

**Comparative Performance Tests of ODBC Drivers
and Proprietary Database Application
Programming Interfaces**

**Prepared by: Ken North
August 3, 1995**

Executive Summary

Standard or common programming interfaces for accessing SQL data provide connectivity to multiple databases by the use of a single application programming interface (API). One such interface is Open Database Connectivity (ODBC), an API from Microsoft, derived from the call level interface (CLI) published by the SQL Access Group and X/Open in 1992. An issue important to developers considering the use of ODBC is whether ODBC's performance is comparable to that of proprietary programming interfaces, the native data access interfaces supported by individual DBMS products.

Resource Group, Inc., conducted verifiable, independent tests comparing performance between ODBC drivers and proprietary database programming interfaces. This report compares the performance of ODBC drivers with proprietary SQL APIs when commanded to execute a suite of identical queries.

SQL Application Performance

SQL application performance will vary for a number of reasons other than the choice of API. To clarify the effect that an API has on performance, the tests used the same clients, servers, network, tables and SQL statements. The performance tests used a benchmark program that permits the user to select either native API or ODBC calls to process SQL statements. The program executes queries that retrieve data using SELECT statements, updates tables using UPDATE statements, deletes rows using DELETE, and so on. In simple terms, each test performs the same operations against identical data but varies the programming interface. This technique produces test results that are a valid comparison of API solutions: native APIs versus ODBC. ODBC tests used DataDirect ODBC Drivers from INTERSOLV, Inc., of Rockville, Maryland.

Performance Test Organization and Operation

The software used for the performance tests is a Microsoft Windows client application comprised of several programs written in C.* The main benchmark application is a Windows executable (EXE) program that selectively calls dynamic link libraries (DLLs) to execute the logic for either ODBC or native API access to a DBMS. The benchmark application executes one or more tests from among a user-selectable menu of ten tests. The benchmark programs use an employee and department table with columns representative of typical data, such as hire date and salary. The application records the execution time for the following:

- insertion test
- five SELECT (retrieval) queries
- two tests of updates
- two tests of deletions

The tests include logic that uses various SQL programming techniques, such as direct or immediate execution queries and prepare and execute queries. The application records the following information in two log files:

- test results counts
- SQL statements executed
- execution times
- other pertinent information

The tests used a variety of client and server hardware and software, including Intel and RISC processors. Also used were a variety of network libraries and operating systems (Windows 3.1, NetWare, Windows NT, OS/2, Sun Solaris). The suite of tests compared ODBC driver performance with embedded SQL or call level interfaces for the following database systems:

- Oracle 7
- INFORMIX 5.01
- Sybase System 10.0.2

The benchmark summary presented here represents multiple tests using tables populated with 2500 employees and 70 departments. Multiple runs produced execution times used to compute a mean time for each test.

The tables that follow are the results from calculating the mean of the execution time for native API and ODBC access. Line one is the average execution time using native access. Line two is ODBC access. Line three is the difference (in seconds) calculated by subtracting line two from line one. A minus in line three indicates a test where ODBC was faster, a plus indicates a test where ODBC was slower than native performance. Line four is ODBC execution time expressed as a percentage where native execution time equals 100 percent.

Oracle 7: Oracle Call Interface and ODBC

API	Pop	Qry 1	Qry 2	Qry 3	Qry 4	Qry 5	Upd	Up 1	Del
Native	206.15	14.78	2.54	30.59	7.16	68.80	8.81	145.23	19.34
ODBC	199.78	14.02	2.58	33.32	6.81	51.63	9.93	146.16	17.98
Diff (sec)	-6.37	-0.76	+0.04	+2.73	-0.35	-17.17	+1.12	+0.93	-1.36
Diff %	96.9	94.8	101	109	95.1	75	113	101	93

INFORMIX 5.01: Embedded SQL and ODBC

API	Pop	Qry 1	Qry 2	Qry 3	Qry 4	Qry 5	Upd	Up 1	Del	Del1
Native	36.03	8.90	2.41	18.81	7.75	74.68	4.00	34.27	6.50	36.01
ODBC	39.16	9.38	2.16	19.26	7.98	73.93	3.75	32.35	5.94	33.80
Diff (sec)	+3.13	+0.48	-0.25	+0.45	+0.23	-0.75	-0.25	-1.92	-0.56	-2.21
Diff %	109	106	89.6	102	103	98.9	93.7	94.4	91.4	93.9

SYBASE 10.0.2: CT-Library and ODBC

API	Pop	Qry 1	Qry 2	Qry 3	Qry 4	Qry 5	Upd	Up 1	Del	Del1
Native	55.87	11.81	1.61	27.39	20.95	252.38	8.94	296.00	4.77	105.29
ODBC	58.54	10.25	1.22	22.39	13.32	258.34	5.68	171.16	5.88	111.34
Diff (sec)	+2.67	-1.56	-0.39	-5.00	-7.63	+5.96	-3.26	-124.84	+1.11	+6.05
Diff %	105	86.8	75.8	81.7	63.6	102.0	63.5	57.8	123.0	106.0

Legend:

- Pop = populate tables
- Qry 1 = select all employees
- Qry 2 = select by hire date
- Qry 3 = join and order by
- Qry 4 = salary grouped by department
- Qry 5 = select 1 row
- Upd = update all rows
- Up 1 = update 1 row by employee ID
- Del = delete all rows
- Del1 = delete 1 row by employee ID

Note: To correct a technical error in earlier tests, the Qry 5 tests were repeated using a test configuration different from the other SYBASE tests. The times for the Populate query are unaudited.

Analysis

An examination of the mean execution time of 29 tests reveals that 12 tests showed faster native performance, while 17 tests showed faster ODBC performance. In the 17 tests in which ODBC had a performance advantage, the difference was greater than 10 percent in eight tests. In the 12 tests in which native performance was better, the difference was greater than 10 percent in two tests. The overall performance difference summary by database is:

- Oracle 7 - 97.64%
- INFORMIX 5.01- 98.67%
- Sybase System 10 - 86.52%

Note: Results are meaningful only for an API to API comparison. These performance tests did not use the same combination of hardware and software for every database system, so it is not feasible to do a valid comparison of database servers by using these results.

Conclusion

These tests demonstrate that performance using ODBC may often be superior to native performance, and when native performance is superior, the difference is often minor. The obvious conclusion is that there is no performance degradation from using ODBC.

*Source code for the native and ODBC programs is available from *Dr. Dobb's Journal*. "Performance Testing, ODBC, and Native SQL APIs" describes the programs and includes a link for downloading the source code.

"Performance Testing, ODBC, and Native SQL APIs"
<http://www.ddj.com/documents/ddj9613b/>