Patricia Seybold's UNIX IN THE OFFICE

VOL. 5, NO. 1
ISSN: 0887-3054
JANUARY 1990

PRODUCTS • TRENDS • ISSUES • ANALYSIS

Oracle's RDBMS

The Product behind the Marketing

By Laure Brown

IT'S NO EASY task for a successful company to remain on the leading edge. The tendency is to ride out whatever it is that brought success in the first place.

So here we find Oracle—by far the leading relational database management system (RDBMS) vendor. It's actually the world's third largest software vendor, and more successful than ever. Even the EEC (European Economic Community) has recognized Oracle and chosen it as its standard RDBMS—quite a coup.

The company prides itself on the fact that its RDBMS was ahead of its time. Introduced in (*continued on page 3*)

• EDITORIAL •

Goodbye, Unix. Hello, Mach!

A Salute to Pioneers

By Judith S. Hurwitz

THE OTHER DAY, I came across a summary of a Microsoft online conference in which Bill Gates, the president of that OS/2 giant, commented on some of Unix's deficiencies. Not surprising. Gates has a lot of reasons for not wanting Unix to overshadow OS/2. And, these days, Gates is also in the habit of heavily promoting that old workhorse, DOS. Some of his observations have to be taken with a grain or two of salt, but others are worth discussing.

Gates stated that Unix has no chance to capture the desktop because there is no consensus on issues like instruction set, printer drivers, and the like. Unless these issues are resolved, he said, Unix cannot hope for a key role. And, given the current state of affairs in the Unix world, he is skeptical that a consensus will be reached soon. Surprisingly, Gates contends that NeXT and its Mach operating system might be the one bright spot.

We think that Bill Gates might be right. The position of NeXT as a strategic vendor in the workstation market has shifted with the growing importance of Mach. Steven Jobs's first announcement that he would use Mach as the operating system for his new workstation was met with some skepticism. Mach was properly viewed as a research tool, not yet ready for prime time. But now that the Open Software Foundation (OSF) has selected Mach for its operating system and will do the work to commercialize it, we suspect that Mach will be viewed very differently in the marketplace.

Mach (and perhaps NeXT) could be the X-factor in the Unix story. What distinguishes Mach from Unix is its foundation. Mach does not suffer from the patches and repatches that have typified the last 20 years of Unix. Although it is definitely tied to Unix (it is compatible with the Berkeley version), Mach is enough of a clean slate to satisfy many of the Unix detractors—those who think Unix is too old and brittle for commercial use.

Does this mean that NeXT will become the future Macintosh of the Unix community? Not necessarily (of course, anything is possible). NeXT may suffer the fate of many a pioneer—showing the way to the future and watching others grab the lead. Today, NeXT is too small an island to be viewed as a power in its own right. But we expect that others will latch onto the idea of using this next-generation operating system as a way of leveraging new ideas like multiprocessing and true multitasking.

The meaning of all these debates, wars, and discussions is becoming clear. We are at the beginning of a new generation of computer technology. Change is coming fast on many fronts: hardware, software, memory, operating systems. Users stuck with the reality of their day-to-day work look at all of this and wonder what is going on. When they ponder the possibility of adding 4 or 8MB of memory into their desktop systems, they gasp in astonishment. When they look at OSF's plans to drop traditional Unix and move to Mach, they are perplexed. What's wrong with good old Unix anyway?

It is even harder for users to understand what all of this stuff is for. There aren't too many applications out there that need it. Where is the next Lotus 1-2-3 that users simply can't live without? Ironically, a simple event such as 1-2-3 may never happen again. Instead, users will slowly realize that, to do multimedia applications and multiwindowing with any speed and efficiency, they will have to move to the future. But they'll move kicking and screaming. At the same time, we need pioneers like Steven Jobs, the Open Software Foundation, and even Bill Gates to challenge us and tickle our imaginations about what is possible. ◐

Patricia Seybold's
Office
Computing
Group

Publisher **PATRICIA B. SEYBOLD**

Editor-in-Chief **JUDITH S. HURWITZ**

Managing Editor
JOHN R. RYMER

Senior Editors
JUDITH R. DAVIS
Telephone: (617) 861-3926
RONNI T. MARSHAK
MICHAEL D. MILLIKIN

Associate Editor
LAURE BROWN

News Editor
DAVID S. MARSHAK

Contributing Editor
GARY J. NUTT
Telephone: (303) 444-0341

Sales Director
RICHARD ALLSBROOK JR.

Circulation Manager
DEBORAH A. HAY

Customer Service Manager
MICHELLE MANN

148 State Street, Suite 612, Boston, Massachusetts 02109 Telephone: (617) 742-5200 FAX: (617) 742-1028
MCI: psocg / 312 2583 Intemet: psocg@dcm1sg.das.net TELEX: 6503122583

Patricia Seybold's UNIX in the Office (ISSN 0887-3054) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold's Office Computing Group, 148 State Street, Suite 612, Boston, MA 02109. Second-class postage permit at Boston, MA and additional mailing offices. POSTMASTER: Send address changes to *Patricia Seybold's UNIX in the Office*, 148 State Street, Suite 612, Boston, MA 02109.

•ORACLE•

(continued from page 1) 1979, it was the first commercial RDBMS and the first commercial DBMS product to use SQL. The second RDBMS, Ingres, and the next SQL-based contender, IBM's SQL/DS, didn't hit the market for another two years. SQL compatibility catalyzed Oracle's success to a certain point, and it continues to be uppermost in Oracle's vision of the future.

Oracle once led the RDBMS pack in terms of both sales and technological enhancements. Although it still retains its sales lead by a wide margin, we have seen advancements in technology come from other vendors over the past few years. Meanwhile, Oracle clings to SQL compatibility. While this isn't necessarily bad, it has slowed Oracle's migration to the next-generation of RDBMS functionality. We originally reviewed Oracle two years ago (Oracle Version 5.1 in 1987—see Vol. 2, No. 10), and we expected more enhancements by now.

> *Although Oracle still retains its sales lead by a wide margin, we have seen advancements in technology come from other vendors over the past few years.*

In the following pages, we provide a clear picture of the product line Oracle offers. We examine both Oracle's overall architecture and strategy as well as its specific products.

Background

Oracle Corporation was founded in 1977 by Larry Ellison, president, and Bob Miner, senior vice president. The Oracle RDBMS was originally developed and introduced on a Digital PDP/11 under RSX11 and then rewritten in C and ported to the VAX/VMS environment. And Digital's success certainly proved to be an advantage for Oracle.

In recent years, though, the company has made a dramatic commitment to Unix. Most of Oracle's internal MIS development is under Unix (everyone on the development staff has a workstation connected via Ethernet to a Sequent back end), and now even some new products are released under Unix before they come out on other platforms.

STAFFING AND FINANCIALS. Oracle went public in May 1986. The company has doubled its sales for the past four years in a row—an impressive showing. Specifically, revenue for fiscal 1989 increased 107 percent to $584 million, while net income grew 90 percent to $82 million.

Oracle's staff is soaring, too. The company employs 5,500 people worldwide—that's up from a staff of only 200 five years ago, which is a phenomenal growth rate. You'd think Oracle would have trouble managing such growth, but, so far, it seems to be doing just fine.

PRODUCT POSITIONING. Oracle's product and marketing philosophy is based on three major principles: compatibility, portability, connectivity. These have been the building blocks for product development, and Oracle feels that it has made considerable improvements in one or more of these areas every year. (Generally, the company has a major release every year.)

Compatibility. SQL compatibility is the most basic tenet of Oracle's strategy. The company is committed to both ANSI and ISO standards. Oracle's implementation of SQL is compatible with and extends IBM's implementation in SQL/DS and DB2. In addition, Oracle is committed to maintaining full compatibility with DB2 application.

Portability. Oracle runs on a wide variety of hardware and operating system platforms—over 100 Unix variations, PCs, and most proprietary systems. The company recognized early on that its major customers—primarily Fortune 500 companies—had heterogeneous systems resulting in incompatibility at the hardware, operating system, and application levels. Too much time and money was spent moving applications between systems; thus, portability became a strategy for Oracle.

Connectivity. Connectivity has been a development priority since the introduction of SQL*Net in 1986. Oracle can connect different operating environments, from PCs to minis to mainframes, and it can provide portable applications across all of them. In addition, Oracle's distributed database and client/server architectures were developed out of a commitment to connectivity. Stan Tims, director of Unix marketing, also points to Oracle's "zillions" of protocols and gateways (well, not quite zillions—see "SQL*Net" below).

MARKETING STRATEGY. Oracle targets the Fortune 500, and it has built its products with these customers in mind. These are the companies most concerned with adherence to standards, and Oracle has been incredibly successful in selling its SQL-based RDBMS. Oracle is also riding Big Blue's coattails with DB2 compatibility.

While Oracle introduced distributed processing with SQL*Net in 1986, the company is now focusing on two marketing thrusts: online transaction processing (OLTP) and end-user computing (i.e., the interface, integration, and consistency among modules). It has also plugged into consulting and applications development. Oracle racked up $50 million in consulting revenues in 1989, up from nothing four years ago. The company sees consulting as particularly important to its major customers—big companies with large IBM and Digital systems. Its consulting group is now 600 strong.

The Competition. Because Oracle's revenues are more than triple those of its nearest competitor (Ingres), it has a substantially larger R&D budget (somewhere between 10 and 15 percent of its revenues). In the past, that budget was evident in enhancements made to the Oracle RDBMS. More recently, however, Oracle has been diversifying itself, and some of the important advancements in performance are coming from competitors (especially Ingres and Sybase). Tims maintains that competitors are "ratifying what [Oracle] has done all along." (He points, as an example, to the Informix Star architecture as an answer to Oracle's SQL*Star architecture.) But lately, we see just the opposite happening. To wit: Oracle's recent adoption of a multithreaded server and online backup and recovery came after people started noticing those features in Sybase. In some stages of the race, Oracle has fallen behind. For instance, it doesn't yet have fully implemented referential integrity. And Oracle could learn a trick or two from Ingres's recently announced "intelligent" server (see Vol. 4, No. 12, December 1989).

Oracle's commitment to Unix three years ago was next to nothing, but today, Unix is a strategic operating environment.

The Unix Market. In the past few years, Oracle has really sat up and taken notice of the Unix market. Tims maintains that Oracle's commitment to Unix three years ago was next to nothing, but today, Unix is a strategic operating environment.

Revenues from Unix sales are now over $165 million, up from only $1 to $2 million a few years ago. Unix revenues break down as follows:

- Direct sales (end users) 70-75 percent
- OEMs 15-20 percent
- VARs 5-10 percent

Product Line

Oracle is an SQL-based RDBMS plus a set of integrated software tools for users and application developers. Applications development tools include forms generation and interfaces to most major programming languages. Oracle offers a Lotus-like spreadsheet and, most recently, integrated electronic mail and a graphics system that runs on bit-mapped screens. And, much to the chagrin of some of its VARs, Oracle has recently plunged into the applications market (see "Applications" below). The database kernel and tools are each licensed separately. Thus, customers can mix and match.

CORE PRODUCTS. The Oracle RDBMS, the back-end database management kernel, is the core of the Oracle product line. The latest release is Version 6, which came out in 1988, and it offers an optional OLTP package. (Yes, optional; you have to pay for the extra performance.) New modules bundled with Version 6 are SQL*DBA, a database administration utility that includes diagnostic monitoring, an archive log, and recovery functions; and SQL*Loader, an improved loading utility.

The RDBMS has integrated core products as well. SQL*Plus provides a command-driven interface to the RDBMS for creating and maintaining the database and for ad hoc queries and reports. SQL*Forms is a forms-generation tool which features application triggers. Both SQL*Plus and SQL*Forms have been available in one form or another since Oracle was first introduced.

Host Language Hooks. Oracle supports two levels of interface with traditional third-generation programming languages (3GLs): Precompilers (Pro*C, Pro*Cobol, etc.), which allow embedded SQL in programs written in C, Cobol (in various forms), Fortran, PL/1, Pascal, and Ada; and a predefined set of subroutine calls (Pro*SQL) as a host-language interface that can be used with any programming language. Pro*SQL provides lower level calls to access data in an Oracle database directly, rather than through SQL.

DEVELOPMENT TOOLS. In addition to SQL*Forms and SQL*Plus, Oracle offers applications developers SQL*Menu and CASE tools. SQL*Menu lets you create customized menu interfaces for applications. The CASE tools (Dictionary, Design, and Generator) let you design an application from scratch—a graphical "fill in the form" approach to describing and generating an application. However, CASE tools are not widely available under Unix. (See Illustration 5.)

END-USER TOOLS. Several modules have been designed for the end user. These include: SQL*Calc, a spreadsheet; SQL*Graph, business graphics; Oracle Graphics, a high-end graphics package; SQL*QMX, a query manager and reporting tool; and Oracle Mail.

SQL*Calc. SQL*Calc is a Lotus-like spreadsheet with the ability to embed SQL statements in cells. With SQL, the user can extract data from an Oracle database and enter it into a range of cells on the spreadsheet. The SQL statement can refer to another cell on the spreadsheet, enabling the user to change the value of variables and extract a different set of data from the database. The user can also create tables and enter data into tables from the spreadsheet.

Oracle*Graphics. Oracle also has an optional business graphics package called SQL*Graph. However, the company is downplaying SQL*Graph in deference to the newer, slicker Oracle*Graphics.

Oracle*Graphics is a chart and graph generator as well as a graphic design tool. It supports not only Oracle data, but DB2 and SQL/DS databases, and data imported from spreadsheets (Lotus, Excel, and dBase). With it, you can create and edit (e.g., customize, resize, rotate, group, layer, etc.) charts and graphics. The context-sensitive, hypertext-like Help implementation guides you smoothly around the system. Oracle*Graphics uses pull-down menus, dialog boxes, and artists' palettes via a mouse. You can also use command keys, once you've developed some expertise with the system. However, menus don't include corresponding command keys (so developing that expertise might take a while). They should.

Oracle*Graphics only runs on Sun computers under Unix thus far. Ports to Hewlett-Packard (HP) and Sequent machines are scheduled for midyear.

SQL*QMX. Oracle recently developed a dynamic query manager and reporting tool called SQL*QMX. Essentially, SQL*QMX simplifies querying for novice users by providing Query-by-Example (QBE). It's similar to (and compatible with) IBM's Query Management Facility (QMF). Users indicate which columns they want displayed or sorted within the pictorial representation of a table. They can retrieve data on specified conditions, as well as insert, update, delete rows, and even perform table joins. Furthermore, users can convert QBE queries into SQL statements via a command within SQL*QMX. Oracle maintains that, in time, frequent SQL*QMX users will recognize and understand SQL syntax, so it's almost a learning vehicle.

The advantage of Oracle applications is that they are modular. The interface is consistent across all modules, and modules integrate with each other and with existing applications.

SQL*QMX is limited under Unix, currently running on certain Sequent, HP, and AT&T products in addition to VAX/VMS, VM, MVS, DOS, and AOS/VS. Additional Unix ports are on the agenda.

Oracle*Mail. Oracle's mail system, which is built on top of SQL*Star, is implemented well. Oracle*Mail is consistent and portable among different hardware platforms and operating systems. Communications among heterogeneous systems are transparent; a Unix user can send a message to a user in an MVS environment and be completely oblivious of user locations, operating systems, or networks. In addition, Oracle*Mail interfaces to Unix and VMS mail via standard communications gateways (e.g., X.400).

Oracle*Mail's interface is straightforward. Pull-down menus guide you through all the mail functions. A nice feature is that messages are customizable. If, for instance, you're sending a phone message, you choose "phone message" from a menu, and the system automatically configures the message accordingly. Oracle*Mail also has some extra productivity and flexibility features. You can organize and cross reference your mail into various folders. You can also integrate applications; forms, reports, batch programs can be exchanged electronically. A sales tracking application, for example, can automatically notify key managers if sales fall below forecast. Or routine reports can be automatically distributed to a designated mailing list. This is the beginnings of procedural automation.

Applications. We mentioned that Oracle is branching into application software, particularly financial applications and manufacturing automation. The advantage of Oracle applications is that they are modular, flexible systems built on a relational database. The interface is consistent across all modules, and modules are designed to integrate with each other and with existing applications. Modules can also be customized to meet specific needs of users. The Oracle Financials family of products includes Receivables, Revenue Accounting, General Ledger, Payables, Purchasing, and Assets. Oracle Core Manufacturing is a material control and tracking system for parts, supplies, tools, resources, etc. It includes Inventory, Bill of Materials, Work in Process, Master Schedule, and MRP.

In its quest for application software, Oracle has been competing with—and in fact losing—some of its VARs. The company maintains that VARs are still crucial to its marketing strategy. It's just that Oracle has shifted its partnering focus to companies like Quadratron and Uniplex—office systems vendors. Meanwhile, manufacturing and financial software vendors are picking up other RDBMS companies. It's a trade-off. Application development is lucrative, but it cuts off a marketing avenue. Obviously, Oracle believes it has made the right choice.

Architecture

Version 6 was adjusted architecturally to facilitate transaction processing. All the same, its previous client/server architecture, which separates the processing of its DBMS into a front-end user process and a back-end kernel process, was well-suited for the adjustment, so that stayed intact. The architecture has a number of features worth mentioning:

- An array interface, which reduces the number of system calls, thus making data transfer faster and more efficient. The system sends a package of system calls (the number in the package is predetermined by the systems administrator) rather than one at a time.

- Shared memory, which also improves performance. Rather than moving data around in the system from process to process, the processes exchange semaphores (messages) identifying where the data is in memory. When a user issues a request for data, Oracle first checks in shared memory to see if the data is already there. If so, Oracle need not access the disk. In addition, users share a single image of the database kernel, instead of storing a duplicate copy in memory for each user.

- Write-through cache, which helps in data integrity, though it is by no means a complete solution. Write-through caching flags changed (committed) data in the Unix buffer cache and forces the data to be written to the disk immediately. While this actually slows performance down a little, the company considers it an absolute requirement for data integrity.

DISTRIBUTED ARCHITECTURE. Oracle's client/server architecture is also well-suited to its distributed environment. We are disappointed, however, that Oracle's distributed database scheme hasn't changed much. While it has added features for fault tolerance and performance, the company has stubbornly avoided the implementation of server-side referential integrity, reasoning that doing so goes against its commitment to ANSI standards. That may be true, but Sybase and Ingres have implemented very workable solutions nonetheless. Centralized data validity is essential for mission-critical, online, transaction-oriented distributed environments. It's simply too difficult to control what version of an application runs on a workstation and to ensure that every developer has implemented data protection rules correctly. Oracle promises database-enforced integrity for Release 7, which is scheduled for release later this year.

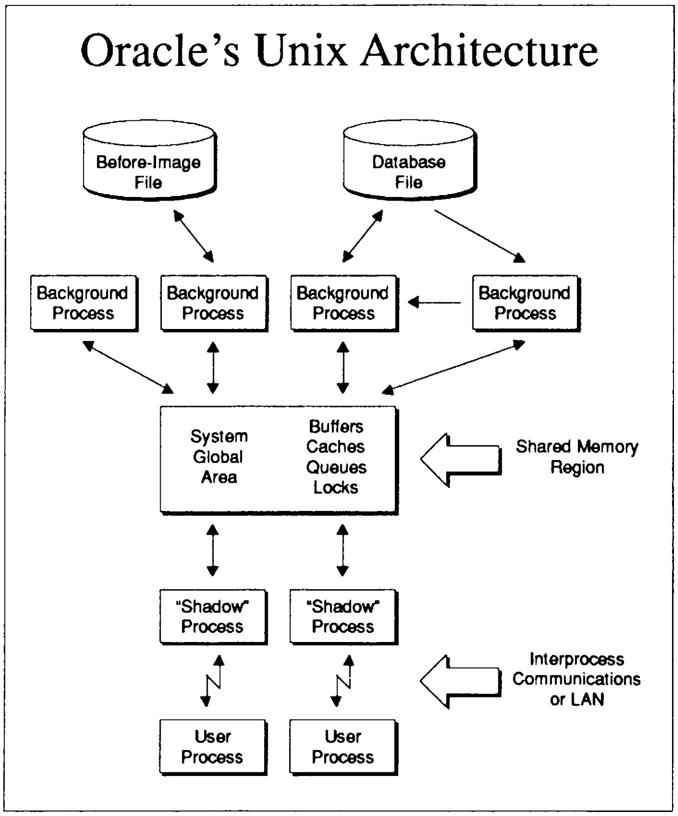


Illustration 1. Oracle uses Unix shared memory for all database locks, buffers, caches, and queues. Oracle also uses shared text; only one copy of Oracle needs to be loaded into memory.

Oracle is also missing the ability to update multiple databases in a single transaction. Most RDBMSs handle transaction management with a two-phase commit. It works like this: The node controlling the transaction instructs each remote node on the operation it is to perform. As each participating node is ready to commit, it sends a message to the originating node. Once all nodes have given their assents, the originating node sends out messages to commit the transaction. If any node fails, the transaction is rolled back.

When last we looked at Oracle (Version 5.1), we were told that a distributed transaction management implementation would be forthcoming in Release 6. It's now slated for Release 7. Likewise, Oracle hasn't yet implemented partitioned database tables, in which the rows of a single table can be distributed across different nodes. Oracle is behind here; many consider partitioned tables a first step in the implementation of distributed databases.

SQL*Star. SQL*Star is Oracle's distributed environment. It provides two major characteristics: location transparency and the ability to access multiple databases on different machines in a single query (i.e., joins and unions across network nodes).

SQL*Star comprises two products: SQL*Net and SQL*Connect.

- SQL*Net, Oracle's basis for distributed processing, lets applications access remote Oracle databases. SQL*Net protocols support several different networks: DECnet, TCP/IP, async, and 3270. New protocols under Unix include support for LU6.2, synchronous, and Manufacturing Automation Protocol (MAP).

- SQL*Connect, which has been added since Version 5.1, is Oracle's gateway to non-Oracle databases. Users can log onto a non-Oracle database running on a remote host and issue read-only SQL commands against the foreign database. SQL*Connect supports access to Digital's RMS and IBM's two SQL-based DBMSs, DB2 and SQL/DS. Subsequent releases of SQL*Connect will feature access to non-SQL-based datafiles such as IMS and VSAM.

More Interoperability? While SQL*Connect makes some headway into interoperability, we put on our user's hat and asked why Oracle doesn't support access to other major Unix RDBMSs. The company doesn't see this as being an appropriate move. But its customers (and potential customers) do. After all, user organizations typically have a mishmash of hardware and software. They want to be able to mix and match various databases. Oracle isn't alone in its stance on this issue. But we think it's time for software companies to stop tree-hugging and give users what they need: more interoperability.

The newly-formed SQL Access Group may offer some solutions here. The group's goal is to define a set of specifications based on the existing SQL standard so that one vendor's SQL application can access other vendors' SQL-based DBMS servers. (See News Analysis, page 21.)

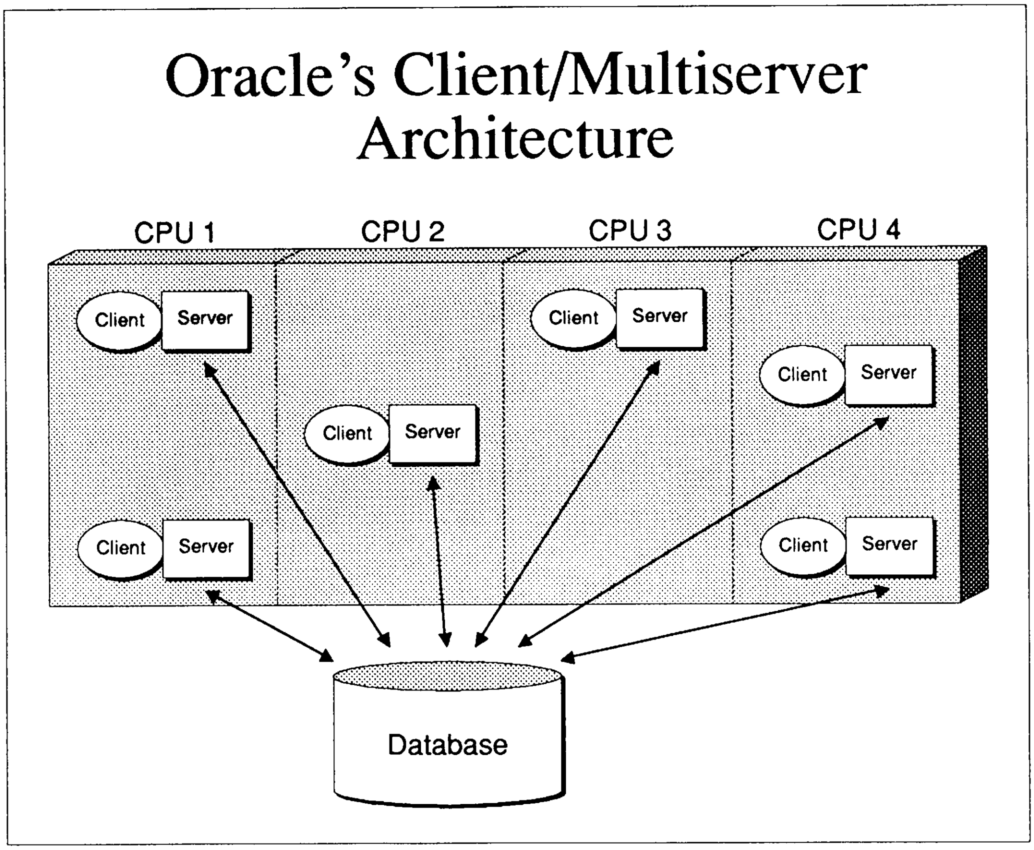


Illustration 2. Oracle's client/multiserver architecture optimizes symmetric multiprocessors such as IBM's 3090, Digital's VAXcluster, Sequent, and Pyramid. Logically, a server is dedicated to each user and is responsible for all database activities. While each client logically has its own server, physically, all clients share the same server code.

Server Strategy. As an answer to the demand for multivendor hardware support, Oracle has developed three server options: Unix 386, OS/2, and Banyan VINES. These allow PCs, Macintoshes, and Unix boxes to act as front ends on a distributed Oracle network. Oracle must be installed on each user workstation on the LAN to access the database server. You connect to the servers using standard connectivity software (e.g., LU6.2, TCP/IP, async, and 3270). With the Unix server, the LAN options are Novell NetWare and 3Com 3+Open.

In addition to the servers, Oracle has developed add-ons for popular PC and Macintosh applications, including dBase, Lotus, and HyperCard, which let you access Oracle from within the application.

TRANSACTION PROCESSING OPTION. OLTP functionality was introduced with Version 6. Other vendors offer OLTP functionality with their core products. Of course, Oracle has always sold its products in pieces, but transaction processing is quickly becoming essential to distributed database schemes. The company should rethink its position here, and stop making customers pay extra for fundamental technology.

Oracle has zeroed in on what it sees as two trends in the relational database market: the need for OLTP, complete with high performance and fault tolerance with relational productivity; and the need for enterprise-wide computing, where users have instant access to information on distributed mainframes, minis, and PCs. In other words, Oracle is seeking to provide OLTP in the context of enterprise-wide computing. To that end, Oracle developed a new OLTP architecture.

A new programming language, PL/SQL, is also available with the transaction processing option. PL/SQL isn't quite a 4GL; Oracle prefers to call it a transaction processing language or a procedural language extension to SQL. Used in conjunction with SQL*Forms, it allows an application to package multiple transactions, send the package for processing, and receive back the appropriate rows of data.

Multiprocessing. Oracle developed a client/multiserver model for multiprocessing (see Illustration 2). Theoretically, if there are enough processors to go around, each user would get a CPU to bash against. All of the processors would access shared memory, cached tables, and programs. The architecture delivers scalability by providing incremental throughput as you add CPUs. Version 6 is also multithreaded, allowing multiple requests per server simultaneously. The multiserver architecture also features dynamic adjustments—servers can be dropped or added as system loads change, thereby improving throughput.

Row-Level Locking. Oracle implemented a row-level locking mechanism that increases multiuser performance. In essence, the DBMS locks a single row instead of an entire block of memory. It's an issue of granularity; more users can access data. You can retrieve, update, and delete data from a specific row in a table, leaving other parts of the table open for updates. You only have to wait if you're attempting to update a row that another user is currently updating.

For read consistency, Oracle has a "snapshot model," which allows query to read rows without locks. As a row is being updated, the database records a pre-update snapshot of the row in the roll-back segment of memory. A user querying the database will see that snapshot, and thus will see the proper information. This snapshot model enforces the following conditions:

Oracle is in the process of integrating its products with a cohesive, consistent user interface across all components. However, it seems to be slow going.

- Queries do not block queries.
- Queries do not block updates.
- Updates do not block queries.

Oracle also talks about "piggy-backed" writes, a feature that also helps performance. If the database is hit with multiple transactions, it processes all of them together, rather than serving each in turn.

Redo Logging. With Version 6, Oracle implemented an online database backup. Oracle uses a redo log file to guarantee that committed transactions remain permanent. Each transaction is recorded in the log file, and the transaction is not committed until the changes have been written to it.

If the system fails, Oracle can roll forward and roll back transactions. The database is recovered using the online redo log files. An archive process copies full redo log files to tape. These archived logs contain a complete history of all the changes made since the last backup.

The redo log writer is also responsible for writing multiple transactions so they can commit simultaneously.

Using Oracle

We evaluated Oracle Version 6.0 on an NCR Tower 32 under Unix System V. Version 6 requires approximately .5MB of memory per user (it varies according to which modules you have in place), 5 to 15MB of disk storage space, and support for shared memory.

USER INTERFACE. Evaluating Oracle's current user interface situation was difficult. It's actually a user interface in transition. As yet, no "umbrella" menu exists that encompasses all the Oracle modules. Each module is invoked separately from the Unix prompt. Your username and password are required each time you call an Oracle module. (Your Oracle login can be made part of your Unix login to avoid this problem.)

Oracle is in the process of integrating its products with a cohesive, consistent user interface across all components. However, it seems to be slow going. Oracle was in the same process when we reviewed it two years ago.

In the past, Oracle has pushed Easy*SQL as a menu-driven user interface, but Easy*SQL is being phased out (it's available only on demand with Version 6) and a more graphical-oriented interface will be phased in. Oracle has designed its own portable graphical user interface layer—internally called OraKit—that is slowly being moved to all of Oracle's products. It will provide users with the same look and feel across its entire product line. Some components have it already—Oracle*Mail, Oracle*Graphics, SQL*QMX, SQL*Menu, SQL*ReportWriter, SQL*Forms BMI, and Oracle's CASE, financial, and manufacturing products. The word from Oracle is that any new product will sport this interface, and remaining products will eventually adopt it. In time, Oracle plans to support Motif, Open Look, and Presentation Manager interfaces. (In other words, an Oracle application will automatically be portable to any of these interfaces.) SQL*Forms will be one of the first products to be released with the Motif interface (time frame: midyear). Meanwhile, Oracle is pushing SQL*Menu as a means to create a custom menu environment for the user.

Help. Oracle's Help function is also in a state of disarray and varies from product to product. A new context-sensitive, windowed Help is available for some tools (i.e., it's part of the new interface), while others still suffer from mediocrity.

Drawbacks of the current Help function:

- You can't scroll freely through Help text.

- You can only access Help on one topic at a time.

- SQL*Forms has no Help function at all—only a screen of function keys, but that's not Help. (And, on our system, a keyboard-mapping problem got us stuck in a field. However, a new interface is due for SQL*Forms in February.)

- Error messages are sometimes vague. For example, if you enter a duplicate value in a field where the value must be unique, you are told, "Oracle error occurred while trying to update record." That could mean a lot of things; we prefer something more precise.

SQL*Plus. SQL*Plus is still the primary interface to the Oracle RDBMS on the Unix system. This interactive, command-driven interface is the only way to create and modify tables, create indexes, grant access permissions to the database, etc. SQL*Plus consists of SQL commands and extensions to SQL—fittingly called SQL*Plus commands—for formatting results, setting options, and editing and saving command files.

Commands are entered at the SQL> prompt, and current commands are stored in a buffer for editing (either with a rudimentary line editor or the Unix system editor).

There is a degree of inconsistency between SQL commands and SQL*Plus commands. For example, in our last review of Oracle, we found it annoying that SQL*Plus commands were not automatically stored in the buffer. This is still the case. If you make a mistake, you cannot just edit the commands; you have to reenter the entire command. In addition, whereas SQL commands can span multiple lines, SQL*Plus commands are limited to single lines. Another loose end (which may only exist in the NCR port) is that some table/view names demand uppercase to run a query, even though the documentation states that object names are case insensitive.

*Oracle is promoting SQL*Menu, its fourth-generation menu development component, as an integrated front end to other RDBMS tools.*

Although SQL*Plus is command driven, we found it relatively easy to learn and use. Developers wouldn't have a problem with it. One nitpicky complaint is that, if you misspell the first word of a command (e.g., enter "selct" instead of "select") and try to execute it, Oracle rebuffs you with an error message: "unknown command 'selct'—rest of line ignored". Then it's gone. The command doesn't stay in the buffer for editing; you must reenter it. Oracle explained that, if the first word of the command is misspelled, the system doesn't recognize that it's SQL, so it trashes the statement. However, it can be annoying after you've just typed in a long, multilined command.

SQL*Menu. Oracle is promoting SQL*Menu as an integrated front end to other RDBMS tools. SQL*Menu is Oracle's fourth-generation menu development component. It's made up of six objects: applications, menus, menu items, substitution parameters, procedures, and variables. (These days, it's always a good idea to find out what people mean by "object." In this case, Oracle refers to a type of data in the database that you can copy, move, or delete as one entity in a single operation. An object can contain—or be contained by—other objects.)

SQL*Menu can be linked with SQL*Forms and SQL*Plus to make an integrated package. In doing so, you can invoke either a form or report from within SQL*Menu by calling SQL*Forms or SQL*Plus as subroutines. In addition, SQL*Menu preserves any SQL*Forms global variables, which, in essence, means that applications built by using SQL*forms will be able to use SQL*Menu as a front end. Oracle recently added a menu macro command into SQL*Menu, so that designers can reduce to a minimum the number of menus that a user must wade through.

Documentation. We like the Oracle documentation. The manuals are well written, well illustrated (with nice, multicolor screen shots), clear, straightforward, and easily understood by a new user. Every now and again, they're even entertaining; some of these writers have a sense of humor.

The documentation comes in the form of separate booklets. Each module has at least one and possibly up to three or more of these—for example, SQL*Forms has a Designer's Tutorial, a Designer's Reference, and an Operator's Guide. In addition, each module has a quick reference card. If the documentation came bound in plastic rings or were three-hole-punched so they would lay flat while you were keyboarding, they'd be perfect.

DATABASE DESIGN. Oracle has a single-database design. Tables are not stored as individual files, but rather as part of one database. Thus, there is no concept of creating separate databases for different applications (e.g., a customer database, an inventory database, an employee database), and you need not create a database in Oracle before you can create tables. When Oracle is installed, a standard database file is created called oracle.dbs. When you create tables, they are all part of this database.

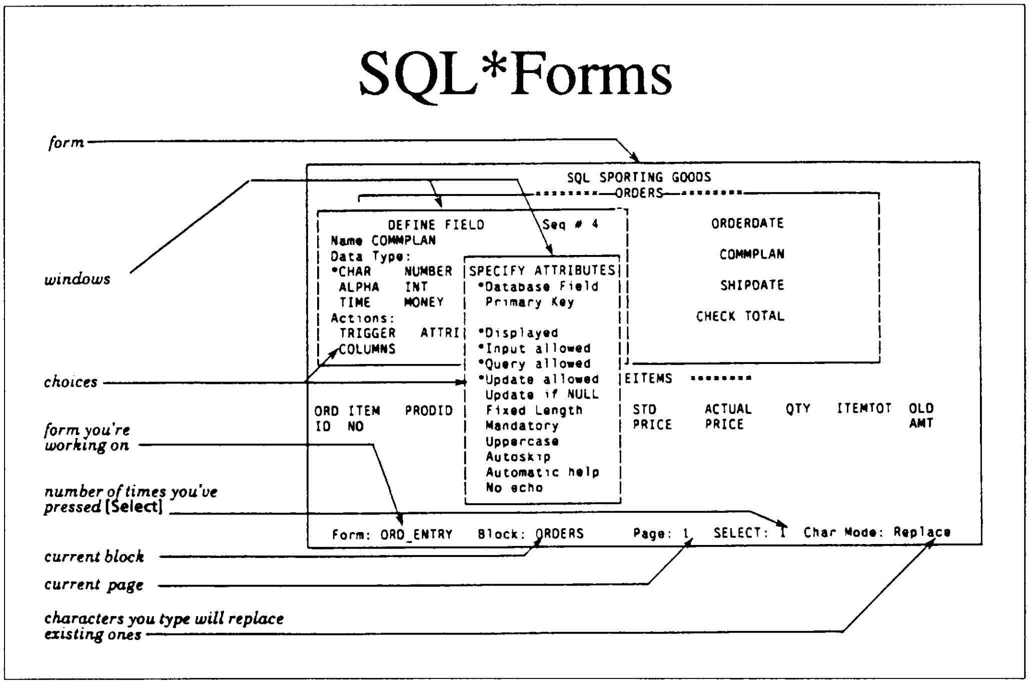
The advantage of a single-database design is speed; you by-pass the Unix file manager. The disadvantages involve recovery and flexibility; you need to recover the whole database even if only part of it becomes corrupted. To alleviate this disadvantage, you can create additional databases. Oracle also offers "table spaces" as a way to segment the databasc. An administrator can take a table space on or off line at will, while the rest of the database stays online. Table spaces allow you to, for example, store specific tables in the database on different disks, or make a portion of the database unavailable for updates.

Oracle stores all information about tables, columns, views, indexes, users, access privileges, and storage allocations in the data dictionary.

Creating Tables. Table creation hasn't changed since our last evaluation of Oracle. You use the "create table" SQL command in SQL*Plus, enter the table name and each column name, and enter the type and length (if appropriate).

Oracle supports five generic data types: char and varchar (up to 240 characters), numeric (up to 38 digits), date, long char (allowing a column to be 64KB in length), and raw. (We were somewhat disappointed that Oracle made no mention of plans to support image or user-defined data types.)

SQL*Forms lets you refine your definition of character and numeric data types. You can specify alpha (only spaces and lower/upper-case letters), integer (no decimals), money, and a number of other restrictions. Unfortunately, these only apply to


SQL*Forms
form
windows
choices
form you're working on
number of times you've pressed [Select]
current block
current page
characters you type will replace existing ones

```
                     SQL SPORTING GOODS
            ********——ORDERS——********
  DEFINE FIELD        Seq # 4
 Name COMMPLAN
 Data Type:
 *CHAR    NUMBER  |SPECIFY ATTRIBUTES|      ORDERDATE
  ALPHA   INT     | *Database Field  |
  TIME    MONEY   |  Primary Key     |      COMMPLAN
 Actions:         |                  |
  TRIGGER   ATTRI | *Displayed       |      SHIPDATE
  COLUMNS         | *Input allowed   |
                  | *Query allowed   |      CHECK TOTAL
                  | *Update allowed  |EITEMS  ********
                  |  Update if NULL  |
ORD ITEM   PRODID |  Fixed Length    | STD     ACTUAL    QTY   ITEMTOT  OLD
ID  NO            |  Mandatory       | PRICE   PRICE                    AMT
                  |  Uppercase       |
                  |  Autoskip        |
                  |  Automatic help  |
                  |  No echo         |
 Form: ORD_ENTRY   Block: ORDERS      Page: 1   SELECT: I   Char Mode: Replace
```

*Illustration 3. SQL*Forms screen layout for character-based terminals. Forms designed with SQL*Forms can be used to enter and edit data, perform queries, and build complex applications. A graphical Motif-based version is due in next month.*

data entered on the form, not data entered with SQL statements in SQL*Plus.

Oracle supports the concept of nulls, or unknown values, and any column in a table can be defined as null or not null. Oracle can display null values with a specified value, such as "N/A." Oracle also supports views, which make it possible to manipulate the logical structure of the application while hiding the physical representation of data. Views can be queried, and if the viewer has permission, the underlying table can be changed through the view.

Modifying Table Structure. As is the case with most RDBMSs, you can't change the table structure once it's been defined except with the "alter table" SQL command. In Oracle, the only way to delete or rename columns is to create a new table from the old one, using a nested query to select those columns desired from the old table. To add a non-null column to a table after rows have already been inserted in the table, you add the column as accepting nulls, fill in values for the existing rows, and then modify the column to non-null status.

There ought to be an easier way. We'd appreciate some additional options for table modification—and not just from Oracle. Most RDBMSs are found lacking in this. It's not easy designing database tables, and you'll be hard-pressed to get it right the first time.

FORMS. Forms designed with the SQL*Forms module can be used to enter and edit data, perform queries, and build complex applications (see Illustration 3). The current implementation of SQL*Forms hasn't changed much since our last review, either. But that's okay. We were impressed with its power and flexibility then, and we are now.

A form is made up of an unlimited number of "blocks." Each block represents data from a single table and can display multiple rows of data. Pre- and post-processing triggers can be added at the field, block, and form level for additional validation and database processing. A trigger can be an SQL command, a function-key macro, or a 3GL program. They can be chained together for sophisticated routines.

Menus, fill-in-boxes, and overlapping windows guide you through forms and application design, so it's not difficult to create a form. After you identify the appropriate blocks and fields, SQL*Forms generates a default form, which you can modify and customize with a screen painter. It's character-based, so the screen-painter isn't exactly slick or intuitive. But once you get the hang of the function keys, it's not difficult.

Defining Fields. You define field attributes and validation criteria for each field simply by selecting items on a list in a window. Likewise, you can define a default value and a customized help message. Data validation can be very complex and sophisticated, providing referential integrity in the application, a list of valid values (from another table), and a range of values. You can also connect master records to detail records on a multitable form.

Defining Blocks. At the block level, the forms designer can specify options, such as the sequence number for blocks in a multitable form and default ordering ("where" and/or "order by" clauses that apply to all queries for the block).

New Interface. Because of the innate richness of SQL*Forms, it naturally takes some time and effort to master. Once you start creating intricate applications, it gets a little mind-boggling. Triggers involve programming and can be especially tricky. They demand a clear understanding of the objectives, implications, and appropriate timing of trigger steps. Oracle has done a decent job on the current user interface with the incorporation of menus and overlapping windows. We have some interface complaints concerning the version of SQL*Forms we were working on—e.g., inflexible keystroke requirements and an unorganized menu structure.

However, Oracle has released a bit-mapped interface for SQL*Forms for Sun and Ultrix workstations. And X- and Motif-based versions will be available in February. The mouse-driven, icon-based interface is much better suited for application and forms design. The interface uses windows, menus, and buttons that match the standard functions found in the traditional SQL*Forms product.

DATA ENTRY AND EDITING. You can enter and edit data in a table using either SQL*Plus (SQL commands) or SQL*Forms. We recommend using SQL*Forms.

SQL*Plus. SQL doesn't let you add multiple records to a table with one command (although in SQL*Plus you can use parameters to enter one record after another without having to reenter the entire SQL command). Furthermore, validity checking is limited. Oracle ensures that data entered is the right data type and length, that all mandatory (not null) fields are entered, and that uniqueness is maintained where specified in the index. But that's it. Other validity checks, such as ranges for a value or a list of acceptable values, are not available. Oracle recognizes this problem, and told us that a "with check" option helps to some degree by extending default data checks. Additional solutions are under development.

*Because of the innate richness of SQL*Forms, it naturally takes some time and effort to master. Once you start creating intricate applications, it gets a little mind-boggling.*

SQL*Forms. SQL*Forms is a better approach for entering and editing data. As we mentioned, on a form, you can include extensive validity checks for entering and updating data. However, these checks do not affect data entered via SQL commands. Since Oracle has no central repository of data validation and integrity rules, they must be included at the application level (in a form created with SQL*Forms or in a program written in PL/SQL or a 3GL).

External Files. The SQL*Loader can read external (non-Oracle) flat files in a sequential, fixed- or variable-length format. This utility is designed for a developer or, at least, a very experienced user.

INDEXING. Indexing in Oracle is straightforward and flexible. Once created, an index is automatically maintained. The number of indexes for a table is unlimited, and indexes can be created or dropped at any time. The index key can be based on multiple columns but cannot be longer than 240 characters.

The existence of indexes is transparent. Oracle automatically keeps track of what indexes exist and uses them as necessary in optimizing query-processing. The user never needs to specify the use of an index when creating a query statement.

QUERIES. Queries are generated using SQL*Forms (for Query by Example) and SQL*Plus. Unfortunately, SQL*QMX is not yet widely available under Unix.

SQL*Forms. An SQL*Form can be used to query by example. You display the form, hit the "query" function key, and enter your criteria. You can enter an exact match or an SQL "where" statement in any of the fields. SQL*Forms does not tell you how many records it retrieved unless you hit a function key (of course, you can always scroll through all of them to get to the final count).

SQL*Plus. In SQL*Plus, you query the database using the standard SQL "select" command. SQL*Plus tells you how many rows were retrieved in a query. As we have mentioned, you can edit the current query (the one stored in the SQL buffer), and save and retrieve repetitive queries. Stored queries can be executed at the time you enter SQL*Plus or at any time from within SQL*Plus with the "start" command. Stored queries can also include variables, or parameters, that can be entered by the user when the query is run. This is particularly helpful when, for instance, you need to add several new records to the database, or when you wish to see the results of different sets of values in the "where" clause.

The "column" command provides customized formatting and headings for a column in a list of query results. A column command is independent of the table; the command applies to all instances of that column name, even if it appears in multiple tables. This command stays in effect until you exit from SQL*Plus or change the column characteristics. To avoid having to reenter column commands over and over, you can store them in a file and have the file executed automatically at the beginning of each SQL*Plus session. One feature we would have liked but couldn't find anywhere is the ability to list the current settings for column headings and other parameters in SQL*Plus—a status screen of some sort.

The number of tables that can be joined in a single query is unlimited. And tables can be joined based on criteria other than exact matches. Oracle supports outer joins.

In addition, Oracle has a "soundex function," which accommodates spelling variations (e.g., Alan, Allan, Allen, and Allyn). It can also interpret data in tables that contain hierarchical relationships.

REPORTS. Oracle's premiere reporting tool, the SQL*ReportWriter, has a menu-driven, forms-based interface. This is a step up from Oracle's previous report writer, RPT/RPF (it's still available, by the way), which, though flexible enough, requires that you write a program to generate a

(continued on page 14)

Oracle Features Chart

ARCHITECTURE	
Client/server	Yes
Multithreaded	Yes
Support for symmetric multiprocessing	Yes
Open Architecture (APIs available)	No
DATABASE PARAMETERS	
Databases/server	Unlimited
Tables/database	Unlimited
Fields/record	254
Record size	126KB
USER INTERFACE	
Menu bypass	Yes
Contextual help	Yes
Online tutorial	No
Documentation	Yes
Ability to customize standard menus	Yes (via SQL*Menu)
DATA TYPES	
Character	char and varchar (240 character max.)
Integer	Yes
Float	Yes
Currency	No
Date/time	Yes
Binary	Yes
Long text	Yes
Image	No
Support for arrays	No
User-defined data types	No
User-defined functions and operators	No
B-TREE INDEXING	Yes
Maximum number of indexes	Unlimited
Maximum number of fields/index	16
Maximum size of index key	Unlimited
Order options	Ascend/descend
Unique index	Yes
Clustered index	Yes
Other file access methods (hash, etc.)	None
SCREEN FORMS	SQL*Forms
Default form generator	Yes
Customized	Yes
Multiple tables/form	Yes
Multiple screens/form	Yes
Embedded processing (if-then-else, display aggregates)	Yes, via triggers

FIELD ATTRIBUTES ON FORMS	
Case conversion	Yes
Default value	Yes
Required field	Yes
Acceptable values	Yes
Verification (enter data twice)	Yes
Formatting of data	Yes
Calculated fields	Yes
Display only (no entry/update)	Yes
Hidden	Yes
Prompt (for data entry)	Yes (at bottom of screen)
Error message	Yes
Customized help	Yes
Video display	Yes
Ability to change field attributes dynamically	No
QUERY-BY-FORMS	
Exact match	Yes
Relational operators	Yes
Ranges	Yes (through SQL)
List of values	Yes
Wildcards	Yes
Maximum/minimum values	Yes
Print query results	Yes
Pass results to writer	Yes (through host command)
Text search	No
SQL	
Standard SQL statements	
Data definition language (DDL)	Yes
Data manipulation language (DML)	Yes
Query language	Yes
Extensions to SQL	
Commit/rollback transactions	Yes
Execute operating system commands	Yes
Load/unload data to/from file	Yes (via SQL*Loader)
Additional data definition statements	Yes
Control-of-flow logic	Yes
Can be embedded in C/Cobol programs	Yes
Can create new table with query results	Yes
Stored queries	Yes (via PL/SQL)
Case-insensitive (e.g., field names)	Yes
Can call C routines	No (Yes in tools)
How to create SQL queries/statements	SQL*Plus and SQL*Forms
Query optimizer	Yes
Syntax independent	Yes

UNDERLYING FILE STRUCTURE	Unix
REPORT WRITER	SQL*Plus, RPT/RPF, SQL*ReportWriter
Nonprocedural	Yes
Default report generator using screen forms	Yes
Interactive debugging	No
Input source	Query
Multiple tables	Yes
Page formatting	Yes
Headers and footers	Yes
Data formatting	Yes
Sort data	Yes
Aggregate functions	Yes
Logical processing (if-then-else logic)	Yes
User variables	Yes
Prompt for input variables at run time	Yes
APPLICATION GENERATOR	SQL*Forms/ SQL*Menu
Ability to design application menus	Yes (SQL*Menu)
Default menu generator	Yes (SQL*Forms)
Custom help	Yes (SQL*Forms)
ABILITY TO CREATE VIEWS	Yes
TRANSACTIONS	
Logging	Yes
Commit/rollback transaction	Yes
Roll forward	Yes
INTEGRITY	
Referential integrity (how implemented: primary/foreign keys, triggers)	No
Field validation	Yes
Support for business rules	No (at application level only through SQL*Forms)
STORED PROCEDURES	No
CONCURRENCY CONTROL—LOCKING LEVELS	
Database	Yes
Table	Yes (optional)
Record	Yes
Other data isolation levels	None

DATABASE SECURITY	
Login password	Yes
Multilevel access control	Yes
User	Yes
Group	Only with SQL*Forms/ SQL*Menu
Application	Only with SQL*Forms/ SQL*Menu
Database-level access	Yes
Table-level access	Yes
Record-level access	Yes
Field-level access	Yes
Access by time of day	Yes
Access by location (workstation)	Yes
Ability to define resource limits on user queries	No
AVAILABILITY	
Online backup	Yes
Online database changes	Yes
RAW INPUT/OUTPUT	Yes
I/O REDUCTION TECHNIQUES	Fast commit, group commit, asynch I/O
DATABASE CAN SPAN MULTIPLE PHYSICAL DEVICES	Yes
NETWORK SUPPORT	TCP/IP, asynch, LU6.2, DECNet, NetWare
DISTRIBUTED DATABASE CAPABILITIES	
Location transparency	Yes
Distributed query processing	No (scheduled for V.7)
Distributed query optimizer	Yes
Distributed transaction processing	No (scheduled for V.7)
Support for data replication	No (scheduled for V.7)
Access to heterogeneous databases	Yes (via SQL*Connect: DB2, SQL/DS, RMS)
INTERNATIONAL LANGUAGE SUPPORT	Yes
Sorting sequences	Yes
Error messages	Yes
2-byte character set	Yes

(continued from page 11) report. While mastering SQL*ReportWriter takes some time, it offers a good degree of flexibility. For instance, it allows for multiple queries, and it bolsters Oracle's distributed architecture by allowing applications to access data from different hardware platforms and operating systems.

SQL*Plus. SQL*Plus also offers some relatively easy formatting commands for simple reporting requirements. In addition to the "column" command, you can define "break on" and "compute sum" commands to calculate totals and subtotals on one or more columns. SQL*Plus reports provide centered page-headings and automatic page-numbering and date.

CUSTOMIZING APPLICATIONS. Oracle's primary tools for application development are SQL*Forms and SQL*Menu. The triggers in SQL*Forms let a developer write applications without really programming. However, as we have seen, triggers are, essentially, a programming language. And, since they're so sophisticated and complex, the company offers training classes to help developers master them. PL/SQL provides the 3GL functionality missing in SQL*Forms, so if you purchase the transaction processing option, Oracle gives you a 4GL-like environment.

DATABASE SECURITY. We mentioned earlier ("User Interface") that a username and password ensure that an individual is an authorized Oracle user. Once in the DBMS, Oracle provides several levels of data security using the standard "grant" and "revoke" commands.

The database administrator issues each user one of three levels of access:

- Connect access lets you log onto Oracle, query or update tables for which access has been granted, and create views; a "connect" user cannot create tables or indexes.
- Resource access includes connect privileges plus allowing you to create tables and indexes and to grant and revoke privileges on these to other users.
- DBA access includes all of the connect and resource privileges and allows you to access any data in the database, grant and revoke access privileges at the database level, create public synonyms, create and alter partitions, and perform full database exports, as well as other functions.

Version 6 has no notion of access to groups or applications, as does Ingres. In a distributed environment, these two levels of access are particularly important, and Oracle might want to consider implementing them.

Futures

Along the way, we've mentioned some of the enhancements planned for Oracle's next release, Version 7. With Version 7, Oracle hopes to extend its OLTP and distributed capabilities with support for distributed transaction processing, data replication, and database-enforced integrity. (We hope these capabilities won't be sold as a separate option.) In addition, Version 7 will include PL/SQL stored procedures and role-based security management.

Despite its insufficient OLTP functionality, the company is determined to make a show in that area by porting Version 6 to powerful transaction processing hardware environments—Stratus Computer's XA2000 system and Pyramid's MISserver line. Both computer lines are specifically architected to handle parallel multiprocessing and fault-tolerant transaction processing, which can only enhance Oracle's performance. (Interestingly, Stratus is a marketer of Sybase.) Along with the port to Pyramid, Oracle announced a joint venture with Independence Technologies (ITI) that will integrate the RDBMS with ITI's Transaction Manager, a transaction monitor facility. The Transaction Manager is ITI's version of AT&T's Tuxedo product. It's basically a traffic cop, monitoring and routing client processes to available RDBMS servers in a distributed environment for systems that deal heavily in OLTP. Although the product will run initially on Pyramid's MISserver line, availability for other Unix platforms will be released later in the year.

Oracle made another important announcement with NCube Computer Corporation (Freemont, Califor-

Oracle products are available on most Unix implementations, including:

Altos
Amdahl
Apollo
Apple
Arete
AT&T
BBN
CCI
Convergent
Convex
Cubix
Data General
Digital's Ultrix
Edge
Elxsi
Encore
Gould
Harris
Hewlett-Packard
Honeywell
IBM RT
ICL
Intel
Masscomp
MIPS
Motorola
NCR
Nixdorf
Plexus
Pyramid
Prime
Ridge
Sequent
Siemens
Sun
System V
Unisys
Xenix

Illustration 4.

Oracle Products Under Unix

PRODUCT	RUNS ON	PRICE*
Core Products		
Oracle RDBMS (Includes SQL*Loader and SQL*DBA)	See Illustration 4	Price ranges from $17K to $240K
SQL*Plus	See Illustration 4	25%
SQL*Forms	See Illustration 4	30%
SQL*Menu	See Illustration 4	15%
Developer Tools		
CASE tools:	HP, Sun3	
CASE*Dictionary		50%
CASE*Design		200%
CASE*Generator		30%
SQL*ReportWriter	See Illustration 4	30%
Pro*SQL	Varies depending on platform;includes interfaces for Cobol, C, Fortran, Pascal, and Ada	15%
User Tools		
SQL*Calc	See Illustration 4	20%
SQL*Graphics	Sun workstations only (ports to HP and Sequent scheduled for midyear)	Not Available
Oracle*Mail	Available on request (ports to date include Sun, Sequent, HP, Pyramid, AT&T, NCR, and Digital); should be generally available later this year	50%
SQL*QMX	Sequent, HP, and AT&T	20-25%
Oracle Financials	Sequent, HP, Data General, Pyramid, and Sun	100% per module
Oracle Core Manufacturing	Sequent, HP, Data General, Pyramid, and Sun	Not Available
Transaction Processing Option (Includes PL/SQL)	All platforms that V.6 runs on (see Illustration 4)	Ranges from 0 to 40%
Distributed Databases:		
SQL*Net	See Illustration 4	20%
SQL*Connect	Support for DB2 and SQL/DS	15%

* (Based on percentage of RDBMS kernel)

Illustration 5.

nia) last June. NCube peddles supercomputers, and, during this year's first quarter, the Oracle RDBMS will be available for NCube 2, making Oracle available for the first time on a supercomputing platform. Additional porting plans include Intel 80486 and, perhaps, i860.

The Bottom Line

Oracle has some impressive strengths, most notably its application development and end-user tools. We hear good things from developers about SQL*Forms—especially when used in conjunction with PL/SQL. The implementation of SQL*ReportWriter, with its menu-driven interface, is a significant improvement over Oracle's previous report writer. SQL*Menu, Oracle's fourth-generation menu-development component, provides a smooth integration among Oracle components.

Oracle's Unix environment, however, is incomplete—there are so many Unix variations out there. Not all the tools—especially the new ones—are generally available under Unix.

Oracle's recent interest in applications and consulting might be distracting the company from developing advanced RDBMS functionality. Oracle is obviously not just a database company any more. Its consulting avenue seems to us to have an inherent conflict. What's in it for Oracle to recommend someone else's product? Clearly some objectivity gets lost.

But where we found Oracle seriously lacking was in the RDBMS core: no server-side integrity rules or stored procedures, no user-defined data types, no data replication, no distributed transaction management, no partitioned database tables... While Oracle plans to remedy these shortcomings with Version 7, we'll postpone any comments on those remedies until the product is actually out. However, given Oracle's abundance of cash, we hope the RDBMS won't suffer any technical deficiencies for long. Advanced functionality for distributed computing and transaction processing may not have been originated in Oracle's R&D labs, but that doesn't mean the company can't capitalize on it once it's been incorporated. Considering Oracle's presence and marketing, it may profit substantially from technologies introduced elsewhere.

NEWS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

ANALYSIS

•HEWLETT-PACKARD•

Absorbing Apollo's RISC

It has been more than six months since Hewlett-Packard swept troubled Apollo Computer into its fold. While in many ways the union made sense, in some areas, the two had overlapping technology—primarily in RISC (Reduced Instruction Set Computing) workstations. Now, after a lot of work and analysis, HP seems to have found a comfortable and sensible spot for Apollo. In an update to analysts last month, HP explained that Apollo now has Research and Development responsibility for its RISC workstation family.

RISC STRATEGY. In line with the consolidation of resources between the two companies, HP plans to consolidate the two RISC product lines. With its careful attention to detail, HP is providing a slow evolution from Apollo's PRISM RISC architecture and its own Precision Architecture (HP-PA): By 1993, HP expects to have a composite RISC architecture based on the best of Apollo's PRISM and the best of HP-PA. For example, it will pick up the 64-bit address space from the PRISM architecture as well as the parallel reduced instruction set that Apollo has supplied since PRISM's introduction in 1983. The base of the product will still be HP's architecture. HP's strategy is to evolve a virtual instruction set that will be the same on all RISC platforms and will carry the company into the '90s. However, HP is careful to note that it expects the instruction set to change over time. In such a competitive market, it would be a mistake for HP to lock itself into technology that could be obsolete in a matter of months.

MIDLIFE WORKSTATIONS. Before reaching this long term goal of a single RISC architecture, HP will have an interim generation out in 1990-91. This generation will provide two platforms: one based on the Apollo Series 10000 PRISM machine, and another based on the HP Series 800 line. Both products were already in the pipeline, and it makes sense for HP to offer these higher powered models as midlife products.

These products will be nothing to sneer at. For example, the new CPUs will have as much as twice the power of the earlier workstations (as much as 44 MIPS and 12 MFLOPS per CPU) as well as have two to four times the parallelizing and vectorizing compilers. Both workstations will be aimed at scientific/engineering applications, where paging and power are king.

THIRD-GENERATION RISC. The combined architecture has ambitious goals. The workstations are targeted to run at more than 100 MIPS per CPU and perform at 60 to 90 MHz, executing a single instruction per cycle.

REFOCUSING AWAY FROM MINIS. There is no doubt that HP's efforts in developing workstation technology for the '90s are on target. We expect that these plans will be competitive with what IBM will be announcing later in the month and with technology from Sun and Digital/MIPS. But what may be the most significant part of the strategy is its separation from HP's minicomputer roots. Company management has stated clearly that its future is in engineering and in commercial workstations and servers. Smart move. It doesn't take a genius IQ to notice that people aren't buying minicomputers these days. Servers and distributed computing are the directions of the future, and HP seems to have this market clearly targeted.

INTEL AND MOTOROLA, TOO. In addition to its RISC moves, HP is promising to continue its aggressive development on both the Motorola and

Intel platforms. For example, this month it announced two new Motorola-based workstations: Models 345 and 375. The 345 will be based on the 50 MHz version of the 68030, while the 375 will be the first machine based on the 40 MHz 68040. Pricing is aggressive—hitting below the magical $10,000 barrier (Model 345, with 12 MIPS, will start at $8,995). The high-end Model 375 will begin selling at $21,995. Both models will ship during the first quarter of 1990.

AND SOFTWARE, TOO. Targeting these markets is a good move for HP. But the acid test will be software. HP is well-positioned here, too—mostly thanks to Apollo's Network Computing System (NCS) and Network License Server. Apollo has long had many of the commercial pieces, like backup and software distribution, needed to implement distributed computing in a commercial environment. HP is finding ways to leverage this software as Apollo was never able to do on its own.

CONCLUSION. HP is on a roll. It is making good use of Apollo resources in the workstation and distributed computing arenas. It is keeping pace with the frenetic world of RISC technology. It is also doing a good job of developing new software for both office and the CASE environment. Therefore, the picture is rosy for the most part. However, the platter is quite large and varied. HP is committed to supporting a great number of operating systems (Apollo's proprietary Unix-like operating system, HP/UX, DOS, OS/2, the future OSF operating systems) and hardware platforms (Intel, Motorola, and two varieties of RISC). The challenge will be to keep all of these tasks going at an equally aggressive level. As HP moves more and more into commercial Unix, it will have to find new channels of distribution and new methods for supporting a less sophisticated user population (compared to engineering workstation users). Although HP seems to have the infrastructure in place for the future, it will have to do a careful balancing act.

—*J. Hurwitz*

NewWave Gets Its Unix

When NewWave was introduced in October 1988, HP disclosed its timetable for developing a truly distributed object environment based on it. By the end of 1991 (or perhaps 1992), the company said, NewWave would provide distributed object management in an environment that includes Unix.

Well, 1991 is only a year away, and HP has taken some steps towards the stated goal. With the announcement of NewWave Office on November 28, 1989, HP introduced interoperable shared objects in a networked environment—not yet distributed, mind you, but networked nonetheless. Estimated delivery date is end of first quarter '90.

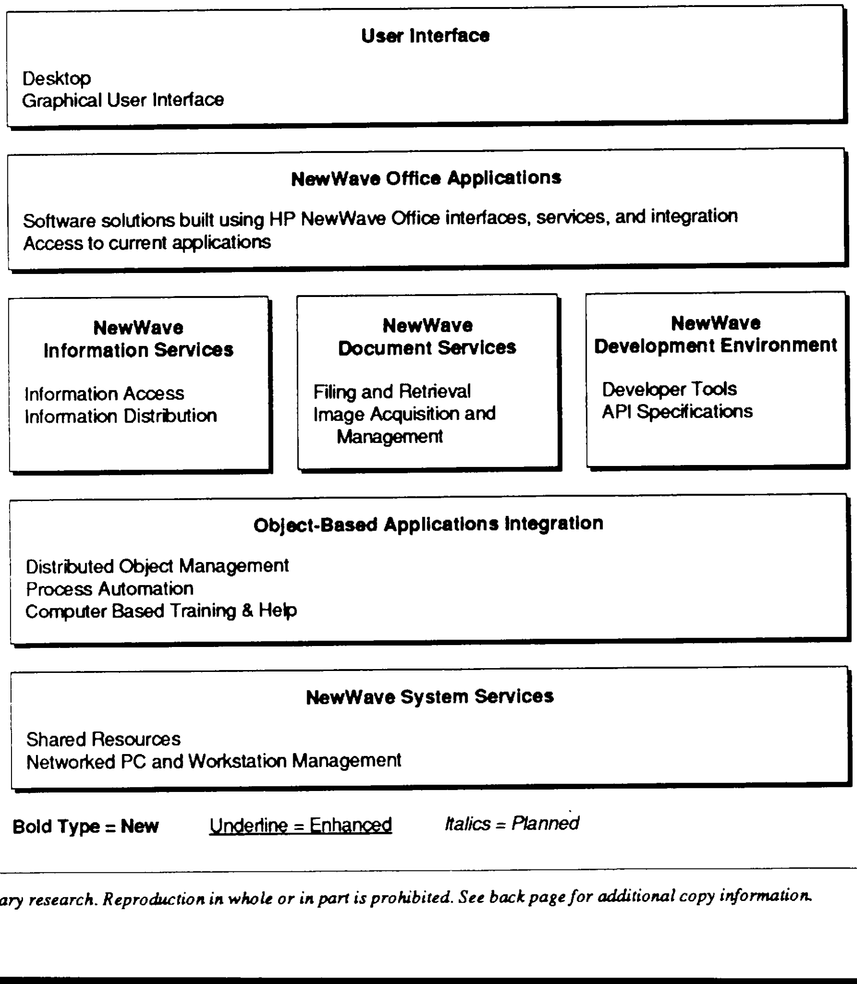
HP has now fulfilled its commitment to making Unix a player in the NewWave environment by including HP-UX as a server for NewWave Office. (See illustrations, below.)

COMPONENTS OF NEWWAVE OFFICE. NewWave Office includes the following tools and services:

- NewWave environment, which offers a consistent graphical user interface and integration of existing and new PC applications, as well as providing task automation and the HP Object Management Facility (OMF) which links data and applications together so users can move transparently across different types of software.

- Information Access, which allows PC users to retrieve data from MPE

NewWave Office: Functional Definition



User Interface

Desktop
Graphical User Interface

NewWave Office Applications

Software solutions built using HP NewWave Office interfaces, services, and integration
Access to current applications

NewWave Information Services

Information Access
Information Distribution

NewWave Document Services

Filing and Retrieval
Image Acquisition and Management

NewWave Development Environment

Developer Tools
API Specifications

Object-Based Applications Integration

Distributed Object Management
Process Automation
Computer Based Training & Help

NewWave System Services

Shared Resources
Networked PC and Workstation Management

Bold Type = New Underline = Enhanced *Italics = Planned*

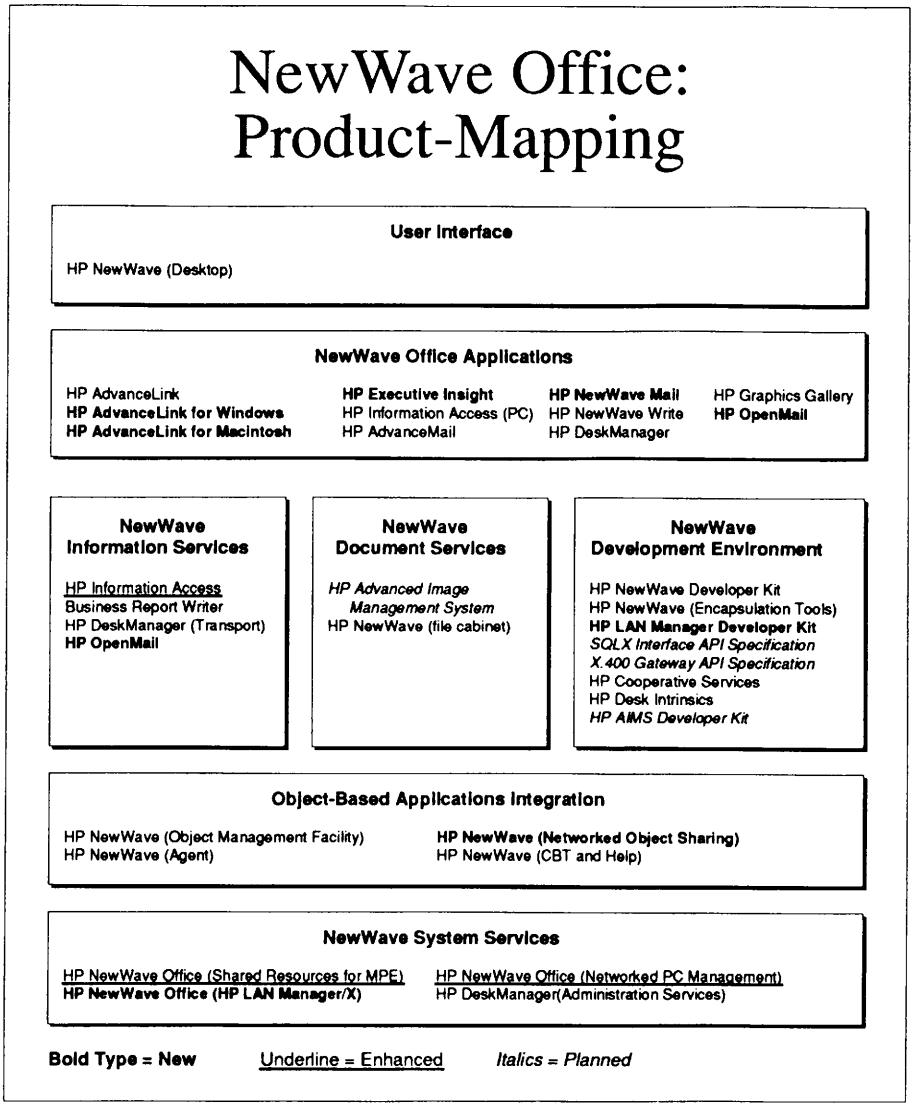
(HP's proprietary operating system), HP-UX (HP's Unix implementation), and OS/2 servers, and output the data to the PC application of choice. Information access supports Oracle and HP SQL databases on both MPE and HP-UX systems. RBase, dBase, and Lotus 1-2-3 are supported on OS/2.

- Information Distribution, which enables users to exchange information on both private and public mail systems. Based on HP OpenMail and HP DeskManager E-mail systems, information distribution offers connectivity with X.400, Unix system mail, PROFS, and DISOSS.

- Shared Resources, which allow users to share disks, printers, etc. across networks, including HP LAN Manager-based offerings, 3Com's 3+Open, and (to a limited degree) Novell NetWare.

- Networked PC Management, which provides central distribution and updating of PC software, simplified PC software installation, and automated PC backup capabilities.

- Development tools built on standard components, such as Microsoft Windows, Presentation Manager, LAN Manager, ANSI SQL, and X.400. Also available are the HP NewWave Developer Kit with encapsulation tools to integrate existing PC applications into the NewWave environment, the LAN Manager Developer Kit, and capabilities to integrate existing HP3000-based applications.



NewWave Office:
Product-Mapping

User Interface

HP NewWave (Desktop)

NewWave Office Applications

HP AdvanceLink
HP AdvanceLink for Windows
HP AdvanceLink for Macintosh
HP Executive Insight
HP Information Access (PC)
HP AdvanceMail
HP NewWave Mail
HP NewWave Write
HP DeskManager
HP Graphics Gallery
HP OpenMail

NewWave Information Services

HP Information Access
Business Report Writer
HP DeskManager (Transport)
HP OpenMail

NewWave Document Services

HP Advanced Image Management System
HP NewWave (file cabinet)

NewWave Development Environment

HP NewWave Developer Kit
HP NewWave (Encapsulation Tools)
HP LAN Manager Developer Kit
SQLX Interface API Specification
X.400 Gateway API Specification
HP Cooperative Services
HP Desk Intrinsics
HP AIMS Developer Kit

Object-Based Applications Integration

HP NewWave (Object Management Facility)
HP NewWave (Agent)
HP NewWave (Networked Object Sharing)
HP NewWave (CBT and Help)

NewWave System Services

HP NewWave Office (Shared Resources for MPE)
HP NewWave Office (HP LAN Manager/X)
HP NewWave Office (Networked PC Management)
HP DeskManager(Administration Services)

Bold Type = New Underline = Enhanced *Italics = Planned*

NEW PLATFORMS. The NewWave Office announcement builds on the NewWave environment. NewWave had previously run only on DOS and MS Windows as a standalone environment. These NewWave clients could take advantage of MPE services (though NewWave code did not run on the HP3000 itself). The new announcement expands server platforms to both Unix (HP-UX only at this point) and OS/2. Not announced, but definitely planned for the future, are a Unix Motif client and an OS/2 Presentation Manager client. No definite time frame was given. The Macintosh has also been made part of the NewWave family. But, unlike the other existing and planned client models, the Mac operates in terminal emulation only.

Not Created Equal. Though both the Unix and OS/2 servers were announced, only the Unix server will offer equal functionality with MPE. The OS/2 server does not yet offer any network management services.

LAN Client Support. NewWave Office also supports local area networks allowing shared resources—something that standalone NewWave did not do. At this time, all servers support both HP NewWave clients and 3Com 3+Open clients through LAN Manager for Unix (LM/X). However, the 3Com folks get none of the NewWave functionality unless the clients themselves are running NewWave. Novell NetWare clients are supported only on MPE servers via an HP NS LAN gateway.

HP has also added tools to help manage networked PCs and large distributed networks. Services include:

- Centralized client software distribution, including both HP and third-party software

- Client software update service

- Easier network software installation

- PC hard disk backup (MPE only)

- Server backup of shared disks

NETWORKED OBJECT-SHARING. One of the more interesting elements of the new announcement is networked object-sharing, anticipated for delivery in April. NewWave Office workgroups can share objects among group members as well as with other users on the network. Even more significantly, users can share objects across platforms. For example, a user's agent can include objects from both an MPE server and a Unix server. The facility includes a central object storage and management facility as well as concurrency control to track objects as they are checked in and out of central storage; basically, HP is using a logical drive on the network to share stored objects.

The actual sharing of objects is very transparent. Opening an object from a shared cabinet implicitly checks the object out, making it "read only" to others. Returning the object to the cabinet implicitly checks it back in. Locating the object, however, is not transparent enough. You have to know where—which cabinet, not necessarily which server—the object resides.

NewWave gives considerable power to the user, allowing him or her to move items in and out of shared cabinets (and private cabinets, for that matter) by dragging them around. There are a few problems with this easy manipulation, though. Any user who can access the shared objects can delete one. If the object is referenced in other objects (compound documents) or agents (procedures), they are clobbered, and they never know what hit them. Also, HP allows users to name objects anything, and the names need not be unique. Thus, a shared cabinet can contain any number of objects called "monthly report." This can be very confusing.

Transparent sharing of objects is key to the effective use of an object-oriented environment. As we mentioned, the capability is not truly distributed—shared objects must reside in the central storage facility. But HP plans to move to a fully distributed model in the future.

THIRD-PARTY APPLICATIONS. While we have long been fans of the NewWave environment, including the concepts it stands for, there was a time when we feared for its longevity. ISVs had a lot of platform options from which to choose, and, though we had confidence in HP's vision of the office, we had to wait and see if third-party developers did, too.

Well, as of announcement day on November 28, 1989, over 80 NewWave Partner applications were being developed. However, only 15 of them are or will be available within a few months, and none is a real big-timer (Samna's Ami Professional and Channel Computing's Forest and Trees are representative of the crop). But remember, off-the-shelf DOS applications can be encapsulated in the NewWave environment. However, we do hope to see some of the big guns write to the NewWave specifications.

CONCLUSION. With NewWave Office, as it currently is defined, HP has taken a big step in the right direction toward distributed computing and object management across multiple platforms. The company appears to be keeping to its timetable nicely. And this first step, is in many ways, a doozy. Shared interoperable objects in a networked environment should make life a lot easier for end users and for developers.

But, with the new networked environment, HP faces new challenges. Administration of a workgroup system is very different from keeping track of a bunch of standalone machines. Once you start sharing items, you have to be sure they aren't misplaced. Remember, in kindergarten, we learned the rules of sharing (for example, rule number one: If you take a toy, be sure to put it back where you found it so others can enjoy it, too). These sharing rules are critical, and administering them is no easy task. HP needs to provide management tools and security with sufficient granularity to make sure we all share nicely.

—R. Marshak

•NCR•

NetWare Comes to Unix

The success of Unix hangs on a number of variables—among them, coexisting and enhancing the computer configurations already in place in most corporations. In recognizing this, NCR has taken advantage of Novell's prominent position in the world of PC LANs and implemented portable NetWare for its Tower series of Unix processors.

FEATURES. NCR's adaptation of NetWare enables Tower processors to become servers in a client/server architecture. The product offers most standard NetWare features, including:

- Transparent file storage and sharing. DOS, OS/2, and Macintosh users can store files on the Tower NetWare server in a separate directory area. Although they're stored as Unix files, they're accessible through native naming conventions.
- Host printer sharing.
- Security (combining NetWare security with Unix security where appropriate).
- Resource management and diagnostics.

New features include terminal emulation for DOS users and a system console that performs administrative functions such as displaying and modifying server information and backing up the file server.

COMPATIBILITY AND INTEROPERABILITY. A Tower NetWare server is compatible with existing NetWare clients. Through bridges, the Tower supports Token-Ring, ARCNet, and AppleTalk, as well as TCP/IP and Ethernet-based NetWare environments.

ISV SUPPORT. A number of ISVs plan to take advantage of the Tower's server role in a NetWare environment and are porting their software accordingly. Database vendors see it as a vehicle for enhancing their distributed-database schemes. Oracle has announced limited release this month, with general release later this quarter. Informix and Ingres announced plans for support as well. Other vendors, such as Uniplex, Access Technologies, and WordPerfect have also announced plans for support. And Cheyenne Software, a network management software vendor, plans to have Tower NetWare products available this month.

NETWARE AND DCE. While it might not make much of a difference to NCR, a staunch member of Unix International, most companies who are interested in distributed computing are watching the results of OSF's request for technology for Distributed Computing Environment (DCE) very carefully. NetWare, as well as PC-NFS, is a possible solution to the integration of PCs and PC LANs into DCE. However, it appears that LM/X (LAN Manager for Unix, or "portable" LAN Manager), which has the powerful support of IBM, Digital, Hewlett-Packard, Microsoft, and others (including AT&T, which, though not formally supporting the submission, is an LM/X OEM), is likely to be chosen as the integration standard. Where this would leave NetWare, with its tremendous installed base, is unclear.

CONCLUSION. For NCR, it's a practical move; there's a lot to be gained in providing the multiuser, multiprocessing functionality of Unix as a server for PC clients. Although the Tower series isn't exactly on the cutting edge of hardware technology, Novell is tickled about the whole thing nonetheless. Tower NetWare strengthens its reputation as a vendor of open network computing solutions. Furthermore, it may just open up a whole new avenue for Novell—Unix. *—L. Brown*

•SQL ACCESS GROUP•

Will Database Interoperability Become a Reality?

At last, someone is attacking the nonstandard SQL problem. SQL Access Group is dedicated to promoting interoperability among SQL-based relational database management systems (RDBMSs). Under development is a set of detailed specifications for a standard programming interface and a network communication protocol to allow applications and DBMS servers from different vendors to communicate. The SQL Access Group is focusing its efforts to communication among SQL-based applications and DBMSs, basing its specifications on the SQL standard and on the evolving ISO Remote Data Access (RDA) standard. The group plans to publish a technical guide to interoperability in early 1990; a working demonstration of its implementation will follow in the second half of the year.

THE GOAL. The goal is transparent access between a client application and a database server. Ideally, any application on any system will be able to access any data over any network. Needless to say, achieving this is a complex problem with several dimensions. The problem applies not only to accessing relational databases (the focus of the new group), but to accessing the vast amounts of data stored in nonrelational and file systems as well. This is far more than just a networking problem.

Client/Server Protocols. Client applications and database servers use different client/server protocols to talk to each other. An Oracle application, for example, cannot connect to a Digital Rdb server because it sends SQL statements in a different sequence than the one Rdb expects to receive. The messages sent by the client are not intelligible to the server and vice versa. The client/server protocol must be consistently implemented to achieve interoperability. It must also be implemented the same way regardless of whether the transport is SNA, DECnet, TCP/IP, etc. and accommodate multiple networks.

The ISO/RDA committee is addressing this client/server protocol issue. The communications portion of the SQL Access Group's solution will be RDA-based. One objective is to speed up the work currently being done by the ISO/RDA committee.

SQL Implementations. With the existence of an SQL standard, one would expect to see SQL implemented consistently across RDBMSs. But this is not the case. As written, the SQL standard is not detailed enough to ensure a standard implementation. Not only are there semantic differences between vendor implementations of the same SQL statement, but each vendor has also chosen its own set of extensions to its set of "standard" SQL statements.

A related issue is inconsistent datatypes among RDBMSs. Different databases support different datatypes, and can even define the same datatype (e.g., floating point) in different ways. Other problems include different international character sets, and the need to translate between ASCII and EBCDIC.

Error Codes. Another aspect of incompatibility is each vendor's set of proprietary error codes. A foreign client application cannot understand the error messages received from the server.

Data Dictionary. There is a wide divergence among RDBMSs in how the data dictionary is structured and accessed. This is a critical issue since the data dictionary describes the database and must be accessible to both the developer and the application/user.

THE CURRENT SOLUTIONS. There are three approaches to DBMS interoperability today: the SQL Access Group, Sybase's Open Client/Server, and Open

Desktop. None of these provides a complete solution.

SQL ACCESS GROUP. The SQL Access Group's focus is on developing an SQL server specification detailed enough to produce database interoperability and application portability when implemented by multiple DBMS vendors. The key is implementation consistency. The same front-end tools and applications will operate against any compliant DBMS server, allowing a developer to write vendor-independent database applications.

Technical Specification. The specification will be based on existing standards, and will include both an SQL Application Programming Interface (API) and a formats and protocols (FAP) specification. When published, the specification will be available to anyone for the cost of the documentation. It will also be submitted to the standards bodies for adoption. The group did not indicate that it would provide certification services, although it will develop a working, heterogeneous demonstration of the specification.

The SQL API developed by the SQL Access Group will be of the "common denominator" variety (you can choose the appropriate modifier—"greatest" or "least"—depending on your perspective). It will also include escape mechanisms to allow use of a DBMS's proprietary language and extensions to the standard SQL.

The FAP specification will cover communication between a single-vendor client and a single-vendor server. There will be no distributed database capabilities initially. The RDA standard simply defines communication between a single client and a single server. It does not include communications between servers, nor does it handle distributed database issues, such as a client accessing multiple database servers in a single transaction or SQL statement.

Providing a consistent SQL interface across multivendor DBMSs is the first step in building a strong platform for heterogeneous distributed databases. The group stated that the second release of the specification will include a two-phase commit protocol, although this by itself is not sufficient for distributed databases. There also must be an automatic recovery manager, and each vendor would have to build its own until the specification includes it.

Members. The SQL Access Group, formed in August 1989, is a non-profit organization open to all vendors. Current members include Ashton-Tate, Digital, Fujitsu America Incorporated, Hewlett-Packard, Informix, Ingres, Metaphor, NCR, Oracle, Sun Microsystems, Tandem Computers, Teradata Corporation, and Wang. Notably missing is IBM. The group expects third-party activity to define SQL access to DB2 if IBM does not join. Sybase originally participated but dropped out since it felt that its Open Client/Server approach was more appropriate.

SYBASE OPEN SERVER. Sybase is licensing its client and server APIs to provide hardware-, software- and network-independent client/server communications. Developers and user organizations will now be able to integrate non-Sybase front ends and servers into the Sybase environment using the Sybase Open Client and Open Server APIs. The most significant aspect of the new Open Server: It can interface to any server, and is not limited to SQL-based servers. We covered the Sybase Open Server in some detail in last month's News Analysis.

Sybase has solved the communications problem, but not the application incompatibilities. In the Sybase approach, any Open Client can communicate with any Open Server, but the solution is not language- and error code-independent. The language encapsulated in a Sybase remote stored procedure is proprietary to the server. The user must also provide some of the translations necessary, for example, between proprietary and Sybase generic error code constructs.

One advantage provided by the Sybase Open Client/Server APIs is the ability to communicate server-to-server as well. Neither Open Desktop nor the SQL Access Group has stated plans to tackle this problem.

OPEN DESKTOP. Open Desktop is a package bundled for the 386 Unix platform. It includes software from Digital and HP (OSF/Motif), Ingres (Ingres RDBMS), Locus (distributed file system), and SCO (the Unix operating system). As the DBMS provider for the Open Desktop suite, Ingres has implemented features to facilitate access to heterogeneous DBMSs.

One is its global communications architecture (GCA), which is an implementation of RDA for the client/server protocol and a gateway server that runs on the platform with the DBMS server. With Ingres 6.0, these servers can be Ingres, IBM's DB2 or SQL/DS, or Digital's Rdb. (When RDA reaches international standard status—which is expected in 1991—Ingres will move to the full standard.)

Ingres has also developed what it calls Open SQL, a set of SQL statements common across the set of DBMSs accessible to an Ingres application. Open SQL is a subset of the full Ingres SQL. Like the SQL Access Group's API, Open SQL is a common denominator among the DBMSs. Ingres has also chosen to implement some features even though not all accessible DBMSs support them. For example, Open SQL supports repeat queries, emulating this for DBMSs that don't.

The Ingres approach achieves portability of applications. As long as the application is written with Open SQL, it doesn't matter whether the back-end DBMS is Ingres, DB2, SQL/DS, or Rdb. The gateway will be able to make the appropriate translations.

Current limitations to the Open Desktop solution include the fact that the application must be an Ingres application, and that the Ingres-developed gateway server is necessary to translate SQL statements, datatypes, error codes, and the data dictionary. It also cannot access non-SQL-based DBMSs.

—J. Davis

Upcoming Conferences from Patricia Seybold's Office Computing Group

The Second Annual
Executive UniForum Symposium

The Applications Development Environment of the 1990s:
Can Unix Set the Innovation Agenda?

May 22, 23 & 24, 1990, Four Seasons Biltmore Resort Hotel, Santa Barbara, California

Sponsored by Patricia Seybold's Office Computing Group, UniForum, and X/Open

The theme of this year's Executive UniForum Symposium is the Unix software and applications development environment. If Unix is to become a commercially viable operating system, it must be able to support, attract, and nurture the next generation of applications and development tools. Is Unix up to the task?

This conference will present the views of industry leaders, including hardware, software, and networking vendors, as well as the views of commercial users. It will also provide a forum to spotlight the next generation of Unix applications.

> *"It synthesized the relationships between all the major players in the Unix industry."*
>
> *—1989 Executive UniForum Symposium Attendee*

Registration Fee:

- $895 for all registrations *paid by* February 12, 1990.
- $1095 for all registrations *paid after* February 12, 1990.

The Fourth Annual
Patricia Seybold's Technology Forum

Distributed Network Computing and
Object-Oriented Environments: Pillars for the 1990s

April 2, 3, 4, 1990 , Cambridge Marriott Hotel, Cambridge, Massachusetts

Two emerging technologies will become the supporting pillars of the application architectures of the next decade: *distributed network computing* and *object orientation.*

This year's Technology Forum explores each of these critical technology sets separately in twin tracks, exploring current options and solutions. Throughout the conference, attendees will have the option of moving back and forth between tracks, or staying on a single track from start to finish. In joint sessions at the opening and closing of the conference, we will examine the intersection of these two technologies, as well as the future directions for each.

> *"As with all Seybold Forums, I was 'quantum leaped' into the future of new technology."*
>
> *—1989 Technology Forum Attendee*

Registration Fee:

- $895 for Office Computing Group Subscribers
- $1095 for non-subscribers

For more information about either of these conferences, call Deborah Hay at 1-800-826-2424 (in Massachusetts, call 617-742-5200).

Patricia Seybold's Computer Industry Reports

ORDER FORM

Please start my subscription to:		U.S.A.	Canada	Foreign
□ *Patricia Seybold's Office Computing Report*	12 issues per year	$385	$397	$409
□ *Patricia Seybold's UNIX in the Office*	12 issues per year	$495	$507	$519
□ *Patricia Seybold's Network Monitor*	12 issues per year	$495	$507	$519
□ *P.S. postscript on information technology*	12 issues & tapes per year	$295	$307	$319
□ *P.S. postscript on information technology*	12 issues per year	$ 95	$107	$119

Please send me a sample of: □ *Office Computing Report* □ *Network Monitor*
□ *UNIX in the Office* □ *P.S. postscript on information technology*

Please send me information on: □ Consulting □ Special Reports □ Conferences

□ **My check for $________ is enclosed.** □ **Please bill me.** □ **Please charge my subscription to: Mastercard/Visa/American Express** (circle one)

Name: ________________ Title: ________________

Company Name: ________________ Dept.: ________________

Address: ________________

City, State, Zip code: ________________

Country: ________________ Bus. Tel. No.: ________________

Card #: ________________

Exp. Date: ________________

Signature: ________________

Checks from Canada and elsewhere outside the United States should be made payable in U.S. dollars. You may transfer funds directly to our bank: Shawmut Bank of Boston, State Street Branch, Boston, MA 02109, into the account of Patricia Seybold's Office Computing Group, account number 20-093-118-6. Please be sure to identify the name of the subscriber and nature of the order if funds are transferred bank-to-bank.

IU-0190

Send to: Patricia Seybold's Office Computing Group: 148 State Street, Boston MA 02109; FAX: 1-617-742-1028; MCI Mail: PSOCG
To order by phone: call (617) 742-5200

Topics covered in Patricia Seybold's Computer Industry Reports in 1989:

Back issues are available, call (617) 742-5200 for more information.

STATEMENT OF OWNERSHIP, MANAGEMENT AND CIRCULATION. **1A. Title of Publication:** *Patricia Seybold's UNIX in the Office*. **1B. Publication no.:** 08875054. **2. Date of filing:** November 1, 1989. **3. Frequency of issue:** Monthly. **3A. No. of issues published annually:** 12. **3B. Annual subscription price:** $495.00. **4. Complete mailing address of known office of publication:** 148 State St., Suite 612, Boston, MA 02109. **5. Complete mailing address of the head-quarters of general business offices of the publisher:** 148 State St., Suite 612, Boston, MA 02109. **6. Publisher:** Patricia B. Seybold, 148 State St., Suite 612, Boston, MA 02109. **Editor:** Judith S. Hurwitz, 148 State St., Suite 612, Boston MA 02109. **Managing Editor:** John Rymer, 148 State St., Suite 612, Boston MA 02109. **7. Owner:** Patricia Seybold's Office Computing Group, Inc., 148 State St., Suite 612, Boston MA 02109, Patricia B. Seybold, 148 State St., Suite 612, Boston MA, 02109, Ronni T. Marshak, 148 State St., Suite 612, Boston MA 02109, Michael D. Millikin, 148 State St., Suite 612, Boston, MA 02109, Judith S. Hurwitz, 148 State St., Suite 612, Boston, MA 02109. **8. Known bondholder, mortgagees, and other security holders owning or holding 1 percent or more of total bonds, mortgages, or securities:** None. **9. For completion by nonprofit organizations authorized to mail at special rates:** Not applicable. **10. Extent and Nature of Circulation:**

	Average 12 Mos.	Single Issue
A. Total no. copies printed	600	1000
B. Paid Circulation		
1. Sales through dealers and carriers, street vendors and counter sales	None	None
2. Mail Subscriptions	450	904
C. Total Paid and or Requested Circulation	450	904
D. Free distribution by mail, carrier or other means samples, complimentary, and other free copies	25	41
E. Total Distribution	475	945
F. Copies not Distributed		
1. Office use, left over, unaccounted, spoiled after printing	125	55
2. Returns from news agents	None	None
G. Total	600	1000

I certify that the statements made by me above are correct and complete.
(signed) Patricia B. Seybold, Publisher

UNIX in the Office

1989—Volume 4

#	**Date**	**Title**
3	Mar.	FrameMaker 2.0—The Art and Science of Document Processing
4	Apr.	CMU's Andrew Project—A Prototype for Distributed, Object-Oriented Computing
5	May	Sybase—The Next Generation
6	June	Thoroughly Modem Unix—The Shape of New Operating Systems
7	July	Surveying the Users—Taking the Pulse of Commercial Unix
8	Aug.	Graphical User Interfaces—A Developer's Quandary
9	Sept.	The Double Standard—DOS under Unix Moves to the Next Generation
10	Oct.	Reality Check—Commercial Users Speak Out on Barriers to Unix Acceptance
11	Nov.	To Lead or Follow?—Which Route Have Uniplex, Applix, and Quadratron Chosen?
12	Dec.	Targon Office—Nixdorf's Object-oriented Platform for Application Integration