

# ORACLE7 (TM) ODBC DRIVER

## RELEASE NOTES

\*\*\* THIS IS IMPORTANT INFORMATION. \*\*\*

\*\*\* PLEASE DISTRIBUTE TO ALL SUPPORT AND TECHNICAL PEOPLE. \*\*\*

### Oracle7 ODBC Driver Version 1.1.0.0 (Level 1 Driver)

#### KNOWN ISSUES

##### Date and Time default

If you use a DATE data type column and you don't specify a date component, the default date is the first day of the current month. If you use a DATE data type field and you don't specify a time component, the default time is 12:00:00am.

##### Time format

Oracle7 only stores hours, minutes and seconds. It does not store decimal for seconds. The valid time format is {t '10:20:43'}. If you show a BOUND column for data type DATE, the decimal time portion is returned (10:20:43.0) but it is not stored on the Oracle7 server.

##### Cannot retrieve dates in the BC era

The driver will let you insert a date in the BC era with "insert into table values (TO\_DATE('0001-10-18 BC','YYYY-MM-DD BC'))", but it will return an error message "Numeric value out of range" when you retrieve the value. This is because the ODBC specification treats negative and zero year values as invalid.

##### Misleading error message for Full Year less than or equal to zero

As part of the ODBC specifications, values for a full year less than or equal to zero are invalid. If a value less than or equal to zero is inserted as a year, the error message "Full year must be between -4713 and +4713" will be returned. This message is misleading. Oracle7 allows a full year to be less than or equal to zero, but to comply to the ODBC specifications, the Oracle7 ODBC Driver doesn't.

### **Numeric truncation and rounding**

Inserted numeric values are truncated and/or rounded without any notification to the user. e.g. 1) Inserting 123456.1256 into a data type of number(8,2) results in storing the value 123456.13 (the last 2 decimals are truncated and the second decimal is rounded up). 2) Inserting 123.4 into a data type of number(8,0) results in storing the value 123 (the two decimals are truncated).

### **Inserting LONG RAW**

The insertion of a large amount of data (e.g. using Access with an OLE data type which is mapped to a LONG RAW) is limited by the amount of free memory on the user's machine.

### **LONG RAW and RAW are not searchable**

The data types LONG RAW and LONG are NOT searchable. They cannot be used in: WHERE, GROUP BY, ORDER BY, CONNECT BY, or DISTINCT operator in SELECT statements.

### **Entering data into a LONG or RAW**

When you enter character data into a LONG or RAW column, you must enter 2-digit pairs. Oracle7 treats the 2-digit pair as a 1-hex digit. (i.e. inserting 12345 would store 1234).

### **Valid Value Range Limited by Oracle7 BCD Data Format**

Since Oracle7 stores real numbers in BCD format, with a limit of 22 characters, the valid numeric range is approximately  $1e-129$  to  $1e+126$ . This means that a number may have a maximum of 126 digits when inserted.

### **Numeric Scale Values Can Now Exceed the Precision Values**

With the ORACLE V6 ODBC driver (bundled with MS Access 1.1 and MS Visual Basic 3.0), scale values could not exceed the precision value for the number data type (i.e. number(precision, scale)). With the Oracle7 ODBC Driver, this limitation has been removed. For example, it is now possible to have a specification of "number(38,127)" with the scale exceeding the precision.

### **Precision Corrupted For Some Float Values**

When a string of digits is inserted into a float field, they are first converted by the server to BCD format for storage. When they are retrieved, they are again converted from BCD format to float format. This conversion process is not always exact, so there is occasionally some small conversion error introduced (e.g. if the value 41.256 is inserted, the value retrieved is 41.256000000000001).

### **SQLProcedureColumns: Sorting is by COLUMN\_NAME**

Currently the results from SQLProcedureColumns are sorted by COLUMN\_NAME. The ODBC specifications specify that the sorting should be by COLUMN\_TYPE. Correcting this would require extensive modifications to SQLProcedureColumns. This will be remedied in the next version of the Oracle7 ODBC Driver. The current sort order will not affect application programs that use the driver.

### **Installer Overwrites Existing Driver Without Notification**

The Oracle7 ODBC Setup program (used to install the driver) will automatically overwrite any existing copy of the Oracle7 ODBC Driver residing in the directory where you are installing it. If you wish to retain the older copy of the Oracle7 ODBC Driver, it should be relocated to a different directory or copied to a floppy diskette prior to starting the installation. The filenames that need to be relocated are sqora7.dll and sqorast7.dll.

### **PacketSize Options**

In order to send a specific packet size to the server, add the line "PacketSize=" to your ODBC.INI file. Only 1024, 2048, 4096 and 8192 are valid sizes (in bytes). When you add an ODBC DataSource, the line "PacketSize=2048" will be placed in your ODBC.INI file but only for TCP/IP servers ("t:" prefix). If there is no entry in your ODBC.INI file, then no PacketSize will be sent in the connection string.

### **Transactions are for DML**

Transactions (COMMIT, ROLLBACK) are only for DML (e.g. alter table, etc.). If you perform a transaction for a non-DML, it won't return any error but the transaction will not be performed.

### **Execution of DDL Statements**

All DDL statements (CREATE, DROP, GRANT) which are prepared using Oracle are also executed by the server. The Oracle7 ODBC Driver has no control over this functionality and therefore preparing any DDL statements using the Oracle7 ODBC Driver will also result in execution of the statement.

**Timestamp: decimal value on seconds**

Inserting a decimal value on seconds for a Timestamp datatype while using the Timestamp parameter will result in truncating the decimal portion of the seconds, without reporting a truncation error. Example: insert into table values ({ts '1992-12-31 23:59:59:123'}) will result in storing the value '1992-12-31 23:59:59'.

### **WHERE clause is executed at Fetch time**

The "WHERE" clause in an SQL Statement is not executed when SQLExecute is called. It is executed when SQLFetch is called. If there's an error in the "WHERE" clause of the SQL Statement, SQLExecute will return SQL\_SUCCESS, but SQLFetch will return SQL\_ERROR.

### **Parameter Limitation with Stored Procedures**

Currently the Oracle7 ODBC Driver only allows a maximum of 30 parameters being passed to an Oracle Stored Procedure. This limitation may be fixed in the next release of the Oracle7 ODBC Driver.

## APPLICATIONS LIMITATIONS

### Microsoft Access 1.1

Connect, SQL Statements, Read, Write

In order to be able to update a table while using Access 1.1, the user has to have at least one unique index setup on that table on the server side before attaching to it in Access 1.1. Otherwise the table cannot be updated, but will still be accessible for read-only.

You can control the table types Access sends to an ODBC driver (SQLTables) by adding the following to your MSACCESS.INI under the [ODBC] section:

```
[ODBC]
AttachableObjects='TABLE'
```

This would only retrieve 'TABLE' type for SQLTables. Doing so will increase performance at connection time, as only tables will be looked for, not views, system tables, aliases or synonyms. If you omit this line, the default will include all types (tables, views, system tables, aliases and synonyms). To list 2 or more types, you must enclose them in single quotes and separate them by commas. For example, to list tables and views you would add the line:

```
[ODBC]
AttachableObjects='TABLE','View'
```

You can use the TransferDatabase Action macro to attach to a specific table. This will again increase the performance (i.e. decrease connect time) because Access will bypass getting a list of all tables on the database. The syntax is:

```
DoCmd TransferDatabase [transfertype], databasetype, databasename [, objectype],
source, destination [, structureonly]
```

For further information, please refer to the MS Access 1.1 documentation.

### Lotus 1-2-3 Release 4.0

Connect, SQL Statements, Read

To use Lotus 1-2-3 with ODBC Drivers, you must add the following section to your LOTUS.BCF file:

```
DN="ODBC"
DL="DLODBC"
DD="All ODBC Datasources";
```

To connect to Oracle7 with Userid and Password (UI, PW), you must add the following section to your LOTUS.BCF file. Substitute the name of your data source that you will be using with the 'Data Source Name' listed below.

DB="Data Source Name"  
DN="ODBC"  
AC=UI,PW;

Note that Lotus 1-2-3 does not allow spaces in Data Source Names. This is a limitation with the Lotus DataLens Driver.

Column Names must be alphanumeric and cannot contain spaces (A-Z, a-z, 0-9 are permissible).

A column name in your Oracle7 database table cannot have the same name as any cell address in Lotus 1-2-3. For example, your table cannot have column names like C1, D1, etc. because these are cell references in Lotus 1-2-3.

Problems with updating the database. After a table is opened, you need to select "Query | Update Database" in order to update a table. An error message comes up "Driver cannot perform this database operation". Lotus Corp. has been informed about this limitation.

The performance of the Oracle7 ODBC Driver with Lotus 1-2-3 R4.0 is slow due to issues in the ODBC design of Lotus 1-2-3 R4.0

### **Microsoft Query Beta**

Connect, SQL Statements, Read, Write

Problems with lowercase column names. MS Query will return the error "Invalid Column name" if you query a column that was created as lowercase. (e.g. "create table test ('col1' char(3))"). MS Query does a select with "select col1 from test" which looks for COL1 in uppercase. The proper SQL statement should be "select 'col1' from test" when a column is created as lowercase. Microsoft has been informed about this limitation.

### **Visual Basic 3.0**

Connect, SQL Statements, Read, Write

You can control the table types a Visual Basic application sends to an ODBC driver (SQLTables) by creating an initialization file (INI) with the same name as the application, and adding the following information:

```
[ODBC]  
AttachableObjects='TABLE'
```

This would only retrieve 'TABLE' type for SQLTables. Doing so will increase performance at connection time, as only tables will be looked for, not views, system tables, aliases or synonyms. If you omit this line, the default will include all types (tables, views, system tables, aliases and synonyms). To list 2 or more types, you must enclose

them in single quotes and separate them by commas. For example, to list tables and views in the Visual Basic application, VBTEST.EXE, create an initialization file called VBTEST.INI and add the following information:

```
[ODBC]
AttachableObjects='TABLE','View'
```

Similarly, you can control the table types Visual Basic 3.0 sends to an ODBC driver (SQLTables) by adding the following information to your VB.INI which is located in your Windows directory:

```
[ODBC]
AttachableObjects='TABLE'
```

This would only retrieve 'TABLE' type for SQLTables. If you omit this line, the default includes tables, views, system tables, aliases and synonyms. To list 2 or more types, you must enclose them in single quotes and separate them by commas. For example, to list tables and views in a Visual Basic Table List Box, add the following information:

```
[ODBC]
AttachableObjects='TABLE','VIEW'
```

For further information, please refer to the Visual Basic 3.0 documentation.

Problems with the sample application Visdata's Row finder. It does not recognize the last record when performing the find option. This is a problem with the application Visdata of Microsoft Visual Basic 3.0.

Problems with running a query through Query Builder in Visdata application. The error message: "Couldn't find 'SCOTT.MDB'. Error #3024" comes up and the query is terminated. This is a problem with the application Visdata of Microsoft Visual Basic 3.0. Microsoft has been informed of this limitation.

*Programmer's notes for Visual Basic 3.0:*

- The Close function needs to be called whenever a dynaset or snapshot is not being used anymore. This will reduce the server cursor hold up.
- If ODBC functions are called directly, after calling SQLAllocStmt to allocate a statement, SQLFreeStmt should be called to free up the cursor in the database server. The default number of cursors limit for Oracle7 is 50. This number can be altered by the DBA. Please refer to the DBA manual for more information on cursors limit.
- If ODBC functions are called directly, all the necessary ODBC functions and constants have to be declared in the Visual Basic application.
  
- The CreateDynaset function fetches bookmarks (primary keys) and a small amount of data around current scrolled to position (e.g. pagedown, goto last record, etc.). The CreateSnapshot function fetches all data up to the position scrolled to. Therefore, CreateDynaset can have faster response than CreateSnapshot at the beginning, but in

general its performance will be slower for any scrolling command.

### **PowerBuilder 2.0a**

Cannot Connect

PowerBuilder 2.0a performs an SQLExecDirect to create its own catalog file. The SQL Statement is built with ????? as data types. This is because PowerBuilder is trying to use a data type not supported by Oracle7 DBMS. Powersoft has been informed of this limitation and the workaround is to upgrade to PowerBuilder Version 3.0.

### **PowerBuilder 3.0**

Connect, SQL Statements, Read, Write

There is a problem if a table or a column name is UPPERCASE. PowerBuilder converts all table names and columns names to lowercase. A table created as 'Create table newtab (col1 char)' is read through PowerBuilder as 'Select "col1" from "newtab"'. The table or column is not found as they are stored in uppercase as NEWTAB and COL1. This is a problem with PowerBuilder but Powersoft has provided the following workaround: in the PBODBC30.INI file, insert the line "IdentifierCase=3" under the heading "Oracle7".

### **ODBC Test 1.0**

Connect, SQL Statements, Read, Write

When retrieving data from a LONG data type column that is longer than 298 characters or digits long, ODBC Test inserts the word "TRUNC" after every 298th character in the result set. This is a problem in ODBC Test 1.0. Microsoft has been informed of this limitation and there's currently no workaround.