds an executable named rb_client. The rb_client program asks the user to enter a data source name (DSN) and then a SQL statement. The SQL statement is then executed on the Red Brick Warehouse database corresponding to the DSN specified.

Note: The DSN must be defined in the $HOME/.odbc.ini file and the RB_CONFIG environment variable must be set.
Using the Sample ODBC Program
Setting Up the Sample Program
Symbols
.odbc.ini file
   configuring data sources with 3-9
   sample 3-10

C
C compilers, supported 3-2
cases, tracked by technical support xi
compiling and linking 3-3
connectivity architecture 1-2
conventions
   syntax diagrams x
   syntax notation ix
conversion functions 2-16
Customer Support Center x

D
data source names (DSNs), configuring
   with .odbc.ini file 3-9
datatypes, ODBC 2-17
date functions 2-15
datetime extensions 2-13
documentation
   list of Red Brick Systems vii
   support xii

E
e-mail addresses, for Red Brick Systems x

L
LD_LIBRARY_PATH environment variable 3-11
LIBPATH environment variable 3-11
libraries, platform-specific 3-4

N
notation conventions ix
numeric functions 2-14

O
ODBC API conformance 2-3
ODBC application, running on UNIX 3-9
ODBC datatypes 2-17
   .odbc.ini file
      configuring data sources with 3-9
      sample 3-10
outer-join extensions 2-13

R
rb_client, building application with
   make 4-3
RB_CONFIG environment variable 3-9
Red Brick ODBC Driver
   requirements 2-2
Red Brick ODBClib
   compiling and linking with 3-3
   directory structure 3-3
Red Brick Warehouse
   connectivity architecture 1-2
   introduction 1-1

S
sample ODBC program, setting up 4-2
scalar function 2-13
shared libraries 3-11
SHLIB_PATH environment variable 3-11
SQL conformance 2-12
SQL extensions 2-13
SQL grammar 2-12
SQLAllocConnect function 2-3
SQLAllocEnv function 2-3
Index

SQLAllocStmt function 2-3
SQLBindCol function 2-3
SQLBindParameter function 2-3
SQLCancel function 2-3
SQLColAttributes function 2-3
SQLColumns function 2-3
SQLConnect function 2-3
SQLDataSources function, conformance level 2-3
SQLDescribeCol function 2-3
SQLDisconnect function 2-3
SQLDriverConnect function 2-3
SQLDrivers function, conformance level 2-3
SQLError function 2-3, 2-4
SQLExecDirect function 2-3
SQLExecute function 2-3, 2-5
SQLFetch function 2-3, 2-5
SQLFreeConnect function 2-3
SQLFreeEnv function 2-3
SQLFreeStmt function 2-3
SQLGetConnectOption function 2-3, 2-5
SQLGetCursorName function 2-3, 2-5
SQLGetData function 2-3
SQLGetFunctions function 2-4
SQLGetInfo function
   conformance level 2-4
   values returned 2-6
SQLGetStmtOption function 2-4, 2-11
SQLGetTypeInfo function 2-4
SQLMoreResults function 2-4, 2-11
SQLNativeSql function 2-4
SQLNumResultCols function 2-4
SQLParamData function 2-4
SQLPrepare function 2-4, 2-11
SQLPutData function 2-4
SQLRowCount function 2-4
SQLSetConnectOption function 2-4, 2-5
SQLSetCursorName function 2-4, 2-5
SQLSetParam function 2-4
SQLSetStmtOption function 2-4, 2-11
SQLSpecialColumns function 2-4
SQLStatistics function 2-4
SQLTables function 2-4
SQLTransact function 2-4, 2-11
stack size requirements 2-17
string functions 2-14
support
documentation xii
   technical x
syntax diagrams, conventions for x
syntax notation ix
system functions 2-15

T
technical support x
troubleshooting, general problems xii

W
World Wide Web address, for Red Brick Systems x
USA SALES OFFICES
1040 Crowne Point Parkway, Suite 250, Atlanta, GA 30338 +1 770 804 2440
2215 York Road, Suite 409, Oak Brook, IL 60521 +1 630 472 8500
1120 Avenue of the Americas, 4th Floor, New York, NY 10036 +1 212 626 6815
5314 Arapahoe Road, Dallas, TX 75248 +1 972 702 1750
2010 Corporate Ridge, 7th Floor, McLean, VA 22102 +1 703 883 9310

UK SALES OFFICE
Red Brick Systems UK Ltd., 45 Berkeley Square, Mayfair, London W1A 1EB
United Kingdom +44 171 290 8373

AUSTRALASIA HEADQUARTERS
Red Brick Systems Australasia Pty. Ltd., Level 20, 99 Walker Street,
North Sydney, NSW 2060 Australia +61 02 9911 7744

JAPAN HEADQUARTERS
Red Brick Japan Co. Ltd., Level 16 Shiroyama Hills, 4-3-1 Toranomon,
Minato-ku, Tokyo 105 Japan +81 3 5403 4638