
Versant/ODBC Reference Manual

Release 7.0.1.4



Versant History, Innovating for Excellence

In 1988, Versant's visionaries began building solutions based on a highly scalable and distributed object-oriented architecture and a patented caching algorithm that proved to be prescient.

Versant's initial flagship product, the Versant Object Database Management System (ODBMS), was viewed by the industry as the one truly enterprise-scalable object database.

Leading telecommunications, financial services, defense and transportation companies have all depended on Versant to solve some of the most complex data management applications in the world. Applications such as fraud detection, risk analysis, simulation, yield management and real-time data collection and analysis have benefited from Versant's unique object-oriented architecture.

For more Information please visit www.versant.com

Versant US

Versant Corporation

255 Shoreline Drive, Suite 450, Redwood City, CA 94065

Ph +1 650-232-2400, Fx +1 650-232-2401

Versant Europe

Versant GmbH

Wiesenkamp 22b, 22359 Hamburg, Germany

Ph +49.40.60990-0, Fx +49.40.60990-113

© 2008 Versant Corporation.

All products are trademarks or registered trademarks of their respective companies in the United States and other countries.

The information contained in this document is a summary only.

For more information about Versant Corporation and its products and services, please contact Versant Worldwide or European Headquarters.

Table of Contents

CHAPTER 1: Introduction To VODBC	3
Product Overview	4
Third-Party Tools	5
Conformance Notes	6
Handling of User Name and Password	6
 CHAPTER 2: Data Sources Configuration	 7
Configuring Data Sources With the ODBC Administrator Utility.....	8
Format of ODBC.INI, the Driver Manager Initialization File	9
Sample ODBC.INI File With Versant ReVind ODBC Data Source.....	10
Configuring Data Sources on Windows 2000	10
 CHAPTER 3: Conformance Notes	 13
ODBC and SQL Support Levels.....	14
Information the ODBC Driver Returns to SQLGetInfo.....	14
Supported ODBC API Functions.....	29
Versant ReVind Supported Data Types.....	30
Versant ReVind Supported Isolation and Lock Levels	31
Index.....	33

Introduction To VODBC

This Chapter gives a detailed overview of Versant/ODBC.

The Chapter explains the following:

- Product Overview
- Third - Party Tools
- Conformance Notes
- Handling of User Name and Password

Product Overview

Versant/ODBC provides access to Versant ReVind(Versant ReVind) environments from desktop tools and applications that support the ODBC (Open Database Connectivity) interface.

The Open Database Connectivity (ODBC) interface from Microsoft has emerged as the standard mechanism for client applications to access data from a variety of different sources through a single interface. Users of applications that support ODBC merely select a new database from a point-and-click menu to connect, transparently, to data from that data source.

To become accessible from ODBC client applications, database environments must provide software, called a driver, on the client system where the application resides. The driver translates the standard ODBC function calls into calls the data source can process, and returns data to the application. Each data source provides a driver on the client system for applications to use to access data from that source.

The Versant ReVind(VSQL) ODBC Driver extends this plug-and-play interoperability to any proprietary storage system realized as a Versant ReVind environment. It allows any Microsoft-Windows tool or application that supports the ODBC call library to easily use a Versant ReVind environment as a data source. With it, applications based on tools such as Power Builder and Visual Basic can include any Versant ReVind environment as a data source.

The ODBC interface specifies two major components:

- A library of function calls that allow applications to connect with a database system and issue statements through an application programming interface (API).
- Syntax for Structured Query Language (SQL) statements, based on existing standards.

ODBC drivers fit in as a layer of "middleware" in the ODBC architecture. The ODBC architecture includes the following layers:

Application

An ODBC application is any program that calls ODBC functions and uses them to issue SQL statements. For example, many vendors have added ODBC support to their existing Windows-based tools, such as Power Builder and Impromptu, so those tools can use ODBC for data access.

ODBC Driver Manager

A Microsoft-supplied dynamic-link library (DLL), that routes calls from an application to the appropriate ODBC driver for a data source. To an application, the ODBC driver manager and a driver are a single entity that process requests to a particular data source. The ODBC driver manager loads the requested driver in response to an application's call to the ODBC `SQLConnect` or `SQLDriverConnect` functions.

ODBC Driver

A dynamic link library (DLL) with ODBC function calls for a specific data source. The driver connects to the data source, translates the standard SQL statements into syntax the data source can process, and returns data to the application. There are ODBC drivers for every major database system.

Data Source

The combination of a database system, the operating system it uses, and any network software required to access it. (ODBC defines a database system (DBMS) as any vendor's implementation of a data access system that provides an SQL interface.)

Third-Party Tools

The following tools have been tested with the Versant ReVind ODBC Driver:

Vendor	Tool	Version	Platform
Microsoft	Access	97, 2000	Windows
Microsoft	Access	7.0	Windows 2000
Microsoft	Query	1.0	Windows
Microsoft	Excel	5.0	Windows
Microsoft	Visual Basic	5.0	Windows
Microsoft	Word	97	Windows
Microsoft	ODBC Test	3.0	Windows
Microsoft	ODBC Test	3.0	Windows 2000
Crystal Services	Crystal Reports	3.0	Windows

Microsoft Access Notes

- The ODBC Driver does not support the use of the trace flag in Microsoft Access. It's use causes a memory fault in Access. This problem is being investigated.
- Microsoft Access does not correctly handle the `TIME` data type. It requests that data sources return columns of type `TIME` as timestamps, but requires that the date portion have a nonzero value. As a workaround, a default date is always specified in the time to timestamp conversion.

Conformance Notes

ODBC specifies general conformance levels in two areas:

- ODBC application programming interface (API)
- ODBC SQL syntax

The Versant ODBC driver included with this release of Versant/ODBC supports Version 3.0 of ODBC. It supports:

- All Core and Level 1 ODBC API functions, and some Level 2 functions
- Extended SQL syntax

Handling of User Name and Password

When an ODBC application issues a `SQLConnect` call, the Versant ReVind ODBC driver passes any user name or password information to the host system. On the host system, Versant ReVind validates the user name against the password.

The ODBC driver gets the user name and password information as follows:

1. First by reading the input arguments passed from the application's call to the `SQLConnect` or `SQLDriverConnect` functions.
2. If no values are supplied to the function calls, the driver uses the values stored in `ODBC.INI` for the data source.

The driver passes any user name and password obtained from these sources to the server. If none of these sources provide the user name and password, the driver passes empty quoted strings. If the Versant ReVind data source returns an authentication error, the driver displays a dialog box for the user of the client application to supply information.

Data Sources Configuration

This Chapter describes how to add, modify and delete Versant ReVind ODBC data sources on Windows client systems. It describes the usage of ODBC Administrator to perform these operations and also format of the initialization file maintained by the ODBC Administrator.

The Chapter covers the following in detail:

- Configuring Data Sources With the ODBC Administrator Utility
- Format of ODBC.INI, the Driver Manager Initialization file
- Configuring Data Sources on Windows 2000

Configuring Data Sources With the ODBC Administrator Utility

The ODBC Administrator is a Microsoft utility to configure ODBC data sources and drivers.

Invoking the ODBC Administrator Utility

To configure any ODBC data source, invoke the ODBC Administrator Utility (The Versant ReVind ODBC driver installation procedure installs the ODBC Driver Manager software and the associated ODBC Administrator utility if it was not already installed):

1. Open the Control Panel program item in the Program Manager.
2. Double click the ODBC icon in the Windows Control Panel. The ODBC Administrator displays the Data Sources dialog box.

Adding Versant ReVind ODBC Data Sources

Once you invoke the ODBC Administrator:

1. In the Data Sources dialog box, choose the Add button. The Add Data Source dialog box appears.
2. Select VERSANT from the list of installed drivers and choose OK. The Versant/ODBC Setup dialog box appears.
3. Fill in the dialog box fields and choose OK. The Administrator writes the values you supply to `ODBC.INI`.
4. The Data Source Dialog box reappears, and now includes the newly-added data source. Choose Close.

The ODBC driver passes the user name and password you specify to Versant ReVind on the host system. Versant ReVind verifies the user name against the corresponding password before it connects to the database. If you omit this information, the driver prompts users for it when they try to connect to the data source.

Modifying and Deleting Versant ReVind Data Sources

You can modify or delete a Versant ReVind data source after you add it.

Invoke the ODBC Administrator and select the Versant ReVind data source you want to modify or delete. Then choose the `Setup` or `Delete` button.

`Setup`

Choosing the `Setup` button displays the Versant ODBC Setup dialog box with the current values for that data source. Change the values of any fields, then choose OK.

When you modify a data source, the ODBC Administrator modifies the entry for the data source name you specify. For example you could modify a Versant ReVind data source to change the user name and password that connections use.

Delete

Choosing the `Delete` button displays a confirmation box. Choose OK if you want to delete the data source.

When you delete a data source, the ODBC Administrator deletes the entry for that data source in `ODBC.INI`. (Deleting a data source has no effect on any database., only on `ODBC.INI`.)

Format of ODBC.INI, the Driver Manager Initialization File

The ODBC Administrator is a Microsoft utility to configure ODBC data sources and drivers. The Administrator maintains two text files.

\WINDOWS\ODBC.INI

`\WINDOWS\ODBC.INI` details all the ODBC data sources an application can access from a system, and specifies which driver ODBC should use to access each data source. The ODBC driver manager software reads `ODBC.INI` to determine the location of driver DLLs and attributes of each data source.

\WINDOWS\ODBCINST.INI

`\WINDOWS\ODBCINST.INI` details all the drivers that have been installed on a system. The ODBC Administrator reads `ODBCINST.INI` when it invokes a driver's setup DLL to add or modify a data source for the driver.

When you add a data source through the ODBC Administrator, it adds the information you supply to `ODBC.INI`. `ODBC.INI` contains sections for each ODBC driver.

For example when you add a Versant ReVind ODBC data source the administrator:

- In the `[ODBC Data Sources]` section, adds an entry for the Versant ReVind ODBC data source using the following format:

```
data_source_name=VERSANT
```

- Adds a corresponding section for the data source name specified in the [ODBC Data Sources] section. This section uses the following format:

```
[data_source_name]
Driver=file_specification
Description=optional descriptive string
Host=host_server_system_name
Database=server_database_name
User ID=user_name
Password=password
```

Sample ODBC.INI File With Versant ReVind ODBC Data Source

The following sample ODBC.INI file shows the entries and sections required for a Versant ReVind ODBC data source called Demo.

WARNING:- Do not make changes to this file without using either the ODBC Setup program, the ODBC Administrator, or other utilities provided for maintaining installed drivers and data sources. Changing these files directly could result in a loss of information required by ODBC to execute.

```
[ODBC Data Sources]
Demo=VERSANT
[Demo]
Driver=C:\WINDOWS\SYSTEM\versantodbc.dll
Description=Sample ODBC.INI entry
Host=alpha
Database=demodb
User ID=vsqldb
Password=dummy
```

Configuring Data Sources on Windows 2000

On Windows 2000, ODBC.INI and ODBCINST.INI are not text files, but subkeys of the Windows 2000 registry. The Windows 2000 registry is a binary database maintained by the Windows 2000 operating system. The registry partitions information in it through a hierarchy of keys.

The `ODBC.INI` subkey resides two levels below the key `HKEY_CURRENT_USER`. The hierarchy is `HKEY_CURRENT_USER, Software, ODBC, ODBC.INI`. Within this level, the format of `ODBC.INI` is the same as described in the previous section.

Because Windows 2000 is a multiple-user operating system, each user has a distinct version of the `HKEY_CURRENT_USER` database. Because of this, each user must configure Versant ReVind data sources by using the ODBC Administrator as described in the section *Configuring Data Sources With the ODBC Administrator Utility*. Such users need to run the `ODBCAD32.EXE` program to run the administrator the first time.

The `ODBCINST.INI` subkey is a subkey of `HKEY_LOCAL_MACHINE`. Like the text file on Windows, it describes the number and types of ODBC drivers installed on the system. The ODBC Administrator reads the subkey when it invokes driver's setup `DLL` to add or modify a data source for the driver.

This chapter details the ODBC functionality Versant ReVind supports through the ODBC Driver.

The following are covered in detail:

- ODBC and SQL Support Levels
- Information the ODBC Driver Returns to SQLGetInfo
- Supported ODBC API Functions
- Versant ReVind Supported Data Types
- Versant ReVind Supported Isolation and Lock Levels

ODBC and SQL Support Levels

ODBC specifies general conformance levels in two areas:

- ODBC application programming interface (API). Versant ReVind supports all Core and Level 1 ODBC API functions, and most Level 2 functions.
- ODBC SQL syntax. Versant ReVind supports Extended SQL syntax.

More detail on that support has been provided in this Chapter. Specifically, it details the information the Versant ReVind ODBC driver returns when applications call the following functions:

- `SQLGetInfo` returns various details about the driver and its data source.
- `SQLGetFunctions` returns ODBC functions the driver supports.
- `SQLGetTypeInfo` returns data types the driver supports.

Information the ODBC Driver Returns to `SQLGetInfo`

Applications call the `SQLGetInfo` function to retrieve details about support a specific driver and data source provide for different ODBC functionality.

Applications supply the information type through the `fInfoType` argument to `SQLGetInfo`. `SQLGetInfo` returns the information to the `rgbInfoValue` output argument. The following lists each `fInfoType` argument the ODBC Driver recognizes.

For more detail about the `SQLGetInfo` function see the Microsoft ODBC *Programmer's Reference*.

Valid `fInfoType` arguments for `SQLGetInfo`

`SQL_ACCESSIBLE_PROCEDURES`

Guaranteed execute privileges on all procedures returned by `SQLProcedures`.

Returns — Y (guaranteed).

`SQL_ACCESSIBLE_TABLES`

Guaranteed read access to all table names returned by `SQLTables`.

Returns — N (not guaranteed).

SQL_ACTIVE_CONNECTIONS

Maximum number of active connections.

Returns — 10.

SQL_ACTIVE_STATEMENTS

Maximum number of active SQL statements.

Returns — 0 (no maximum).

SQL_ALTER_TABLE

Support for ALTER TABLE clauses.

Returns — SQL_AT_ADD_COLUMN (supports adding columns).

SQL_BOOKMARK_PERSISTENCE

Support for bookmarks.

Returns — SCROLL, DELETE and UPDATE.

SQL_COLUMN_ALIAS

Support for column aliases.

Returns — Y (supports).

SQL_CONCAT_NULL_BEHAVIOR

Result of concatenation of NULL character column with non-NULL column.

Returns — SQL_CB_NULL (result is null).

SQL_CONVERT_BIGINT

Conversion from BIGINT

Returns — SQL_CVT_CHAR | SQL_CVT_NUMERIC | SQL_CVT_TINYINT | SQL_CVT_SMALLINT | SQL_CVT_INTEGER | SQL_CVT_FLOAT | SQL_CVT_DOUBLE | SQL_CVT_BIT.

SQL_CONVERT_BINARY

Conversion from BINARY.

Returns — SQL_CVT_CHAR | SQL_CVT_BINARY.

SQL_CONVERT_BIT

Conversion from `BIT`.

Returns — `SQL_CVT_CHAR` | `SQL_CVT_NUMERIC` | `SQL_CVT_BIT` | `SQL_CVT_TINYINT` | `SQL_CVT_SMALLINT` | `SQL_CVT_BIGINT` | `SQL_CVT_INTEGER` | `SQL_CVT_FLOAT` | `SQL_CVT_REAL`.

`SQL_CONVERT_CHAR`

Conversion from `CHAR`.

Returns — `SQL_CVT_BIT` | `SQL_CVT_BINARY` | `SQL_CVT_NUMERIC` | `SQL_CVT_TINYINT` | `SQL_CVT_SMALLINT` | `SQL_CVT_BIGINT` | `SQL_CVT_DATE` | `SQL_CVT_FLOAT` | `SQL_CVT_TIME` | `SQL_CVT_TIMESTAMP` | `SQL_CVT_CHAR`.

`SQL_CONVERT_DATE`

Conversion from `DATE`.

Returns — `SQL_CVT_CHAR` | `SQL_CVT_DATE` | `SQL_CVT_TIMESTAMP`.

`SQL_CONVERT_DECIMAL`

Conversion from `DECIMAL`.

Returns — 0 (does not support).

`SQL_CONVERT_DOUBLE`

Conversion from `DOUBLE`.

Returns — 0 (does not support).

`SQL_CONVERT_FLOAT`

Conversion from `FLOAT`.

Returns — `SQL_CVT_CHAR` | `SQL_CVT_NUMERIC` | `SQL_CVT_BIT` | `SQL_CVT_TINYINT` | `SQL_CVT_SMALLINT` | `SQL_CVT_BIGINT` | `SQL_CVT_INTEGER` | `SQL_CVT_FLOAT` | `SQL_CVT_REAL`.

`SQL_CONVERT_FUNCTIONS`

Support for conversion functions.

Returns — `SQL_FN_CVT_CONVERT` (supports).

`SQL_CONVERT_INTEGER`

Conversion from INTEGER.

Returns — SQL_CVT_CHAR | SQL_CVT_NUMERIC | SQL_CVT_BIT | SQL_CVT_TINYINT | SQL_CVT_SMALLINT | SQL_CVT_BIGINT | SQL_CVT_INTEGER | SQL_CVT_FLOAT | SQL_CVT_REAL.

SQL_CONVERT_LONGVARBINARY

Conversion from LONGVARBINARY.

Returns — 0 (does not support).

SQL_CONVERT_LONGVARCHAR

Conversion from LONGVARCHAR.

Returns — 0 (does not support).

SQL_CONVERT_NUMERIC

Conversion from NUMERIC.

Returns — SQL_CVT_CHAR | SQL_CVT_INTEGER | SQL_CVT_FLOAT | SQL_CVT_REAL | SQL_CVT_BIT | SQL_CVT_TINYINT | SQL_CVT_SMALLINT | SQL_CVT_BIGINT.

SQL_CONVERT_REAL

Conversion from REAL.

Returns — SQL_CVT_CHAR | SQL_CVT_NUMERIC | SQL_CVT_BIT | SQL_CVT_TINYINT | SQL_CVT_SMALLINT | SQL_CVT_BIGINT | SQL_CVT_INTEGER | SQL_CVT_FLOAT | SQL_CVT_DOUBLE.

SQL_CONVERT_SMALLINT

Conversion from SMALLINT.

Returns — SQL_CVT_CHAR | SQL_CVT_NUMERIC | SQL_CVT_INTEGER | SQL_CVT_BIGINT | SQL_CVT_FLOAT | SQL_CVT_DOUBLE | SQL_CVT_TINYINT | SQL_CVT_SMALLINT | SQL_CVT_NUMERIC | SQL_CVT_BIT.

SQL_CONVERT_TIME

Conversion from TIME.

Returns — SQL_CVT_CHAR | SQL_CVT_TIME | SQL_CVT_TIMESTAMP.

SQL_CONVERT_TIMESTAMP

Conversion from `TIMESTAMP`.

Returns — `SQL_CVT_CHAR` | `SQL_CVT_DATE`.

`SQL_CONVERT_TINYINT`

Conversion from `TINYINT`.

Returns — `SQL_CVT_CHAR` | `SQL_CVT_NUMERIC` | `SQL_CVT_BIT` | `SQL_CVT_SMALLINT` | `SQL_CVT_INTEGER` | `SQL_CVT_BIGINT` | `SQL_CVT_FLOAT` | `SQL_CVT_DOUBLE` | `SQL_CVT_TINYINT`.

`SQL_CONVERT_VARBINARY`

Conversion from `VARBINARY`.

Returns — 0 (does not support).

`SQL_CONVERT_VARCHAR`

Conversion from `VARCHAR`.

Returns — 0 (does not support).

`SQL_CORRELATION_NAME`

Support for table correlation names

Returns — `SQL_CN_DIFFERENT` (supports, but correlation names must be different from table name).

`SQL_CURSOR_COMMIT_BEHAVIOR`

Effect of `COMMIT` operation on cursors and prepared statements.

Returns — `SQL_CB_CLOSE` (closes cursors but statements remain in prepared state).

`SQL_CURSOR_ROLLBACK_BEHAVIOR`

Effect of `ROLLBACK` operation on cursors and prepared statements.

Returns — `SQL_CB_CLOSE` (closes cursors but statements remain in prepared state).

`SQL_DATA_SOURCE_NAME`

Name of the data source as specified to the ODBC Administrator.

Returns — (string containing the name).

SQL_DATA_SOURCE_READ_ONLY

Access limited to read-only.

Returns — N (read-write access).

SQL_DATABASE_NAME

Name of the Versant ReVind data source on the server system

Returns — (string containing the name).

SQL_DBMS_NAME

Name of the database product supporting the data source.

Returns — Versant/SQL.

SQL_DBMS_VER

Version of the database product.

Returns — 7.0.1.3

SQL_DEFAULT_TXN_ISOLATION

Default transaction isolation level.

Returns — SQL_TXN_VERSIONING (serializable, implemented through non-locking scheme).

SQL_DRIVER_NAME

Name of the dynamic link library file for the ODBC Driver.

Returns — DHODBC.DLL.

SQL_DRIVER_ODBC_VER

Supported ODBC version.

Returns — 03.00.

SQL_DRIVER_VER

Current Version of the ODBC Driver.

Returns — 7.0.1.2.

SQL_EXPRESSIONS_IN_ORDERBY

Support for expressions in `ORDER BY` clause.

Returns — `Y` (supports).

`SQL_FETCH_DIRECTION`

Direction that `FETCH` operations can support.

Returns — `SQL_FD_FETCH_NEXT` (fetch next row only).

`SQL_FILE_USAGE`

Single-tier driver behavior.

Returns — `SQL_FILE_NOT_SUPPORTED` (not a single-tier driver).

`SQL_GETDATA_EXTENSIONS`

Supported extensions to `SQLGetData`.

Returns — `0` (does not support).

`SQL_GROUP_BY`

Relationship between `GROUP BY` clause and columns in the select list.

Returns — `SQL_GB_GROUP_BY_CONTAINS_SELECT` (`GROUP BY` clause must contain all non-aggregated columns in select list).

`SQL_IDENTIFIER_CASE`

Case-sensitivity of user-supplied names.

Returns — `SQL_IC_LOWER` (case insensitive, stored in lower case).

`SQL_IDENTIFIER_QUOTE_CHAR`

Character used to enclose delimited identifiers.

Returns — `"` (double quotation mark).

`SQL_KEYWORDS`

Data-source specific keywords.

Returns — `bigint, binary, call, cast, colgroup, compress, database, datapages, dba, default, definition, exclusive, file, hash, identified, indexpages, int, link, lock, log, long, lvarbinary, lvvarchar, minus, mode,`

modify, money, nocompress, nowait, number, pctfree, raw, real, references, rename, resource, rowid, rownum, share, size, space, statistics, storage_attributes, storage_manager, store_in_dharma, synonym, sysdate, systime, systimestamp, tinyint, uid, varbinary, abs, add_months, ascii, chartorowid, chr, concat, dayofmonth, dayofweek, dayofyear, db_name, decode, greatest, initcap, instr, last_day, least, length, locate, lpad, ltrim, months_between, next_day, nvl, object_id, prefix, quarter, rowidtochar, rpad, rtrim, substr, suffix, suser_name, sysdate, systime, systimestamp, to_char, to_date, to_number, to_time, to_timestamp, uid, user_name, week.

SQL_LIKE_ESCAPE_CLAUSE

Support for escape clause in `LIKE` predicates.

Returns — `Y` (supports).

SQL_LOCK_TYPES

Supported lock types.

Returns — `0` (does not support).

SQL_MAX_BINARY_LITERAL_LEN

Maximum length in hexadecimal characters of binary literals.

Returns — `10000`.

SQL_MAX_CHAR_LITERAL_LEN

Maximum length in characters of character string literals.

Returns — `10000`.

SQL_MAX_COLUMN_NAME_LEN

Maximum length of a column name.

Returns — `128`.

SQL_MAX_COLUMNS_IN_GROUP_BY

Maximum number of columns allowed in `GROUP BY` clause.

Returns — `0` (no limit).

SQL_MAX_COLUMNS_IN_INDEX

Maximum number of columns allowed in an index.

Returns — 100.

SQL_MAX_COLUMNS_IN_ORDER_BY

Maximum number of columns allowed in ORDER BY clause.

Returns — 0 (no limit).

SQL_MAX_COLUMNS_IN_SELECT

Maximum number of columns allowed in a select list.

Returns — 0 (no limit).

SQL_MAX_COLUMNS_IN_TABLE

Maximum number of columns allowed in a table.

Returns — 500.

SQL_MAX_CURSOR_NAME_LEN

Maximum length of a cursor name.

Returns — 128.

SQL_MAX_INDEX_SIZE

Maximum number of bytes allowed in the combined fields of an index.

Returns — 900.

SQL_MAX_OWNER_NAME_LEN

Maximum length of an owner name.

Returns — 128.

SQL_MAX_PROCEDURE_NAME_LEN

Maximum length of a procedure name.

Returns — 128.

SQL_MAX_QUALIFIER_NAME_LEN

Maximum length of a table or column qualifier.

Returns — 32.

SQL_MAX_ROW_SIZE

Maximum length in bytes of a table row.

Returns — 2048.

SQL_MAX_ROW_SIZE_INCLUDES_LONG

Whether maximum row size includes LONGVARCHAR and LONGVARBINARY.

Returns — Y.

SQL_MAX_STATEMENT_LEN

Maximum number of characters in a SQL statement.

Returns — 50000.

SQL_MAX_TABLE_NAME_LEN

Maximum length of a table name.

Returns — 128.

SQL_MAX_TABLES_IN_SELECT

Maximum number of tables allowed in FROM clause.

Returns — 250.

SQL_MAX_USER_NAME_LEN

Maximum length of a user name.

Returns — 128.

SQL_MULT_RESULT_SETS

Support for multiple result sets.

Returns — N (does not support).

SQL_MULTIPLE_ACTIVE_TXN

Support for active transactions on multiple connections.

Returns — Y (supports).

SQL_NEED_LONG_DATA_LEN

Whether data source requires length of `LONGVARCHAR` and `LONGVARBINARY` data.

Returns — `N`.

`SQL_NON_NULLABLE_COLUMNS`

Support for `NOT NULL` clause in `CREATE TABLE` statement.

Returns — `SQL_NNC_NON_NULL` (supports).

`SQL_NULL_COLLATION`

Where null values are sorted in a list.

Returns — `SQL_NC_LOW` (sorted at the low end of the list).

`SQL_NUMERIC_FUNCTIONS`

Numeric functions supported.

Returns — `SQL_FN_NUM_ABS` | `SQL_FN_NUM_ACOS` | `SQL_FN_NUM_ASIN` | `SQL_FN_NUM_ATAN` | `SQL_FN_NUM_ATAN2` | `SQL_FN_NUM_CEILING` | `SQL_FN_NUM_COS` | `SQL_FN_NUM_DEGREES` | `SQL_FN_NUM_EXP` | `SQL_FN_NUM_FLOOR` | `SQL_FN_NUM_LOG10` | `SQL_FN_NUM_MOD` | `SQL_FN_NUM_PI` | `SQL_FN_NUM_POWER` | `SQL_FN_NUM_RADIANS` | `SQL_FN_NUM RAND` | `SQL_FN_NUM_SIGN` | `SQL_FN_NUM_SIN` | `SQL_FN_NUM_SQRT` | `SQL_FN_NUM_TAN` (bit masks specifying support for indicated functions).

`SQL_ODBC_API_CONFORMANCE`

ODBC API conformance level.

Returns — `SQL_OAC_LEVEL1` (supports level 1).

`SQL_ODBC_SAG_CLI_CONFORMANCE`

SQL Access Group (SAG) conformance.

Returns — `SQL_OSCC_COMPLIANT` (complies with SAG CLI specification).

`SQL_ODBC_SQL_CONFORMANCE`

ODBC SQL syntax conformance.

Returns — `SQL_OSC_EXTENDED` (supports extended SQL syntax as defined by ODBC).

`SQL_ODBC_SQL_OPT_IEF`

Referential integrity syntax support.

Returns — Y (supports referential integrity syntax).

SQL_ODBC_VER

ODBC version supported by driver manager.

Returns — Returned by ODBC driver manager.

SQL_ORDER_BY_COLUMNS_IN_SELECT

Whether columns in ORDER BY clause must also be in select list.

Returns — N.

SQL_OUTER_JOINS

Support for outer joins.

Returns — Y (supports).

SQL_OWNER_TERM

Term for entity that has owner privileges on objects.

Returns — owner.

SQL_OWNER_USAGE

Statements that support use of owner qualifiers.

Returns — SQL_OU_DML_STATEMENTS | SQL_OU_PROCEDURE_INVOCATION |
SQL_OU_TABLE_DEFINITION | SQL_OU_INDEX_DEFINITION | SQL_OU_PRIVILEGE_DEFINITION.

SQL_POS_OPERATION

Supported operations in SQLSetPos.

Returns — 0 (does not support SQLSetPos).

SQL_POSITIONED_STATEMENTS

Statements that support positioned operations.

Returns — SQL_PS_POSITIONED_DELETE | SQL_PS_POSITIONED_UPDATE |
SQL_PS_SELECT_FOR_UPDATE.

SQL_PROCEDURE_TERM

Term for procedures.

Returns — `procedure`.

`SQL_PROCEDURES`

SQL procedures support.

Returns — `Y` (supports SQL procedures).

`SQL_QUALIFIER_LOCATION`

Position of qualifier in a qualified table name.

Returns — `SQL_QUAL_START` (at the start of the name).

`SQL_QUALIFIER_NAME_SEPARATOR`

Character used to separate table, column qualifiers.

Returns — `.` (period).

`SQL_QUALIFIER_TERM`

Term for object that qualifies table names.

Returns — `database`.

`SQL_QUALIFIER_USAGE`

Statements that support qualifiers.

Returns — `0` (does not support).

`SQL_QUOTED_IDENTIFIER_CASE`

Case-sensitivity of quoted user-supplied names.

Returns — `SQL_IC_SENSITIVE` (case sensitive, stored in mixed case).

`SQL_ROW_UPDATES`

Detect changes to any row in mixed-cursor operations.

Returns — `Y`.

`SQL_SCROLL_CONCURRENCY`

Concurrency control options supported for scrollable cursors.

Returns — `0` (does not support).

SQL_SCROLL_OPTIONS

Options supported for scrollable cursors.

Returns — 0 (does not support).

SQL_SEARCH_PATTERN_ESCAPE

Character to permit wildcard characters in search strings.

Returns — \ (backslash).

SQL_SERVER_NAME

Name of the system where the Versant ReVind data source resides.

Returns — (string containing the name).

SQL_SPECIAL_CHARACTERS

Special characters allowed in user-supplied names.

Returns — _ (underscore).

SQL_STATIC_SENSITIVITY

Whether static cursor changes are detectable.

Returns — 0 (does not support static cursors).

SQL_STRING_FUNCTIONS

String functions supported.

Returns — `SQL_FN_STR_ASCII` | `SQL_FN_STR_LTRIM` | `SQL_FN_STR_RTRIM` (bit masks specifying support for ASCII, LTRIM, RTRIM functions).

SQL_SUBQUERIES

Predicates that support subqueries.

Returns — `SQL_SQL_COMPARISON` | `SQL_SQL_EXISTS` | `SQL_SQL_IN` | `SQL_SQL_QUANTIFIED`.

SQL_SYSTEM_FUNCTIONS

System functions supported.

Returns — `SQL_FN_SYS_USERNAME` (bit mask specifying support for USERNAME function).

SQL_TABLE_TERM

Term for tables.

Returns — `table`.

`SQL_TIMEDATE_ADD_INTERVALS`

Timestamp intervals supported for `TIMESTAMPADD` function.

Returns — 0 (does not support `TIMESTAMPADD`).

`SQL_TIMEDATE_DIFF_INTERVALS`

Timestamp intervals supported for `TIMESTAMPDIFF` function.

Returns — 0 (does not support `TIMESTAMPDIFF`).

`SQL_TIMEDATE_FUNCTIONS`

Date-time functions supported.

Returns — `SQL_FN_TD_DAYOFMONTH` | `SQL_FN_TD_DAYOFWEEK` | `SQL_FN_TD_DAYOFYEAR` | `SQL_FN_TD_MONTH` | `SQL_FN_TD_QUARTER` | `SQL_FN_TD_WEEK` | `SQL_FN_TD_YEAR` | `SQL_FN_TD_HOUR` | `SQL_FN_TD_MINUTE` | `SQL_FN_TD_SECOND` (bit masks specifying support for indicated functions).

`SQL_TXN_CAPABLE`

Support for DML, DDL within transactions.

Returns — `SQL_TC_ALL` (supports both DML and DDL).

`SQL_TXN_ISOLATION_OPTION`

Options for setting transaction isolation levels.

Returns — `SQL_TXN_SERIALIZABLE` | `SQL_TXN_REPEATABLE_READ` | `SQL_TXN_VERSIONING` (bit masks specifying serializable or repeatable read).

`SQL_UNION`

UNION support.

Returns — `SQL_U_UNION` | `SQL_U_UNION_ALL`.

`SQL_USER_NAME`

Name of user connected to the data source.

Returns — (string containing the name).

Supported ODBC API Functions

The Versant ReVind ODBC Driver supports all Core and Level 1 API functions, and most Level 2 functions. Following is a complete list of the functions the driver supports (applications can request this same information through the `SQLGetFunctions` function).

SQLAllocHandle
SQLBindCol
SQLBindParameter
SQLBrowseConnect
SQLCancel
SQLColAttribute
SQLColumnPrivileges
SQLColumns
SQLConnect
SQLDescribeCol
SQLDescribeParam
SQLDisconnect
SQLDriverConnect
SQLExecDirect
SQLExecute
SQLExtendedFetch
SQLFetch
SQLForeignKeys
SQLFreeHandle
SQLGetConnectAttr
SQLGetCursorName
SQLGetData
SQLGetDiagRec
SQLGetFunctions
SQLGetInfo
SQLGetStmtAttr
SQLGetTypeInfo
SQLNativeSQL
SQLNumParams
SQLNumResultCols
SQLParamData
SQLPrepare
SQLPrimaryKeys
SQLProcedureColumns
SQLProcedures

SQLPutData
SQLRowCount
SQLSetConnectAttr
SQLSetCursorname
SQLSetStmtAttr
SQLSpecialColumns
SQLStatistics
SQLTablePrivileges
SQLTables
SQLEndTran

Versant ReVind Supported Data Types

Versant ReVind supports a set of data types that the Versant ReVind ODBC Driver maps to corresponding ODBC SQL data types. Following are the Versant ReVind data types and the corresponding ODBC data types.

VERSANT and Corresponding ODBC Data Types

<i>VERSANT Data Type</i>	<i>ODBC Data Type</i>
BIGINT	SQL_BIGINT
BINARY	SQL_BINARY
BIT	SQL_BIT
CHARACTER	SQL_CHAR
DATE	SQL_DATE
DECIMAL	SQL_DECIMAL
DOUBLE PRECISION	SQL_DOUBLE
FLOAT	SQL_DOUBLE
FLOAT	SQL_FLOAT
INTEGER	SQL_INTEGER
LVARBINARY	SQL_LONGVARBINARY
LVARCHAR	SQL_LONGVARCHAR
NUMERIC	SQL_NUMERIC
REAL	SQL_REAL
SMALLINT	SQL_SMALLINT
TIME	SQL_TIME
TIMESTAMP	SQL_TIMESTAMP
TINYINT	SQL_TINYINT

VARBINARY
VARCHAR

SQL_VARBINARY
SQL_VARCHAR

Versant ReVind Supported Isolation and Lock Levels

The Versant ReVind ODBC driver supports the `SQL_TXN_VERSIONING` isolation level as specified by ODBC. Transactions are serializable, but higher concurrency is possible than with `SQL_TXN_SERIALIZABLE`.

Index

A

adding Versant ReVind ODBC data sources 8

C

configuring data sources on Windows 2000 10

configuring data sources with the ODBC
administrator utility 8

conformance notes 6

F

format of ODBCi.ini, the driver manager initialization
file 9

H

handling of user name and password 6

I

information the ODBC driver returns to
SQLGetInfo 14

invoking the ODBC administrator utility 8

M

microsoft access notes 6

O

ODBC and sql support levels 14

S

sample ODBC.ini file with VERSANT ReVind odbc
data source 10

T

thirdparty tools 5

V

valid finfo type arguments for sqlgetinfo 14

VERSANT and corresponding ODBC data types 30

VERSANT ReVind supported data types 30

VERSANT ReVind supported isolation and lock
levels 31