Patricia Seybold's
Office Computing
Group

UNIX IN THE OFFICE

Guide to Open Systems

Vol. 6, No. 8 • ISSN: 0887-3054 • August 1991

Editor-in-Chief
Judith S. Hurwitz

Interbase Software

Extending The Relational Model To Handle Complex Data

By Judith R. Davis

IN BRIEF: Interbase Software is still relatively unknown when compared to larger competitors like Oracle and Sybase. But the InterBase relational database management system (RDBMS) is beginning to garner attention with its extensions to the relational model for online complex processing (OLCP)—a combination of online transaction processing (OLTP) and the ability to handle complex data. InterBase can store and manipulate BLOBS, has server-enforced integrity and event alerters, and enhances performance by eliminating conflicts between readers and writers. It also supports heterogeneous distributed processing and distributed database. Combining this architecture with the extensive resources of new parent Ashton-Tate (and now Borland International), and partnerships with third-party tools vendors, Interbase Software is worth a look from anyone evaluating relational DBMSs.

Report begins on page 3.

EDITORIAL: BY JUDITH S. HURWITZ

A Dangerous Time for Open Systems

Computer Vendors Must Not Forget about User Requirements in the Midst of the Confusion

CHAOS REIGNS in the computer industry. It is no longer simple to explain its hierarchy or power structure. Anyone who ventures to predict what it will look like next week is doomed to be proven wrong. So what is going on? Here is my perspective. The industry is in the midst of a massive power struggle. This was caused by the failure of the major computer industry players to satisfy user requirements and deliver on their software promises of the past decade. Each major supplier has had at least one major disaster: IBM had serious problems with SAA and OfficeVision; Digital Equipment has taken five years longer than expected to deliver its NAS infrastructure; Hewlett-Packard is still working to produce and deliver the full promise of NewWave and NewWave Office; Microsoft and IBM promised that OS/2 would be the successor to DOS; AT&T promised that there would be one unified version of Unix that would take the world by storm; The Open Software Foundation promised a new standard operating system to replace AT&T's Unix.

I could go on and on with examples of promises not kept. Indeed, the 1980s was a decade of promises. Open Systems was around the corner. We would choose one universal operating system and users would get the type of exciting software they needed within a year or so.

I suspect that the '90s will be remembered as a time when the computer industry started over. It will be remembered as the decade when the industry rebuilt itself from the ruins of the past. We will see new players emerge to lead us into the next century. Perhaps we will look back to the '80s as the time when IBM, Digital, Microsoft, Sun, and Hewlett-Packard (to name but a few) ruled, just as we remember the old BUNCH (Burroughs, Univac, NCR, CDC, and Honeywell) from the 1960s.

I said that this is a dangerous time for the Open Systems movement. Why? Because as the industry realignments emerge, vendors may forget about the work they have to do today to satisfy user requirements for the next five to seven years. Without continued commitment to making their technologies and systems conform to industry standards and interoperate, users will continue to stand still. They will not implement new technologies for fear that they will be caught with obsolete technology. They will not buy because they do not want to face the internal political consequences of making the wrong technology choices.

While users are worrying about finding a good accounting package and integrating their three incompatible databases, vendors are busily talking about new alliances and the new operating systems they will develop. No wonder users are a bit upset.

But these alliances are important. There is too much work to be done and too many problems to be solved for one vendor to do it all. There is too much power at stake for one vendor to try to take it all. As we move into the era when software will be a greater percentage of computer industry revenue, it is suddenly dawning on vendors that good innovative software isn't created in a vacuum. Those shaping the industry must ensure a smooth migration to open, distributed computing. This must happen in a way that will allow users to leverage what they are implementing today and allow them to enjoy the fruits of vendor partnerships tomorrow. Without this bridge from the past to the future, the computer industry will remain in a state of crisis for longer than anyone could imagine.

UNIX
IN THE
OFFICE

Editor-in-Chief
Judith S. Hurwitz

MCI:
JHurwitz
Internet:
jhurwitz@mcimail.com

Publisher
PATRICIA B. SEYBOLD

Analysts and Editors
JUDITH R. DAVIS
DAVID S. MARSHAK
RONNI T. MARSHAK
JOHN R. RYMER

News Editor
DAVID S. MARSHAK

Art Director
LAURINDA P. O'CONNOR

Sales Director
RICHARD ALLSBROOK JR.

Circulation Manager
DEBORAH A. HAY

Customer Service Manager
DONALD K. BAILLARGEON

Patricia Seybold's
Office Computing Group
148 State Street, 7th Floor,
Boston, Massachusetts 02109

Telephone: (617) 742-5200
Fax: (617) 742-1028
MCI: PSOCG
Internet: psocg@mcimail.com
TELEX: 6503122583

Unix in the Office (ISSN 0890-4685) is published monthly for $495 (US), $507 (Canada), and $519 (Foreign) per year by Patricia Seybold's Office Computing Group, 148 State Street, 7th Floor, Boston, MA 02109. Second-class postage permit at Boston, MA and additional mailing offices.

POSTMASTER: Send address changes to *Unix in the Office*, 148 State Street, 7th Floor, Boston, MA 02109.

FEATURED REPORT: BY JUDITH R. DAVIS

Interbase Software

Extending the Relational Model to Handle Complex Data

A Rising Star in the RDBMS World

Borland's Ownership Could Increase Interbase's Importance

At the moment, Interbase Software is a pretty small fish in the Unix relational database management system (RDBMS) pond. With annual revenues of perhaps $15 to $20 million (our guess), the company's size pales in comparison to the major independent Unix RDBMS vendors—Oracle, Ingres, Informix, and Sybase. It is also smaller than either Progress or Unify. One reason could be that its product, InterBase, hit the market at least two years after all of the others except Sybase. On the other hand, Sybase debuted a year after InterBase and is already up to $100 million in revenues. Sounds like a marketing problem, doesn't it? It does to us, particularly since InterBase has some of the same features that made Sybase so popular, such as triggers and high availability. With the acquisition of Interbase's parent company, Ashton-Tate, by Borland, Interbase's role as an important player may accelerate.

A Company Run by Engineers

Interbase is a classic example of a company run by engineers rather than marketeers—sort of like Ingres and Oracle, who were once the same size and fierce competitors. Oracle, very much marketing driven, grew much faster than Ingres, with its engineering-oriented management. Interbase has also focused primarily on the technical market—engineering and scientific applications—and has only recently started pursuing more commercially-oriented markets and applications. Another reason for the company's limited visibility was its initial concentration in the Apollo Domain environment. Although Apollo was an early leader in the workstation market, it couldn't stand up to the onslaught from Sun.

Strength: Engine Design; Weakness: Lack of Tools

But InterBase has some strong points in its favor, particularly the design of its database engine. And the time is right for the company to advertise those points. Interbase is now a wholly-owned subsidiary of PC software giant Ashton-Tate, with access to significant marketing, distribution, and development resources. And Interbase is busy fixing one of its major weaknesses—the lack of good application development tools. Partnerships with two third-party tools vendors have been announced recently—Convergent Solutions, Incorporated (CSI) and Jyacc.

In this report, we highlight the aspects of the InterBase architecture that differentiate it from the competition. This should provide food for thought in evaluating Unix RDBMS products. And we provide an overview of InterBase's two new tools partners, ADS/InterBase from CSI and JAM from Jyacc. Both are designed for database-independent development, but each supports a different application model. ADS/InterBase is language driven, and JAM is forms driven.

The Interbase Design Center

An Extended RDBMS

The best way to characterize InterBase is to describe it as an extended RDBMS. The company has implemented features that are not part of the relational model (such as arrays) but that satisfy important user application requirements for handling complex data. Other key design elements are high performance (transaction throughput, or OLTP capability) and support for distributed environments. (See Illustration 1.)

The Interbase Design Center

Effective Handling of Complex Data

A typical InterBase application might have a single massive updater (e.g., semiconductor testing equipment recording test results) with many users concurrently analyzing the data. The analysis transactions are often long-running and involve complex data that may be stored in BLOBs or arrays. The issue is effectively handling complexity—complex data and complex analysis—within an environment where lots of update transactions occur. The bottom line is providing both data access and performance. While Interbase got its start in the technical arena, these characteristics can also describe financial trading and other real-time commercial applications. This is an area where Interbase is starting to get some visibility.

Not an Object-Oriented DBMS

InterBase is clearly a relational DBMS, and this model has become well-accepted in the commercial world. While Interbase is extending the relational model in ways that overlap to some extent with functionality offered by object-oriented (OO) DBMSs, the company does not want its product to be confused with OODBMSs. If customers have the wrong perception, InterBase runs the risk of being characterized as an "inadequate object-oriented DBMS," rather than as an extended relational DBMS.

Interbase Software at a Glance

Corporate headquarters	209 Burlington Road Bedford, MA 01730 (617) 275-3222
Founded	1984
Product first introduced	1986
Latest release	InterBase Version 3.2
Financial:	
Ownership	Wholly owned subsidiary of Ashton-Tate
Fiscal year	January 1 - December 31
Total Revenues	Not available
Net income	Not available
Geographic breakdown of revenues:	
Domestic	95 percent
International	5 percent
Breakdown of revenues by channel:	
Direct sales	65 percent
VARs, OEMs	35 percent
Breakdown of revenues by platform:	
Unix	86 percent
Proprietary, DOS, and OS/2	14 percent
Distribution channels	
U.S. sales offices	7 sales offices
International sales locations	Sold through Ashton-Tate: international subsidiaries in U.K., Eastern Europe,Japan, and Canada
VARs	Yes; number not available
OEMs	Cognos
Installed base:	
Number of customers	Not available
Number of sites	Not available
Number of licenses	8,500
Number of users	Not available
Number of employees	Not available

Illustration 1.

Company Background

Who Is Interbase Software?

Interbase Software emerged out of the VAX/VMS world. The company was founded in 1984 by James A. Starkey, an engineer who left Digital Equipment after leading several significant design efforts there—Datatrieve and the Digital Standard Relational Interface (DSRI). The InterBase RDBMS clearly inherited much of its architecture from Starkey's Digital experience—support for DSRI (a superset of which is the InterBase server interface), a lock manager designed after VMS's, and transparent data access to Rdb and RMS data.

Interbase Is Now Owned by Ashton-Tate

In March 1991, Interbase became a wholly-owned subsidiary of Ashton-Tate, the third step in a long and slow marriage process. Ashton-Tate had purchased a 20 percent interest in 1987 and increased it to 51 percent in 1989. At that time, there was much talk of how InterBase would become the foundation for all of Ashton-Tate's future (post-dBase IV) database server products. However, at about this time, Ashton-Tate began to get bogged down in its own problems with developing and introducing dBASE IV 1.1. The company has only recently turned its attention to Interbase. Our guess is that Interbase was hoping to take advantage of Ashton-Tate's resources much sooner than it has been able to, and has been disappointed in the lack of support.

Borland Enters the Picture, Acquiring Ashton-Tate

The newest twist, which occurred just as we went to press, is the acquisition of Ashton-Tate by Borland International. Our initial reactions are that this transaction could serve to make Interbase even more strategic than it was with Ashton-Tate alone and that there could be some interesting long-term ramifications. Borland is known for its aggressive marketing, its success in several segments of the PC environment, and its interest in object-oriented software—a nice match for Interbase. The company also has some important relationships to contribute. Borland's ObjectVision development tool is part of IBM's overall approach to objects, and Borland has a close relationship with Novell.

All Will Benefit from the Acquisition

All participants get significant benefits from the acquisition. Borland/Ashton-Tate has bought impressive database server technology that will enable it to play in a multiuser, heterogeneous, distributed computing environment. (We wouldn't be surprised to find out that the opportunity to get Interbase was a factor in Borland's decision to acquire AT.) InterBase obviously has the necessary credibility here. Interbase gets the benefit of an extensive (and desperately-needed—see "Marketing Strategy" below) international distribution network; expertise, high visibility, and an immense installed base in the PC market; and financial backing to help Interbase compete more effectively with its bigger rivals. AT had annual revenues of $230 million in fiscal 1990, and Borland logged $227 million.

Meanwhile, Starkey has moved on to start his own business designing application development tools, returning periodically in a consulting role. The new president of Interbase, Paul Bergeron, comes from Stratus. The major thrust now is to improve Interbase's visibility, and Interbase is taking advantage of its recent acquisition by AT and joint announcements with CSI and Jyacc.

DBMS Architecture

Moving beyond Apollo Domain

InterBase is primarily a Unix RDBMS. InterBase was first introduced on Digital's VAX/VMS and VAX/Ultrix and on Sun and Apollo workstations in 1986. Since then, the company has added several other Unix platforms (see Illustration 2). The company is still very strong in the Apollo Domain arena. However, Hewlett-Packard is discontinuing Domain in favor of HP-UX, and Interbase is clearly aware of the need to transfer its Domain strength to HP-UX and other popular Unix platforms.

Important: This report contains the results of proprietary research. Reproduction in whole or in part is prohibited. For reprint information, call (617) 742-5200

Expanding to Medium-Scale Systems

Interbase plans to expand from its concentration on workstations and small systems into the medium-scale market, including multiprocessor platforms from vendors like Pyramid, Sequent, and Stratus. Interbase anticipates this market growing as fast as the workstation market, and, therefore, wants to play a role here. The company isn't targeting the large-scale end of the market, although one possibility is a port to supercomputers from Cray et al. The interesting scenario here is a supercomputer with hundreds of small workstations on the front end doing complex applications such as weather simulation or weather pattern analysis.

Lack of DOS or OS/2 Support Will Hurt

The big question for Interbase is OS/2, DOS, and PC LANs, none of which is currently supported by InterBase. On the client side, Interbase must implement multiple client support, including DOS and OS/2, if the company is serious about targeting more commercial environments. Since InterBase does not support DOS networking, third-party products that run on DOS must use terminal emulation to access InterBase today.

On the server side, the company states that it will move to OS/2 as the market warrants. We also think Interbase should consider a port to Novell's NetWare.

InterBase Platforms

Unix	Data General Aviion	HP/Apollo DN3xxx/4xxx/5xxx/10000	SCO Unix 386
	Digital MIPS/Ultrix	HP 9000	Silicon Graphics
	Digital VAX/Ultrix	IBM RS/6000	Sun 3, 4, Sparc
Proprietary	Digital VAX/VMS		

Illustration 2. InterBase runs on several of the most popular Unix platforms in addition to Digital's VAX/VMS. It does not currently run on DOS, OS/2, or PC LANs such as Novell's NetWare.

Peer-to-Peer Architecture

DISTRIBUTED PROCESSING. Interbase describes its product as peer-to-peer rather than client/server, viewing client/server as a specialization of the peer-to-peer model.

In the InterBase peer-to-peer model, any instance of InterBase can be either a client or a server, and clients talk to servers the same way servers communicate with other servers. An InterBase application can access databases throughout the network and may go through a server, depending on where the database is. No server, however, is required to access a local database, and an application can talk to multiple servers in a single transaction.

In contrast, Interbase uses client/server to mean:

- The client has one server, and it gets to any database through that server.
- The client must go through a server even to access local data.

A Router Determines if Data Is Locally or Remotely Stored

IMPLEMENTING PEER-TO-PEER. All database access statements and commands generated by an application are first translated into Binary Language Representation (BLR). Then they go through what Interbase calls the "Y-valve," or router. This determines where to send the command—to the local access method for local data or to the remote interface for remote database access. The Y-valve functions as the client interface and is the mechanism through which Interbase implements its peer-to-peer architecture. (See Illustration 3.) The server interface is Open Standard Relational Interface (OSRI), a superset of Digital's Digital Standard Relational Interface (DSRI).

The InterBase Architecture

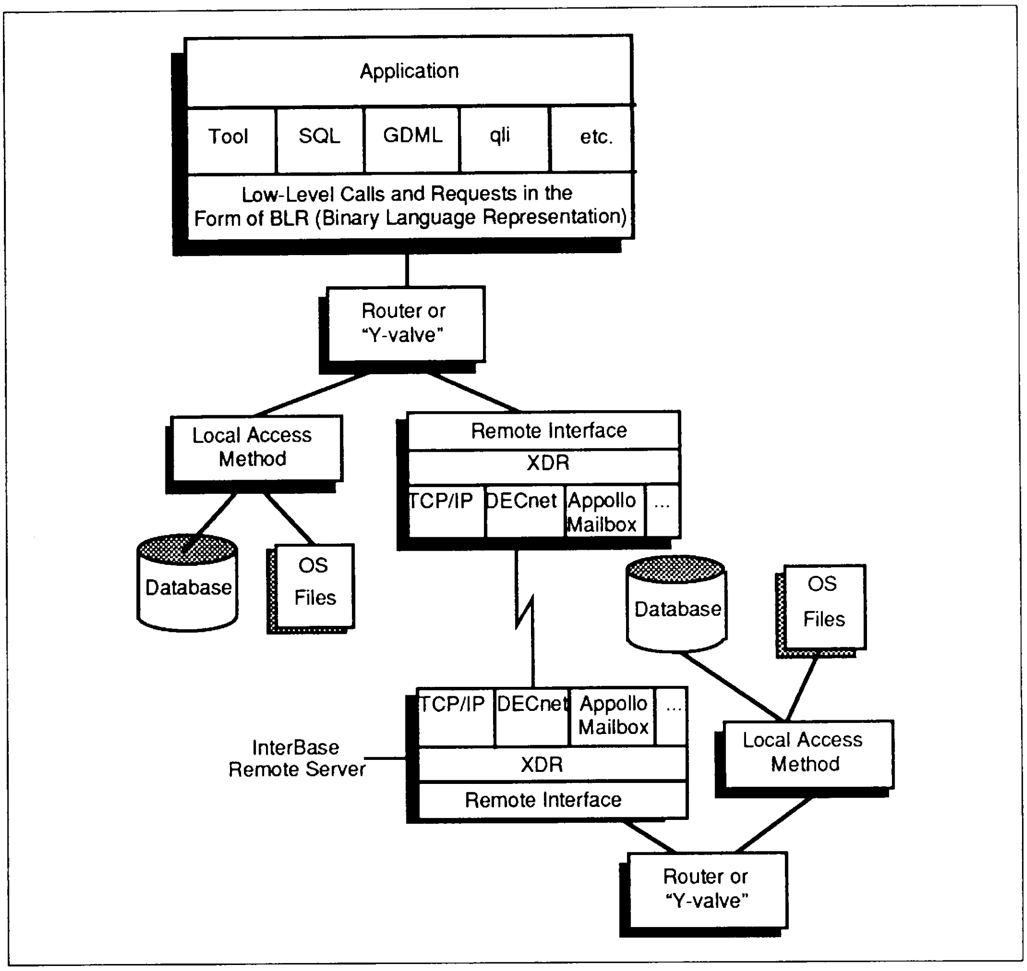


Illustration 3.

Server-per-User Is Standard on All Platforms

The standard InterBase server architecture spawns a database server for each client that attaches to a database. This server-per-client approach is the same as that of Oracle and Informix, and it allows InterBase to take advantage of multiprocessors.

Multithreaded/Multiclient Server Takes Advantage of OS Threading

On VMS and Apollo Domain, InterBase offers a multithreaded, multiclient server architecture. This single server handles all clients concurrently by using a threading capability at the operating system (OS) level. Different threads are created for multiple client tasks. While this is similar to Sybase's architecture, there is one major difference: Sybase does not depend on the operating platform for its threading capability. It has integrated many OS tasks into its DBMS server so that its multithreaded server can run on any platform. In contrast, InterBase is dependent on the OS vendor to provide a threading capability (although Interbase wrote its own threading facility for VMS). Without this threading capability, the InterBase server becomes single-threaded regardless of its internal architecture.

Missing Is a Multiclient Server Architecture for Multiprocessor Platforms

However, even with multithreading, this architecture cannot take advantage of multiprocessors since none of the operating systems allow multiple threads to run on different processors. Some OS vendors, like Apollo and Silicon Graphics, are working on the ability to schedule different threads on different processors. Interbase is hoping that it will be able to wait for the platform vendor to provide this rather than having to modify its own multithreaded architecture to accommodate multiprocessors.

Interbase Lacks the Ability to Tune Configurations

Some competitors, such as Ingres and Progress, offer a "tunable" server architecture where the number of servers and the number of clients per server are both configurable. Benefits here are the ability to control system resources required and the number of processors that the DBMS uses on a multiprocessor platform. Sybase has also just introduced its multiserver version for symmetric multiprocessors. Although Interbase is looking at this approach as an alternative, it considers sophisticated threading packages a better solution to the problem.

Support for Transaction Monitors?

The company has not yet decided on a strategy with regard to support for transaction monitors. And support for transaction monitors may not be relevant unless Interbase chooses to implement a multiserver version of its multithreaded architecture.

Multiprocessor Support vs. Shared Cache

TRADE-OFFS FOR MULTISERVER VS. MULTITHREADED. The benefit of the server-per-client architecture is the ability to use multiprocessors. The multiclient server provides shared cache/buffer space, which can enhance performance if the same set of clients uses the same sets of data regularly.

FUTURE MULTITHREADED PLATFORMS. Interbase is looking at other platforms for its multithreaded architecture. OS/2 has its own threading package, and, as we mentioned, Interbase is evaluating market demand for this environment. Sun is working on implementing what it calls "lightweight" processes in its operating system. While this is not a true threading package, it allows you to do what threading does without the overhead of a full process. Interbase is developing a multithreading version for Sun. Other platforms with threading on the horizon include Silicon Graphics, MIPS/Ultrix, and OSF. The OSF threading package will allow client threads to run on different processors, even those with single servers.

Data Access Languages

InterBase Offers Two Data Manipulation Languages: SQL and GDML

The InterBase developer can use either of two data manipulation languages (DMLs)—ANSI SQL or Interbase's proprietary language, called GDML. With SQL, the developer can take advantage of compliance with standards, portability of data access across different DBMSs, and existing expertise in writing SQL. GDML offers other advantages, such as broader application functionality and access to InterBase features that standard SQL doesn't understand. The developer can stick to GDML for writing an application, can mix GDML and SQL, or can use only SQL to access the database for portability.

GDML Offers Access to All InterBase Functionality

GDML is not a set of extensions to SQL but a separate database language, since GDML has its own version of all of the standard SQL statements plus a lot that SQL doesn't have. The GDML store, modify, read, and erase statements have the same functionality as the SQL insert, update, select, and delete statements. Both languages provide security and transaction management and can be embedded in a 3GL.

GDML adds the ability to use specific InterBase functionality not accessible from SQL—creating triggers, using event alerters and user-defined functions; manipulating arrays and BLOBs, etc. The company also emphasizes that GDML syntax is often easier to use than SQL. The GDML for-loop construct, for example, eliminates the need to use SQL cursors to retrieve multiple records. While SQL only supports a single transaction per user, GDML allows a user to have multiple concurrent transactions in process. SQL has no concept of locks, so any table or database locks must be requested with GDML. Other GDML advantages include form and menu handling and the ability to write recursive routines, which are often necessary in executing procedures such as a parts explosion.

GDML Supports Global Field Definitions

Within GDML, you can define global fields that are independent of a specific table. You can then reference the global field by name only when defining a table. (You can also give the global field a different name within each table.) Global fields can save time and effort and ensure consistency of field definition across tables. If the global field definition is modified, all fields based on that global definition will be changed as well.

Only local fields can be defined within SQL. Local fields are defined within the context of a table definition; if the developer wants to use the same field definition in multiple tables, the field must be fully defined within each table definition.

GDML Offers Additional Field Attributes

When defining fields with GDML, additional field attributes include:

- Validation criteria ("valid if ...")
- Edit string—an edit mask or display format used within qli (e.g., social security number formatted as xxx-xx-xxxx)
- Query header—specifies a column header for displaying query results in qli, InterBase's interactive language editor (you can have multiline headers)
- Query name—specifies an alternate field name for use in qli

Security Can Be Defined with SQL or GDML

SQL and GDML use different basic models for security. In SQL, no one but the creator can access a database table, view, etc. until permission is specifically granted. In GDML, anyone can access the database unless access is restricted. Another difference is that in SQL, access is granted or revoked by individual user name. There is no concept of groups of users in SQL. GDML, on the other hand, uses security classes to define classes of users and access rights. Whatever language you use to create a database object must also be used to define security on that object.

Interbase is Considering SQL Extensions

Interbase Software has explicitly chosen not to extend SQL but is currently rethinking this position. In the past, developers who wanted to use SQL would do so and then use GDML statements where necessary to access additional InterBase functionality. An example here is working with BLOBs. SQL doesn't understand BLOBs, but it can retrieve the BLOB ID, which is stored in the record. The developer can then use GDML statements to get and manipulate the BLOB itself. This mix-and-match at the statement level is relatively clean, and here GDML is essentially behaving as an extension to SQL. The recent addition of support for features like arrays and user-defined functions, however, cannot be handled the same way. These constructs must be embedded within the SQL statement, which requires extending SQL.

Interbase needs to extend SQL for another reason as well. Third-party tools vendors almost always use SQL as the language to access the backend DBMS. If SQL provides only ANSI-standard functionality and no access to InterBase-specific features, InterBase will have trouble competing with other DBMSs which have extended SQL. It is too risky for Interbase to depend on the tools vendor to extend its own environment to access special features within one vendor's DBMS.

Data Access Languages

InterBase Product Line and Pricing

InterBase RDBMS (these prices are for development versions)	
Single platform	$3,000—168,000
Network client (remote interface; includes all supported networking protocols)	$250—53,450
Super Server (InterBase RDBMS, network client, and remote server capability)	$3,000—216,000
InterBase toolkit—includes: Qli (interactive Query Language Interpreter) Gpre (all SQL/GDML preprocessors that run on a platform except C++ and Ada, which are optional) Pyxis (forms package) Fred (forms editor)	$1,250—74,200
Optional SQL/GDML Preprocessors (available on selected platforms)	
C++	$1,000—3,750
Ada	$1,000—30,250
ADS/Interbase	$3,000—108,000
Pictor (end user graphical query tool)	$2,795
Rdb Bridge for VAX/VMS (read/write gateway to Digital's Rdb)	$3,500—82,320

Illustration 4. Pricing for InterBase products varies according to platform.

Managing Performance in a Transaction Environment

Readers Never Block Writers in InterBase: A Performance Issue

MULTIGENERATIONAL RECORD ARCHITECTURE. In environments where database updates and complex data analysis must be performed concurrently, performance can become a serious issue. In most relational DBMSs, the ability to analyze, or read, a set of records and see a consistent view of the database (i.e., get the same answer for a repeated calculation—what is called a "repeatable read" level of data isolation) requires a shared read lock on all records involved for the duration of the transaction. This type of lock creates conflicts between readers and writers since no one can update any of the records until the reader releases the shared lock at the end of the transaction (an update requires an exclusive lock on a record).

InterBase solves this reader/writer conflict with a unique multigenerational record architecture that avoids the need for any read locks on the database. Readers always see a consistent and committed view of the database, even though changes may occur to one or more records while the read takes place, and readers never block writers. Only writers block other writers (and there is no way to eliminate this conflict). This architecture provides a high level of concurrency and throughput for InterBase.

Difference Records Make the Difference

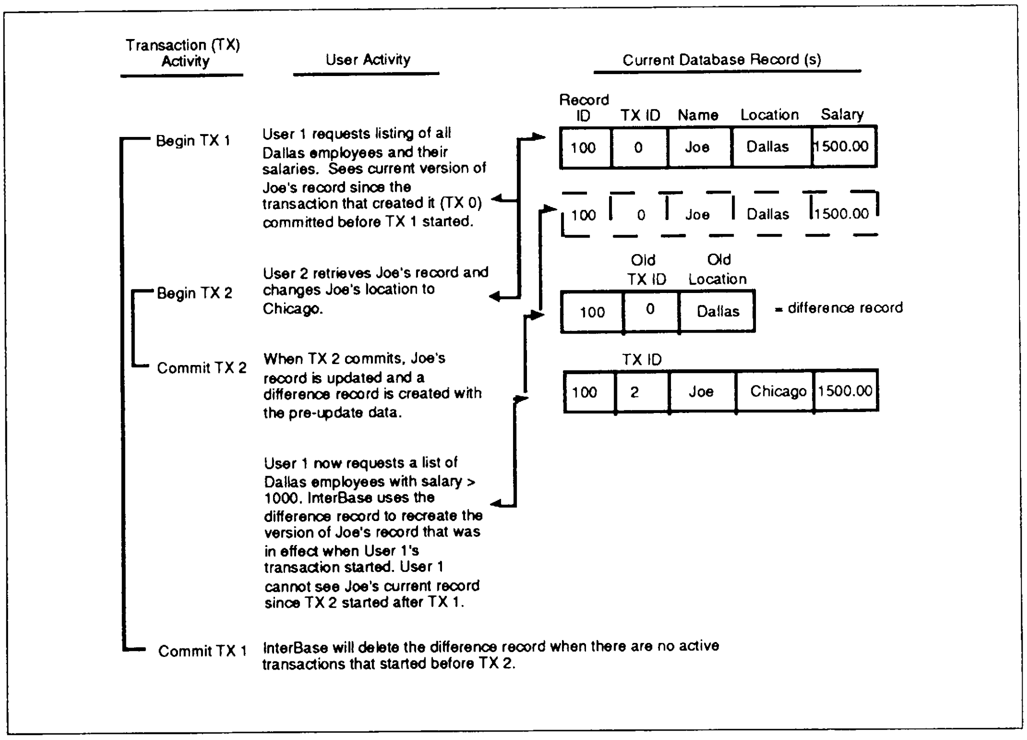
InterBase does this by creating what it calls a "difference record" for any update that takes place. The difference record identifies the record that changed, the transaction that changed it, and what data changed. InterBase then monitors active transactions and uses the difference records to make sure that each user sees the version of the database that was in effect when the user's transaction started. InterBase essentially recreates an old version of a database record from the current version using these backward difference records. (See Illustration 5.)

InterBase tries to keep difference records on the same database page as the current record. If it can do this, both the database record and the difference record can be written in a single write. Other DBMSs require two writes—one to the database and one to the short-term journal.

Managing Performance in a Transaction Environment

InterBase deletes difference records when they are no longer needed (i.e., there are no transactions active that started before the difference record was created). InterBase calls this "garbage collecting," and it takes place dynamically as users access records in the database.

Multigenerational Records in InterBase



User 1 requests listing of all Dallas employees and their salaries. Sees current version of Joe's record since the transaction that created it (TX 0) committed before TX 1 started.

User 2 retrieves Joe's record and changes Joe's location to Chicago.

When TX 2 commits, Joe's record is updated and a difference record is created with the pre-update data.

User 1 now requests a list of Dallas employees with salary > 1000. InterBase uses the difference record to recreate the version of Joe's record that was in effect when User 1's transaction started. User 1 cannot see Joe's current record since TX 2 started after TX 1.

InterBase will delete the difference record when there are no active transactions that started before TX 2.

Illustration 5. Here is a very simple example of how InterBase uses difference records to recreate earlier versions of records and maintain data consistency for readers while not blocking updates to the database.

Multigenerational Records Also Benefit Recovery

InterBase keeps track of transaction status—active, committed, aborted, and limbo (when involved in a two-phase commit)—on the transaction inventory page (TIP), in the database itself. The DBMS also uses the TIP page to provide instant recovery from failures. When the system comes back up, InterBase refers to the TIP page to identify transactions that were active when the failure occurred. These are backed out immediately by simply deleting any difference records created by the uncommitted transactions.

INDEXES ARE EXTENDED AS WELL. Indexes also have special components since they must, in some cases, point to multigenerational records.

The Competition Isn't There Yet

Competitors such as Informix and Sybase have devised other alternatives for reducing read/write conflicts in a database, but none can guarantee readers repeatable read consistency and not block updates at the same time. Oracle's "snapshot" model uses a somewhat similar multiversioning approach to ensure that a reader views committed and consistent data, yet doesn't block writers. When executing a query, the user always sees the version of the database that existed when the query started. However, this view of the database only exists for the duration of a single SQL statement. InterBase takes this one step further, guaranteeing repeatable read at the transaction level. (Oracle does use the snapshot model for "read only" transactions, but these must be declared as such and can include only queries.)

COMPRESSED RECORDS. InterBase uses a fixed length record but compresses it when storing the data on disk. It uses a simple byte-length encoding technique (repeating characters are compressed) to compress the data in all fields except BLOBs.

Sparse Bitmaps Speed Query Optimization

Like most of its competitors, InterBase has a cost-based query optimizer. As part of its query optimization, however, InterBase uses a technique called "sparse bitmaps." When a query is submitted to the database, InterBase looks at the appropriate index(es) and generates, for each criterion, a sparse bitmap of the database. A sparse bitmap shows what records meet the criteria and where the records are stored in the database. If there are multiple criteria in the query, InterBase compares (overlays) the bitmaps to find the intersection (in the case of "and" connectors) or the union (in the case of "or" connectors). This allows InterBase to efficiently retrieve records from the database. The major requirement is that the necessary indexes exist.

Support for Complex Data

InterBase Has Always Supported BLOBs

BLOBs. InterBase offers three levels of BLOB support: a BLOB data type, BLOB subtypes, and BLOB filters. *BLOB* stands for "basic large object," and, contrary to popular belief, Informix Software was not the inventor of the term. Interbase can claim credit here, introducing BLOBs in the first version of its RDBMS engine back in 1986. Unlike any of its competitors, InterBase has supported BLOB data types from the beginning, and it provides methods to extend the system so that applications and the DBMS can understand the data stored in a BLOB.

BLOB Data Type. A BLOB data type can store an unlimited amount of any digital data within the database. BLOBs can be used for images, voice, documents, binary data, etc. InterBase itself uses BLOBs to store information such as view definitions and access control lists in the data dictionary tables.

BLOBs Can Be Manipulated with GDML, but Not SQL

The database record itself contains an 8-byte BLOB ID. This is a pointer to the BLOB itself, which is stored elsewhere in the database. If you change a BLOB, the difference record maintains a pointer to the old version, and the old BLOB is kept until the difference record is deleted. The DBMS reads and writes BLOBs in sequential segments; segments can be any size up to 64K. As a reminder, SQL can access the BLOB ID, but not the BLOB itself. GDML must be used for that.

BLOB Subtypes. When defining a BLOB data type, you can assign it a subtype. The subtype is a user-defined code to identify the BLOB's format, and it is used by BLOB filters to convert BLOB data from one subtype to another. The only predefined BLOB subtype is "1" for text BLOBs.

With BLOB Filters, the DBMS Can More Intelligently Deal with BLOB Data

BLOB Filters. BLOB filters convert BLOB data from one format to another. A BLOB filter could be used to encrypt a C routine on storage and decrypt it on retrieval, to develop a compression algorithm for a BLOB data type, or to convert an image among different platforms for display and editing.

A BLOB filter is a subroutine written in a 3GL and included as part of a BLOB filter library that essentially becomes part of InterBase. If a user retrieves a BLOB as a subtype 5 and the BLOB is stored as subtype 3, the DBMS automatically looks at the filter library to see if there is a filter to convert from 3 to 5. If there is, InterBase applies the filter before giving the BLOB data to the user.

Creating the Filter Library Depends on the OS

The process of creating the filter library and making it part of InterBase depends on the operating system platform. BLOB filters are easiest to use on platforms that provide both shareable libraries and dynamic entry-point lookup. In this case, the developer writes the BLOB filters, links them into a filter library, and tells InterBase that the filter library exists.

Support for Complex Data

The filter library is a separate library and is not linked into the InterBase library (GDSLIB) or kernel. VMS and Apollo offer this capability.

Without dynamic entry-point lookup, the developer must also write a 3GL lookup routine that tells InterBase where to find specific filters in the filter library. Without either capability, the developer compiles the BLOB filter routine and then must link up a new InterBase library.

Multidimensional Arrays Extend Support for Complex Data

In Interbase's traditional engineering and scientific market, arrays are a very useful way to represent certain types of complex data and data relationships. Developers in these environments want the database to understand arrays, and InterBase now supports multidimensional arrays (up to 16 dimensions).

A simple example of data for which an array is useful is the following. Suppose you are measuring the volume of noise an airplane generates as it takes off. You install four measuring posts at 1,000-foot intervals in the area you want to measure, and each post has noise detectors at several standard heights. You want to measure the noise recorded by each detector at multiple time intervals. A three-dimensional array (time, post number, and height) could store all of the measurements in a single field in the record. If you made two measurements at each detector, you would need a four-dimensional array to capture all your data.

In the commercial business environment, a single-dimensional array could be used to capture monthly sales forecasts. Storing an associated exchange rate with each forecasted amount would require a two-dimensional array, and so on.

BENEFITS OF ARRAYS. Storing data elements in an array is an alternative to spreading them out over many fields (without arrays, users can end up building tables with thousands of fields), and to lumping the elements into a BLOB. In the first case, the data elements are more difficult to maintain as a cohesive set and will consume more system resources than an array. In the second case, the DBMS cannot distinguish among the data elements, so the application must interpret and manipulate the data.

Array elements can be any InterBase data type except BLOBs and other arrays. As we mentioned earlier, SQL cannot access arrays (although it can retrieve an array ID); only GDML can.

Parts of Arrays Can Be Manipulated as "Slices"

ARRAYS ARE STREAM BLOBS. An array data type is actually stored as a BLOB, but the BLOB is formatted differently than the BLOB data type. An array is a stream BLOB, and the BLOB data type is a segmented BLOB. Segmented BLOBs are read and written sequentially with no capability to access the data at a specific point in the BLOB. Stream BLOBs, on the other hand, offer random access to the data. This is why the user can manipulate parts of an array, called "slices." InterBase understands how to "get" and "put" slices of an array. In the above example, a slice might be all of the measurements for an individual noise detector.

The only other major independent Unix RDBMS that supports arrays is Progress, but a Progress array can be only a single dimension.

User-Defined Functions Are Used to Extend InterBase Functionality

USER-DEFINED FUNCTIONS. With user-defined functions (UDFs) the developer can extend InterBase's functionality, customizing the DBMS to understand new ways of manipulating data. UDFs allow the developer to centralize commonly used calculations or conversions, simplifying both coding and maintenance. Examples of UDFs include converting text from lower to upper case, calculating the greater of two numbers, converting miles to kilometers or feet to meters, and calculating the absolute value of a number. However, you do not have the syntactic shorthand of using an operator to invoke the function. For example, you can define a

UDF to define what it means to add two BLOBs together, but you can't use the expression "BLOB1 + BLOB2" and expect the DBMS to reference the function you defined.

UDFs are written in 3GL code and linked into a function library. The function library is created in exactly the same way as the filter library (see "BLOB Filters" above). Once defined, a UDF is part of the database and can be used within qli or within GDML commands. UDFs can also be part of a computed field or trigger definition. SQL cannot (yet) access UDFs.

Flexible Triggers Simplify Support for Referential Integrity and Business Rules

TRIGGERS. Triggers have become an important feature in RDBMSs to allow the database itself to enforce referential integrity and business rules. The major advantage of server-enforced triggers is that they maintain integrity regardless of how the user chooses to access the database. Triggers can also be used to create audit trails and to provide an updatable view.

InterBase offers extensive functionality in its implementation of triggers. A trigger executes a specific action when a record is inserted, updated, or deleted. You can have an unlimited number of triggers for a table, and each trigger can be defined as firing either before or after the database change (e.g., pre-update or post-update). For multiple triggers associated with one type of change, you also specify the order in which they fire simply by numbering them (you start with 0, a familiar convention in C programming). If multiple triggers of the same type have the same sequence number, the order of execution is random. The developer can also specify custom error messages for a trigger. Triggers can be modified, deleted, and deactivated.

Triggers cascade (i.e., triggers can call other triggers), and they can fire recursively (a trigger can end up calling itself). Triggers can operate only within a single database; a trigger on a table cannot affect data in another database. Some InterBase developers are using event alerters to allow triggers to span multiple databases.

All trigger actions are under full transaction control. If a transaction aborts, all trigger actions performed within the transaction will also be rolled back. Triggers can only be created and modified using GDML.

The Advantage over Sybase Triggers is Modularity

InterBase triggers offer essentially the same functionality as Sybase triggers, although the two products implement triggers somewhat differently. However, the modularity of InterBase triggers gives the developer more flexibility. Defining multiple triggers for a specific operation rather than one big trigger can simplify writing, testing, and maintenance of trigger code. (Sybase allows only three triggers per table, one each for insert, update, and delete.) Having both pre- and post-operation triggers is significant if it is easier to write the trigger code one way rather than the other. The ability to simply deactivate triggers also eases maintenance.

Ingres Rules Are Optional

In comparison to InterBase triggers, Ingres rules have two disadvantages. If you have multiple rules for a table (a likely situation, since Ingres rules are defined for a particular column and not for the table as a whole), the firing order is not defined and cannot be predicted. In addition, the customer must buy the optional Knowledge Management Extension to get the rules capability.

Event Alerters Automatically Notify Applications of Interesting Database Changes

Event alerters are designed to notify applications on a real-time basis of changes in the database, even across the network. The event manager is an integral part of the InterBase DBMS and eliminates the need for an application to constantly poll the database (which wastes CPU time) or to execute fixed-interval checking (which can miss significant changes).

InterBase is unique among its competitors in implementing event alerters. Although it is a relatively new feature, customers are already experimenting with events. Some are using events to allow triggers to span databases. Others are looking at events to disseminate information on database changes to applications across the network. The Philadelphia Stock Exchange, which recently chose InterBase as the foundation for part of its equity trading system, plans to use event alerters to track stock prices and to automatically notify specialists on the trading floor when to execute specific buy or sell orders. This will eliminate a huge volume of manual tracking for the specialist and will ensure that orders are executed on a timely basis.

Other examples of useful events are sounding an alarm when a temperature fluctuates more than 5 degrees or alerting a network manager when a component of the network gets too busy.

The DBMS Now Has an Event Manager and Event Tables

An event is a specific change to the database. It is defined by a trigger that includes instructions to "post" the event name when the event occurs. Posting an event alerts the InterBase event manager to check an event table for the database to see if the event name is there. The event table lists "events of interest"; if the event is there, it means that an application has asked to be notified when the event occurs, and the event manager will post the event. The application will then receive notification that the event has occurred, and it can do whatever processing is appropriate. If no one has registered an interest in the event, it is not posted. Event alerters can be synchronous or asynchronous.

It is important to note that an event isn't posted until the transaction that triggered the event commits. This avoids setting off events that cannot be rolled back if the transaction aborts. The processing triggered by the event cannot be under transaction control since the DBMS has no control over the application.

No Support for Stored Proceedures

Unlike Sybase, InterBase does not currently support stored procedures. The underlying mechanisms are there, but the company needs to add the appropriate user interface to access them.

Availability

Disk-Shadowing Enhances Availability

InterBase provides several features to enhance availability. It supports online backup, online database schema changes, and software-based disk-mirroring or -shadowing. Support for disk-mirroring is particularly important when fault-tolerance and high availability are important, and neither the hardware nor the operating system provides these capabilities. InterBase does disk-shadowing at the database level. In addition, the multigenerational record architecture allows the database to recover instantly from system failures; the database automatically backs out uncommitted transactions on system restart to create a consistent version of the database.

Distributed Database

InterBase Has Always Had Distributed Database Capability

DISTRIBUTED DATABASE. InterBase was an early supporter of distributed database functionality as an integral part of the DBMS, and it provides great flexibility in accessing multiple InterBase databases. This is important in a distributed computing environment. A user/application can open as many databases as desired within a single transaction; the number is limited only by operating system or network constraints. Interbase is a touch behind in supporting distributed queries but is as far along as anyone in offering distributed transaction management.

Distributed Database

Automatic Two-Phase Commit for Distributed Transaction Management

InterBase supports distributed transactions with an automatic two-phase commit (2PC). This ensures that all database updates within a single transaction commit or that none of them do. Version 3 of Interbase added automatic recovery from a failure during a 2PC process.

Progress also provides automatic 2PC as an integral part of its DBMS. Ingres offers automatic 2PC in its optional distributed database software, Ingres/Star 6.3.

InterBase Needs to Add Functionality in Processing Distributed Queries

InterBase also supports distributed queries (joins and unions across multiple databases), but not in a single command. Multiple language statements are required for distributed queries—nested for loops in the case of GDML, and nested cursors in the case of dynamic SQL and SQL statements contained in InterBase procedures. Here, the user controls the join order by the way the for loops or cursors are nested. InterBase does not yet have a distributed query optimizer, but is working on development in this area.

Location Transparency Is Not Automatic

Location transparency—the ability of the user/developer to access distributed databases without having to know where the data reside—is supported at the operating system level, not in the DBMS itself. InterBase uses OS facilities such as logical names on VMS and links on Unix. Once the user has "attached" (opened) a database, he or she automatically gets an aggregate data dictionary for all attached databases. This process can easily be simplified by the system administrator or database administrator (DBA).

Applications Can Transparently Access Operating System Files and Rdb Data

InterBase provides read/write access to any external flat files on supported operating systems (e.g., RMS on Digital's VMS and ISAM). Here, however, concurrency control and transaction management are not maintained across non-InterBase data. The VMS version of InterBase also includes read/write access to Rdb data. This works bidirectionally: InterBase applications can transparently access Rdb data; Rdb applications can access InterBase data. An Oracle gateway is marketed by InterBase partner Cognos, which sells InterBase as its StarBase product. This Oracle gateway (available on VMS only) uses Oracle's dynamic SQL.

Futures

ENHANCED DISTRIBUTED DATABASE AND MORE EXTENSIONS.

Interbase Software plans to enhance its DBMS in two primary areas. One is distributed database. A first step here will be to allow a single command to span multiple databases. Other enhancements we expect are true location transparency, a distributed query optimizer, and features to maintain local autonomy (e.g., name and data type resolution). Further out will be attention to features such as logical databases that span multiple physical databases (e.g., horizontal and vertical partitioning) and replicated data.

The second area involves continuing to extend the relational model. The company did not want to discuss specifics here, hinting only that this would include making InterBase's data modeling capability more powerful.

Application Development Tools

Lack of Good Development Tools Has Hampered Interbase

Lack of a good application development toolset has been a major InterBase weakness, and several of the developers we interviewed mentioned this. Interbase recognizes that it is difficult for a database vendor to be good at designing both the DBMS and development tools. Therefore, the company's strategy is to develop its own tools only where required (e.g., query and reporting tools), while forming close partnerships and integration with selected third-party application development toolsets. Two recent announcements added partnerships with Convergent Solutions, Incorporated (CSI) and Jyacc.

Convergent Solutions

Tightest Integration Will Be with ADS/InterBase

The closest relationship will be with CSI and ADS/InterBase. This is the CSI language-based CS/ADS toolset for developing complex applications packaged with an interface to InterBase. ADS/InterBase is available only from Interbase and looks as if it will take over the role as Interbase's primary application development toolset. CSI and Interbase are jointly developing a tight integration between the application development tools and the InterBase DBMS.

CSI Is a Recent Unix Convert

MOVING TO UNIX FROM CTOS. CSI is new to Unix. About eight months ago, CSI moved CS/ADS over from the Unisys/CTOS environment, where it was quite well known (22,000 licenses). CSI is looking to Interbase to help it get started in Unix, and ADS fills an obvious gap for Interbase.

Convergent Solutions, Inc.
100 Metro Park South
Laurence Harbor, NJ 08878
Phone: (908) 290-0090

AN INDEPENDENT ENVIRONMENT. Like Uniface (see Vol. 6, No. 6), the CS/ADS application development environment is designed to be independent of the operating system, network, DBMS, and user interface. In addition to InterBase, CS/ADS also supports ISAM files and Oracle. Coming later in the year are CS/ADS interfaces to Informix and Teradata Corporation's ShareBase. CSI also plans to support GUIs. Currently, OpenLook and Motif are partially supported (e.g., no support for radio buttons), and Windows 3.0 and Presentation Manager implementations are coming in the future. There currently is no interface between ADS and any CASE tools. In addition to Unix and CTOS, CSI plans to move into the DOS market for client/server applications (the ADS client runs on DOS now), and to VAX/VMS.

ADS/InterBase

Providing 4GL Access to InterBase Functionality

PRODUCT HIGHLIGHTS. ADS/InterBase centers on a block-structured, procedural 4GL and an environment that provides a high level of abstraction for the application developer. ADS uses late binding techniques, so the developer can design applications without worrying about what the data look like or how they are formatted. What CSI calls "prebuilt default paradigms" allow the developer to define scrolling areas, menus, byte streams, and other application components without dropping into a 3GL. These components can be nested as well (e.g., a scrolling area within a scrolling area).

ADS supports dynamic relationships based on data values (e.g., whether or not there is a corresponding entry in another table depends on the value in a field), cross-references objects (e.g., forms used within procedures), has its own text editor, and offers a WYSIWYG screen painter for forms and report layouts.

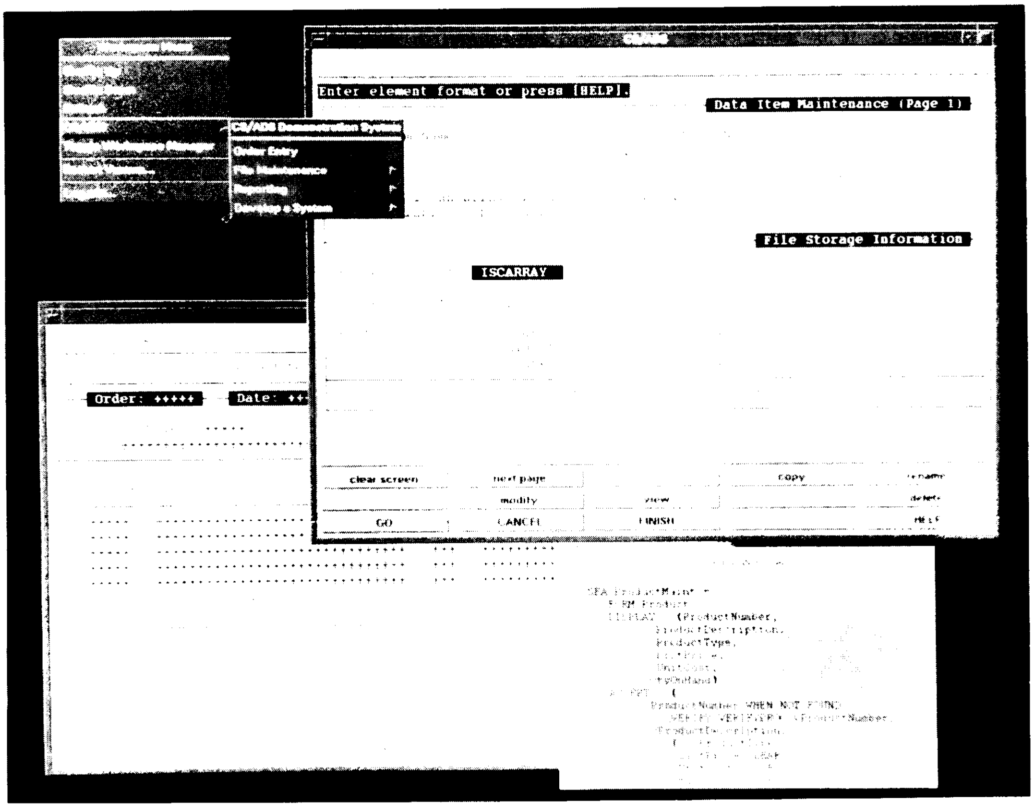
DATA DEFINITION AND ACCESS. The developer defines the database using native DBMS tools (available on the ADS menu by default) or the ADS SQL editor. The DBMS's native SQL is used for database access, which means the user can use extensions to SQL. It is possible also to use database-specific functionality and still have portable applications. For example, prefacing a statement with "interbase." will allow you to use an InterBase-specific SQL command. Using "db." as the preface means the command is portable and can be passed to any DBMS.

EXTENDING THE DBMS DATA DICTIONARY. ADS/InterBase has its own active data dictionary that is an extension, rather than a duplicate, of the DBMS data dictionary. (See Illustration 6). While there is some duplicate functionality between the dictionaries (for example, data validation can be defined in both the front end and the back end), the positive side of this is flexibility for the developer. In general, database objects are defined in the DBMS, and application objects are added in ADS/InterBase. The ADS active data dictionary is then accessed by the application at run time. This approach eliminates the problem of duplicate dictionary maintenance.

The ADS dictionary is stored as an ISAM file. In the future, the user will have the option to store the dictionary in the native DBMS table structure. Two things missing from the data dictionary are referential integrity and support for an application as an object.

INTEGRATION WITH INTERBASE. CSI and InterBase have a three-phase roll-out for integrating InterBase-specific features in the ADS 4GL. In the first release, the emphasis is on BLOBs. Functions include create, copy, or delete BLOBs; edit text BLOBs; and support for BLOB filters. Event alerters, array data, and GDML statements can only be used through 3GL code. (ADS already supports two-dimensional arrays). The second release will add multiple concurrent database support. The third step will be responding to event alerters and manipulating arrays in the 4GL, tighter integration between the data dictionaries, and managing concurrent transactions.

ADS/InterBase



Enter element format or press [HELP].

Data Item Maintenance (Page 1)

File Storage Information

ISCARRAY

Order: +++++

Date: ++

```
    ProductNumber,
    ProductDescription,
    ProductType,
    [illegible]
    UnitCost,
    QtyOnHand)
  (
    ProductNumber WHEN NOT FOUND
      VERIFY VERIFY [illegible] ProductNumber
    ProductDescription
```

Illustration 6. This screen shot shows a data element definition (part of the ADS data dictionary), a procedure, and a screen form in three overlapping windows in the Motif version of ADS/Interbase.

AVAILABILITY. ADS/InterBase will be available on Sun, IBM RS/6000, and Apollo this summer, and on all other InterBase Unix platforms by the end of the year. VMS will follow.

Jyacc

Jyacc
116 John Street
New York, NY 10038
Phone: (212) 608-6753

Jyacc got its start in high-tech consulting in 1978. The company introduced its first tool, Jyacc Form Maker, a screen and window manager, in 1984. This provided the core technology for JAM (Jyacc Application Manager), which was released in 1987.

JAM FOR INTERBASE. Jyacc's JAM is a well-known toolset for developing highly interactive applications. While Interbase will not sell JAM directly, the two companies will participate in joint marketing efforts, and Interbase will certify JAM's interface to the InterBase RDBMS.

ENVIRONMENT INDEPENDENCE. JAM now runs on a multitude of platforms. VMS and Unix are the largest markets (25 to 30 percent of revenues each), followed by DOS at 20 percent. Over 7,500 JAM licenses have been issued. One differentiator is the fact that Jyacc doesn't charge for run-time JAM licenses. Several hardware vendors have included JAM as part of their core technology and sell the product as OEMs. Stratus, Hewlett-Packard, and AT&T are examples.

JAM Already Supports All of the Popular RDBMSs

JAM also supports 14 RDBMSs on the back end with more to come. These include Oracle, Sybase, Rdb, Ingres, Informix, and Progress in addition to the recently announced InterBase connection. To be supported, a database doesn't have to be SQL-based, but it has to have a query language.

Full Support for Windows and Motif, with Others Coming

Jyacc is committed to GUI independence and has already fully implemented support for Motif and Windows 3.0 (although these versions of JAM are still in controlled release). JAM also can provide a character-based and block-mode interface. Coming are OpenLook and Macintosh versions.

Building a Sophisticated Interface for InterBase Applications

FOCUSING ON THE INTERFACE. Aimed at the professional developer, JAM is designed for building highly interactive applications with a sophisticated user interface. The company states that JAM can provide any user interface the client wants. For example, a JAM application can display multiple transactions or multiple applications on the screen concurrently. This type of functionality is very much in tune with the types of complex applications for which InterBase is designed. JAM supports a forms- and event-driven application model; it is not language driven.

The bottom line is that JAM doesn't try to do everything. It does the front-end user interface very well and adds hooks to both database functionality and 3GL routines. (C is available on all platforms; other 3GLs depend on the platform.) The product is geared to give the developer control over database access and access to DBMS functions through SQL.

USING JAM. Typically, the developer first uses the JAM authoring tools to create the user interface. No 4GL or 3GL code is required. The focus here is on fast prototyping and the ability to grasp a clear understanding of user requirements from the beginning. A few of the features supported by JAM include sibling windows, simulated scroll bars on character interfaces, and an unlimited number of open windows on the screen.

The next step is to add any event-driven procedures with the JAM Procedural Language (JPL). JPL is an interpreted language, and procedures are attached to events, forms, and applications. JPL can also be used to create named procedures and, therefore, reusable code. JPL includes flow control constructs like do loops and if/then/else.

Data Dictionary. Today, JAM has its own data dictionary and does not interface with that of the DBMS. In the future, Jyacc will integrate the dictionaries.

INTEGRATION WITH INTERBASE. Like CSI, Jyacc is fully committed to extending its product to access InterBase features. In the first release, since JAM uses native SQL to access the database and Interbase hasn't extended SQL to access InterBase-specific features, the developer can't use these features without dropping into a 3GL. It is possible to write a 3GL program that uses either OSRI calls or GDML commands to access the database, and to attach the procedure to a JAM form or call it from within JPL. So JAM is extendable using a 3GL.

The second release will provide support for BLOBs, event alerters, and arrays from within JAM.

JAM's New Database Interface Architecture Debuts with InterBase Support

New Architecture. With the introduction of JAM for InterBase, Jyacc has also rolled out a completely rewritten version of its generic database interface software. Jyacc is positioning this architecture to provide an open interface to multiple DBMSs. To better support all the features of the many DBMSs that JAM supports, the interface software has been split into two layers: a core set of application services and a binding layer. The core services include a parser for dealing with SQL and DBMS functions. It knows nothing about a specific database, and it is extensible to incorporate special DBMS features. The binding layer does the translation between the core services layer and the DBMS itself.

Marketing Strategy

Moving into Sybase Territory

COMPETITION. Interbase competes primarily with Sybase, since it is the closest to InterBase in terms of DBMS functionality. Interbase is also beginning to overlap Sybase in its target markets as well—financial applications, telecommunications, and government. Oracle is always on the competitor list, but Interbase doesn't see Oracle fighting hard in Interbase's primary target markets. Interbase also competes with Ingres, Informix, and Empress.

DISTRIBUTION CHANNELS. Interbase Software's primary sales channel is its direct sales force, which accounts for 65 percent of sales. Direct sales currently covers both end-user and VAR sales.

Entree to the International Market is a Big Ashton-Tate Benefit

NO INTERNATIONAL PRESENCE. A major marketing issue for Interbase is developing an international presence and distribution strategy. The company is virtually invisible outside the United States right now and is counting on Ashton-Tate for heavy assistance here. Seventy percent of Ashton-Tate's revenues come from outside the United States (dBase's share of the market is still strong in Europe in particular), and every major country has an Ashton-Tate representative. Interbase plans to work with the current Ashton-Tate distribution network to develop its own international distribution channel.

The company is focusing on Germany, Japan, the United Kingdom, and Canada this year. InterBase supports 8-bit data currently and has externalized components such as message files in preparation for translation to other languages. German is planned for this fall, followed by Kanji support by the end of the year.

OEMs. Interbase wants to increase its OEM support and is working with various hardware vendors to generate joint marketing efforts. The company is looking at platform vendors that provide one or more of the following: a good match in terms of technology and product compatibility, a good match in target market segment, a role as a client to VAXs to take advantage of InterBase's Rdb/RMS connection (and the subsequent opportunity to make server-replacement sales), a role in conversion from older, proprietary DBMS technology via gateways. (Interbase can build a gateway to any SQL-based DBMS that supports dynamic SQL.)

Currently, Cognos is the company's only OEM. Cognos sells InterBase as its StarBase DBMS, packaging it with Powerhouse for building applications. The StarBase version of InterBase tends to lag the latest InterBase version by a short time to allow for integration with Powerhouse. There are also some InterBase features that Cognos has chosen not to support, such as arrays.

Conclusion

Support for Complex Data

Interbase Software has some excellent database technology. Its suite of support for complex data—BLOBs, BLOB filters, arrays, and user-defined functions—plus support for event alerters set InterBase apart in the Unix RDBMS market. The company clearly understands the requirements for OLTP and for managing complex data, as well as the challenges inherent in combining the two in a single environment. But unless the company can successfully communicate the benefits of its technology, it will continue to have visibility problems.

Alignment with Tools Vendors Is Key Success Factor

The availability of good application development tools makes InterBase much more accessible for developers interested in high-level development tools (rather than a 3GL) and graphical user interfaces for applications. For a relatively small DBMS company, alignment with tools vendors who support other DBMSs as well can relieve some of the pressure the customer inevitably feels in choosing an unknown vendor on which to build mission-critical applications. An approach like, "Try us, you'll be glad you did. But if not, you aren't stuck," can make it somewhat easier to get a foot in the door.

Enhancements that will help Interbase keep up are implementation of stored procedures, extensions to SQL for access to InterBase features, support for DOS and OS/2 clients, and a more complete implementation of distributed database functionality. The company also recognizes the need to develop an international presence.

Impact of Borland?

It remains to be seen what impact the Ashton-Tate/Borland acquisition will have on Interbase and its future success in the Unix RDBMS market. The strengths are there, but it is easy to be skeptical of acquisitions in the computer industry. InterBase certainly merits more attention than has come its way in the past. It is a good example of how to both support and extend the relational model in a meaningful way to meet real business application requirements. If Interbase can increase its visibility enough to emerge from its position as a niche player in the DBMS market, it will certainly heighten the pressure on the competition. ◐

InterBase Features Chart

DBMS Product	InterBase Version 3.2
Architecture	
Client/server	Yes
Multiserver	Yes; Interbase also offers a multiclient, multithreaded server option on HP/Apollo and VAX/VMS
Maximum number of servers/system	No limit
Maximum number of users/server	One for multiserver; no limit for single server architecture
Maximum number of users/system	No limit
Support for multiprocessors	Yes
Open architecture (APIs available)	Yes—OSRI (documented low-level API; superset of Digital's DSRI), dynamic SQL
Underlying file structure	Unix files; raw input/output is optional
Database parameters	
Database size	No limit
Databases/server	No limit
Tables/database	64,000
Rows/database	No limit
Row size	64K excluding BLOBs
Fields (columns)/row	16K
Indexes/database	No limit
Databases connected to a client	No limit
Maximum number of tables referenced in a single query	No limit
Maximum number of databases referenced in a single query	One; nested queries can access an unlimited number of databases
User interface	
Menu bypass	Yes
Contextual help	No; yes for ADS/Interbase
Tutorial	No; yes for ADS/Interbase
Ability to customize standard menus	No; yes for ADS/Interbase
Support for color	No; yes for ADS/Interbase
Support for graphical user interface	Yes—SunView, X11, DECwindows, Apollo Display Manager; ADS/Interbase supports OSF/Motif and OpenLook
Data types	
Character (fixed/variable length)	Yes—maximum length of each is 32K (variable length character field can be used for small BLOBs up to 32K)
Integer	Yes
Decimal	Yes
Float	Yes
Logical	No
Currency	Yes, with scaled integer
Date/time	Yes
Long text	Yes
Binary (fixed/variable length)	Yes; BLOB data type; unlimited
Image	Yes, BLOB data type; unlimited
Serial	Yes
Support for arrays	Yes; up to 16 dimensions

InterBase Features Chart (continued)

Data types (continued)	
Support for nulls	Yes
Ability to extend base data types	Yes (global fields or domains)
Ability to define new data types	No
Ability to define functions/operators for new data types	No, but can define functions (user- defined functions or UDFs) for existing data types
BLOB filters	Yes
B-tree indexing	
Maximum number of indexes	64 per table
Maximum number of fields/index	No limit
Maximum size of index key	255 bytes
Order options	Ascending, descending
Unique index	Yes
Clustered index	No; sparse bitmaps provide some of the same capability
Other file access methods (hash, etc.)	No
Screen forms	Created with Fred and then used with Qli or a 3GL; with ADS/Interbase, forms are created with the screen painter or the 4GL
Default form generator	Yes
Customized	Yes
Multiple tables/form	Yes
Multiple screens/form	No; yes for ADS/Interbase
Embedded processing (if-then-else, display aggregates)	Yes
Field attributes on forms	
Case conversion	Yes
Default value	Yes
Required value	Yes
Acceptable values	Yes
Verification (enter data twice)	No; yes for ADS/Interbase
Formatting of data	Yes
Calculated values	Yes
Display only (no entry/update)	Yes
Hidden	No; yes for ADS/Interbase
Prompt (for data entry)	Yes
Error message	Yes
Customized help	No; yes for ADS/Interbase
Video display	No; yes for ADS/Interbase
Ability to change field attributes dynamically	Yes
Query-By-Forms	Qli (requires some coding); ADS/InterBase
Exact match	Yes
Relational operators	Yes
Ranges	Yes
List of values	Yes
Wildcards	Yes
Maximum/minimum values	Yes
Print query results	Yes
Pass results to report writer	No for Qli; yes for ADS/Interbase
Text search	Yes

InterBase Features Chart (continued)

SQL	Interbase also provides its own proprietary language, GDML
Standard SQL statements	
Data definition language (DDL)	Yes
Data manipulation language (DML)	Yes
Extensions to SQL	
Commit/rollback transactions	Yes
Execute operating system commands	Yes, via GDML
Load/unload data to/from ASCII file	Yes, via GDML
Additional data definition statements	Yes, via GDML
Control-of-flow logic	Yes, via GDML
Outer join	No; database supports this, but syntax not yet available in SQL/GDML
Support for SQL precompilers (embedded SQL)	Yes; C, Cobol, Fortran, Basic, PL/1, Pascal, Ada, C++, depending on platform
Support for dynamic SQL	Yes
Support for call-level interface to database	Yes, OSRI (superset of Digital's DSRI)
Can create new table with query results	Yes
Stored queries	Yes
Case-insensitive (e.g., field names)	Yes
Can call C routines	Yes, through BLOB filter or user- defined function (UDF)
How create SQL queries/statements	Interactive SQL editor, Qli
Query optimizer	Yes; cost-based
Syntax-independent performance	Yes
Uses table statistics	Yes
Minimum/maximum value	Yes
Average count per value	Yes
Distribution of values	No
Explain capability	Yes
Report writer	Qli; ADS/Interbase
Nonprocedural	Both procedural and nonprocedural; no for ADS/Interbase
Default report generator	Yes; no for ADS/Interbase
Interactive report generator using screen forms	No; yes for ADS/Interbase
Interactive debugging	No; yes for ADS/Interbase
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)	No; yes for ADS/Interbase
User variables	Yes
Prompt for input variables at run-time	Yes

InterBase Features Chart (continued)

Application development tools	
4GL	GDML and ADS/Interbase
Application generator	No; yes for ADS/Interbase
Ability to design application menus	No; yes for ADS/Interbase
Default menu generator	No; yes for ADS/Interbase
Custom help	No; yes for ADS/Interbase
Integrity	
Transaction logging	Yes, journaling
Commit/rollback transaction	Yes
Roll forward	Yes
Referential integrity in data dictionary	Yes, with triggers
Field validation in data dictionary	Yes
Support for business rules	Yes, with triggers
Triggers	Yes
Scope	Single database
Database operations supported	Pre- and post-insert, update, delete
Precompiled	Yes
Preoptimized	No
Number per table	No limit
Can specify execution order	Yes
Forward chaining (cascade)	Yes
Recursive	Yes
Can be deactivated	Yes
Event alerters	Yes
Stored procedures	No
Precompiled	n/a
Preoptimized	n/a
Can be nested	n/a
Concurrency control	Multigenerational record architecture ensures availability of data regardless of concurrent usage
Locking levels:	
Database	Can be requested
Table	Can be requested
Row	Yes
Page	No
Data isolation levels	Multigenerational records provide read consistency, committed read, cursor stability, repeatable read without requiring read locks
Lock types	Shared, protect, exclusive
Database security	
Login password	No
Multilevel access control	
User	Yes
Group	Yes
Application	No
Database-level access	Yes
Table-level access	Yes
Row-level access	Yes
Field-level access	Yes

InterBase Features Chart (continued)

Database security (continued)	
Access by time of day	No
Access by location (workstation)	Yes
Ability to define resource limits on user queries	No
Availability	
Online backup	Yes
Online database changes	Yes
Software-based disk mirroring	Yes
Raw input/output	Yes; optional
I/O reduction techniques	
Fast commit	No
Group commit	No
Parallel checkpointing on multiprocessor systems	No
Database can span multiple physical devices (disks)	Yes
Network support	TCP/IP, DECnet, Apollo Domain
Import/export capability	
Import formats	ASCII delimited, fixed length
Export formats	ASCII delimited, fixed length
Distributed database capability	Included with InterBase RDBMS
Support for partitioned tables	
Horizontal partitioning of tables	No
Vertical partitioning of tables	No
Location transparency	Can be implemented at the operating system level
Distributed query processing	Yes; distributed queries require multiple statements
Distributed query optimizer	No; coming in the future
Distributed transaction processing (two-phase commit)	Yes; automatic
Support for data replication	No; coming in the future
Access to heterogeneous databases	Yes; read/write access to Digital's Rdb and RMS
Maximum number of simultaneously connected databases	No limit
International language support	Support for German and Kanji coming in late 1991
Upper/lower case conversion	No; coming in late 1991
Sorting/collating sequences	No; coming in late 1991
Error messages	No; coming in late 1991
2-byte character set	No; coming in late 1991
Translated documentation	No; coming in late 1991

Illustration 8.

Next month's *Unix in the Office* will address
Uniplex's Strategy for the '90s.

For reprint information on articles appearing in this issue,
please contact Donald Baillargeon at (617) 742-5200, extension 117.

Open Systems: Analysis, Issues, & Opinions

SOFTWARE TRENDS

The Old Shrinkwrap Dilemma

July was an interesting month for open systems watchers. A lot has been happening, and our offices have been flooded with visits from many well-known and some less-well-known software vendors. Software is definitely growing in stature. It's a good thing, too. Fast, glitzy hardware is fine, but only if you have applications to run on it. This issue is becoming especially important in the commercial marketplace.

CONCERNS ABOUT UNIX SOFTWARE. I am growing increasingly concerned about the future of the Unix industry. For example, I got a call the other day from the Information System director of a small manufacturing company in the Midwest. He has been using Unix for several years and really prefers it to DOS. He is at a decision point. Should he return to DOS in order to get the accounting software he needs to run his business or are there software packages for Unix that will get the job done? If he finds them, will he have to pay three times as much for the Unix software?

I've been hearing this same concern for years. Until business software is easily and readily available to meet routine business concerns, users will keep clear. And they probably should.

WHAT WILL CHANGE THIS SITUATION? We anticipate that Application Neutral Distribution Format (ANDF) from the Open Software Foundation (OSF) could make it much easier for vertical application suppliers to economically move their applications to a large variety of hardware platforms. In brief, ANDF isolates all of the hardware-specific items into what is called an installer. The rest of the code is the same no matter which operating system or hardware platform it is running on. An intermediate C compiler is used to move the code to various platforms. Early indications are that the ANDF compiler may actually be more efficient than traditional compilers. Many system and software vendors have been skeptical about the viability of this technology. We predict that it could be the pleasant surprise of the decade.

Unix International: Finding Its Voice?

We've spent the last few years trying to figure out just what Unix International's (UI's) raison d'etre was going to be in this decade. Early on, its role was clearly defined: UI was the cheering squad for AT&T's System V. Indeed, this role was a reaction to the computer industry's demand that AT&T open its development and requirements process to the rest of the industry. With this backdrop, Unix International formed—after the creation of the Open Software Foundation—to provide direct feedback on requirements and actual specifications to AT&T.

COMPARISONS WITH OSF. In some respects, UI has been unable to form a unique identity. The areas that it has focused on happen to be the same areas where OSF has already begun or completed a Request For Technology. Therefore, even if UI has some valuable contributions to the area, its pronouncements are viewed as reactions to OSF rather than as leading edge. A good example of this can be seen in Unix International's latest area of endeavor: distributed computing via its newly announced program, Atlas. Atlas is intended to provide a comprehensive distributed computing platform that encompasses both DCE and ONC. We believe that the goal is a good one—one that is required by users.

TACTICAL ERRORS. However, some of the tactics UI is using to bring its platform forward have us puzzled. For example, it contends that DCE does not provide support for SAA, DECnet, or PCs. This is, in fact, not true. Both IBM and Digital are working hard to make DCE a key component of their networking environments. We don't expect that Unix International vendors will be able to deliver this capability any faster than Digital or IBM. Unix International also has an ambitious goal of providing Application Programming Interfaces (APIs) to all layers of the OSI model. We expect that it will take a lot of work by both Unix Systems Laboratories and UI vendor members to make this a reality. It is not a short-term solution. We also suspect that, by the time USL and UI members succeed at this, OSF members will have the same capabilities.

OBJECT MANAGEMENT IS A PRIORITY. Unix International is also making a big push for the use of object-oriented technology. It has joined the Object Management Group (OMG) and has added object management to its functionality roadmap. This is another move in the right direction, but we suspect that delivering object management will be harder than UI suspects. Object Management is complex technology and requires agreement by most of the major players in the industry. The Object Management Group represents the best opportunity today for a solution, but the timing may not be right for the OMG's success. Given the highly charged political climate in the computer industry, that much cooperation may not be forthcoming. —*J. Hurwitz*

SILICON GRAPHICS

A Software Vendor in Disguise

Silicon Graphics Incorporated (SGI) is one of those companies that is so tightly associated with the technical workstation market that it is often overlooked by users looking for commercial computing platforms. We suspect that, if Silicon Graphics gets its act together and makes a play for the commercial market, it could be extremely successful. The company has an excellent understanding of distributed computing requirements, produces some of the best graphics tools the industry has to offer, and knows how to produce snappy hardware. SGI's most recent announcement demonstrates its potential.

Hardware: Hot Technology, Hot Pricing

RISC WORKSTATIONS. The piece of SGI's recent announcements that is getting the most attention is its new workstations. The Power series is a scalable line of RISC systems that begins at 30 MIPS and goes up to 286 MIPS. The low end of the family, called IRIS Indigo, starts at $7995 for a diskless machine with 8MB of memory and a 16-inch monitor. It includes the same graphics capabilities as the high-end machines. SGI rates this machine at 30 MIPS, 4.2 MFLOPS, and 26 Specmarks. A configuration with a 16-inch color monitor, 16MB of memory, and 432MB of disk storage will sell for $12,500.

FILE SERVERS. In addition to workstations, SGI is also announcing a series of file servers based on the same technology. Configurations of these servers begin at under $35,000 and include 16MB of memory, 1.7GB of online disk storage, and network backup software. Like Digital, SGI uses the MIPS processors as its technology base. Indigo is based on MIPS R3000A processor. And, again like Digital, SGI is making noise about coming out with an ACE-compatible system. How is SGI distinguishing itself in the market? Primarily by emphasizing its graphics capabilities. These include such features as texture-mapping, alpha-blending, accumulation buffering, anti-aliasing, and depth cue. Virtual 24-bit dithering is another built-in feature.

TARGETING MULTIMEDIA AUTHORING. One target market for SGI is multimedia authoring, which has traditionally been done on a customized Macintosh. SGI is offering a Video Bus that will allow an optional live video board to be added for less than $2,000. For this market, SGI has also incorporated audio input and output (including microphone, headphones, analog line-in, analog line-out, and digital in and out). It also supports live or single-frame video in and out, and composite formats including NTSC, PAL, and SVSH. It will be capable of single-frame, 24-bit output for animation.To assist these power-hungry developers, SGI is offering a video library and toolkits. The toolkits will be developed in conjunction with third party developers who primarily have worked in the Macintosh market.

TRADITIONAL TARGET MARKETS. The multimedia market is not SGI's only avenue. As one might expect, the company is also targeting its traditional customer base in CAD/CAM/CAE, Chemistry, and Geosciences.

A Software Company, Too

PROVIDING CASE SOFTWARE. We believe that SGI is much more of a software company than anyone gives it credit for. For example, it has implemented a Computer Aided Software Engineering (CASE) framework called CASEVision that integrates a series of third-party CASE tools such as IDE's Analysis and Design, Frame and Interleaf for documentation; Project Management software such as VUE and Masterplan; and revision, control, and configuration management from Aide de Camp. These are integrated with CASEVision's own lower-level tools such as debuggers, performance analyzers, and languages (Fortran, C, C++, and Ada). Another SGI tool, CodeVision, allows applications to be developed via a visual interface. One of the most impressive aspects of the new software is that CodeVision allows developers to create applications that take advantage of a multiprocessing environment

IRIS EXPLORER: A TOOL FOR CREATING INTERACTIVE APPLICATIONS. Another example of SGI's concentration on software is its newly announced IRIS Explorer.

This tool is one of the first for the emerging visualization market. The idea behind visualization is to allow users (initially, in scientific areas and, long term, in business areas) to understand complex problems by looking at physical renditions from a variety of perspectives. The most obvious example might be the visualization of complex molecular models. A business application might be the ability to visualize a complex network configuration.

Conclusion: Software Could Be the Ticket to Success

SGI is a company that will be able to compete in the fast-paced technical workstation market. It knows better than to try to compete on price and hardware alone. With its new announcements, it will go head-to-head with Sun's new SPARCstation family as well as competing with the rest of the industry (IBM, Hewlett-Packard, et al.). Therefore, SGI's ability to be a leading edge software supplier is the only way it will survive in the RISC jungle. If SGI can take this advantage, apply it to the commercial market, and even port its excellent tools technology to a variety of platforms, it may emerge as a powerful player in the coming years.

—J. Hurwitz

INDUSTRY PSYCHOLOGY

Reflections on Corporate Personality

Did you ever notice that companies have personalities (kind of like people)? I often find it helpful in understanding a company to personalize the organization. I thought it might be interesting to apply a little psychology (child psychology, in the case of the computer industry) to some of the players in this market.

SUN MICROSYSTEMS. Ever since Sun Microsystems became the workstation leader, every company trying to get a piece of the action has taken one or two potshots at it. After all, the industry leader is always the target. Given the brashness and confidence of Sun's management, it is easy to assume that Sun expects this and, in fact, enjoys the sparring. After spending some time with Sun's executives, we are beginning to have a different perspective on this powerful player. In reality, Sun takes the attacks on its products and strategy quite personally. This accounts for its stubbornness and reluctance to join groups like OSF, which it believes were formed specifically to destroy Sun.

The Envy of All. I picture Sun as the "big man on campus." He is attractive, always gets any date he wants, is class president, and has started a side business that earns him a lot of money. Needless to say, he is the envy of everyone. At first, everyone talks about what an incredible person he is. After a while, others, jealous of his fame and fortune, are eager to prove that he isn't as grand as he seems. In reality, our big man on campus is also not so sure he's as grand as he puts on. Underneath, he's as insecure as the rest of his buddies. The unprovoked attacks hurt badly. Over time, he stops trying to be friendly. He becomes more determined than ever that he isn't going to let them ruin his life. He therefore isolates himself from the rest of his peers and goes it alone. Anything that happens along the way he assumes is a personal attack.

IBM. While it isn't possible in this short amount of space to cover the personalities of all my favorite vendors, I thought it would be interesting to look at IBM's personality in contrast to Sun's.

Of the Privileged Class. While Sun Microsystems is uncomfortable being the envy of its peers, IBM, in contrast, is quite used to the role. You might think of IBM as someone who comes from a wealthy family that has, for generations, been the envy of others. IBM not only expects that peers will be targeting it, it actually enjoys being successful and reacting to its competition. On the downside, our privileged friend is accustomed to winning and therefore doesn't know how to play the game when competitors use unconventional methods of attack.

—J. Hurwitz

GROUP SCHEDULING

ClockWise from Phasell Offers Group Scheduling with Added Value

Technology Borrowed from Prelude

Remember the AFCAC 151 RFP that set the open systems industry on its ear? Everyone wanted to win the bid. A good portion of that bid was for an office system, and, in almost every bid, Uniplex was the office solution. Except, of course, for the winning bid from AT&T. That proposal offered Prelude, an integrated office system written specifically to match the RFP by a company called Ventucom.

AN UNSUCCESSFUL PRODUCT. As it turned out, very few licenses of Prelude were sold, and most of the U.S. Air Force ended up with Uniplex or Quadratron, considered acceptable alternates after all was said and done. The problem with Prelude was, as it is with most products developed to match an RFP rather than to solve business problems, that it was short-sighted. It did what AFCAC required, not what users really needed in everyday work.

AN INTERESTING FEATURE. One of the few things we found interesting in Prelude was its task delegation capabilities. And that feature is now found on ClockWise, a Unix-based scheduler and task manager from PhaseII Software (Woburn, Massachusetts). PhaseII was founded by several developers from Ventucom who were committed to providing group solutions for office problems. When they began to design ClockWise, they wanted to include more group-like capabilities than simple meeting scheduling. The task delegation features of Prelude were very appealing, so, rather than recreating the code, PhaseII bought the integrated office system from Ventucom and used that chunk of code in the new product. Prelude is still officially on the market, available from PhaseII, but, in reality, the product languishes on the shelf. Other than using it as support for the very small installed base, PhaseII does not consider it a strategic product.

The ClockWise Product

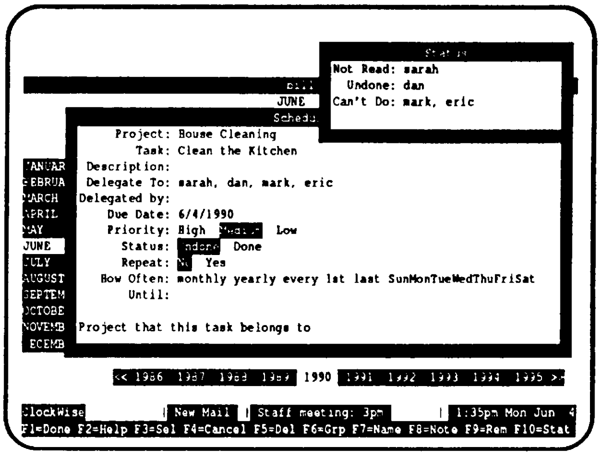
ClockWise is not particularly sexy. It doesn't feature a GU, nor does it offer a lot of advanced features. But it is good, hardworking, and useful software that could eventually evolve into a sexy, advanced, workflow-like management system.

GROUP SCHEDULING. PhaseII bills ClockWise as team management software, stressing the group features of the product. But the core of ClockWise is a group scheduling application. Individual calendars can be maintained for people and for resources. Scheduling a meeting uses the same on-screen form as scheduling a personal appointment. In fact, the same or similar forms are used for almost all the activities in ClockWise. This makes the product very easy to learn. Surprisingly, however, the scheduler will not find the "first available" slot for a meeting. Rather, it will display, in a reasonably graphic format (as graphic as you can be in a character interface), the calendars of all the meeting participants, indicating when each person (or resource) is unavailable. You have to explicitly select an appropriate time for your meeting.

We thought we would hate this method of scheduling meetings, but it was actually very easy to do. Our only concern is what happens when you want to schedule lots of people for a meeting; only 26 or so calendars can fit on a single screen. We do expect PhaseII to provide more automatic scheduling in the next major release.

TASK DELEGATION. In addition to group scheduling, ClockWise lets you delegate and track tasks—the functionality borrowed from Prelude. You can delegate a task to an individual or to a group. Basically, what you are doing is putting an item on someone else's (or your own) To Do list. The item will follow the person from day to day until it is *explicitly* marked "complete." The person delegating the task will also be able to monitor the status. If the task is not completed on the due date, it is labeled "overdue." Tasks can be prioritized, and repeated tasks can be scheduled. Tasks appear on the same screen as your daily events (meetings and appointments). A major limitation of the task feature, however, is that tasks only appear on your calendar/task list on the date they are due! You can't make Clockwise understand that the "reminder" date is different than the "completion" date and have them appear, say, two weeks earlier.

Checking Status of Tasks



```
                                              Status
                                          Not Read: sarah
                                    Bill    Undone: dan
                                    JUNE  Can't Do: mark, eric
                                   Schedu
          Project: House Cleaning
             Task: Clean the Kitchen
JANUAR  Description:
FEBRUA  Delegate To: sarah, dan, mark, eric
MARCH   Delegated by:
APRIL      Due Date: 6/4/1990
MAY        Priority: High  Medium  Low
JUNE         Status: Undone  Done
JULY         Repeat: No  Yes
AUGUST    How Often: monthly yearly every 1st last SunMonTueWedThuFriSat
SEPTEM        Until:
OCTOBE
NOVEMB  Project that this task belongs to
DECEMB

      << 1986  1987  1988  1989  1990  1991  1992  1993  1994  1995 >>

ClockWise          | New Mail | Staff meeting: 3pm      | 1:35pm Mon Jun  4
F1=Done F2=Help F3=Sel F4=Cancel F5=Del F6=Grp F7=Name F8=Note F9=Rem F10=Stat
```

Illustration 1. A delegator can check the status of tasks. Delegatees can refuse a task, as indicated by Mark's and Eric's status of "Can't Do."

LIMITED PROJECT CALENDARS. PhaseII states that you can set up project calendars for a quick view of progress of related tasks, but this is really a workaround. What you actually do is set up a resource calendar as a project and delegate tasks to it. We would like to see a more specific project calendar developed that includes relevant fields like project manager and project team members.

PERSONAL NAME AND ADDRESS LIST. A quickie personal rolodex feature is included with ClockWise. The feature was never intended to be more than a simple aid to the individual user, but PhaseII has found that this is the area where users are demanding more functionality. The company has already improved the performance of the name and address list by over 200 percent, and it plans to add more functionality in the next release.

NOTES. One particularly well-designed function is the ability to add a note to any event, task, person, etc. The method of attaching a note is the same no matter where you are in the system or to what the note is being attached. The text editor for notes, as for all ClockWise applications, is a very simple one, with limited editing capabilities. You can import or export into/from ClockWise in ASCII mode only.

FUTURE ENHANCEMENTS. GUI support is high on the enhancement list, as is improving the rolodex capabilities. The character world, however, will not be left behind. Mixed environments—character and GUI—will both be supported in future releases.

SYSTEM REQUIREMENTS AND PRICING. ClockWise runs on many Unix platforms, including SCO Unix, AT&T Unix/386, SunOS, Interactive Unix/386, AT&T 3B2, IBM RS/6000, DEC 3100/5000, Xenix, and others. The product supports terminals found in the Unix terminfo database. Licenses begin at $859 for 8 to 24 users. Site licenses are available.

Dedicated to Groupware

Bill Spencer, president of PhaseII, is a great proponent of group productivity software (groupware). ClockWise is PhaseII's first group product, and Spencer sees it as the foundation for a sophisticated workflow product. As that product develops, though, users can still take advantage of a pragmatic, usable piece of software that, though it won't set the world on fire, can help automate the difficult problems of group scheduling and task delegation. —*R. Marshak*

Topics covered in Patricia Seybold's Computer Industry Reports in 1990 & 1991:

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

Office Computing Report

1990—Volume 13

#	Date	Title
10	Oct.	Object-Oriented Data Management—Foundation for Adaptable Systems
11	Nov.	Filters—Separating the Wheat from the Chaff
12	Dec.	Workgroup Technologies—Managing the Relationships of Documents

1991—Volume 14

#	Date	Title
1	Jan.	NCR Cooperation—Strategy for Open Integration
2	Feb.	OfficeVision/2 LAN—In need of Revision
3	Mar.	Object-Oriented Development—A New Foundation for Software in the '90s
4	Apr.	Xerox DocuTeam—A Compelling Reason to Take a New Look at Xerox
5	May	The Battle for LAN-Based E-Mail—Lotus, Microsoft, and WordPerfect Go Head to Head
6	June	IBM OS/2 2.0—The Quest for the Desk
7	July	End-User Information Systems—An EIS for the Rest of Us
8	Aug.	The Windows Office—Evaluating Microsoft Windows as the De Facto Desktop Office Environment

UNIX in the Office

1990—Volume 5

#	Date	Title
10	Oct.	Sun's Open Windows–The Workgroup Macintosh
11	Nov.	Open Systems Defined—The Evolution from Proprietary is Complex
12	Dec.	Solbourne Computer—A New Model for Innovation

1991—Volume 6

#	Date	Title
1	Jan.	Open Multimedia Systems—Hurdles and Opportunites
2	Feb.	Desktop Managers—The Commercial User's Access to Unix
3	Mar.	Ingres—A Database Vendor in Transition
4	Apr.	Open CASE—Toward an Open Systems Infrastructure
5	May	Clarity's Rapport—The Designing of an Integrating Application
6	June	Uniface—Developing Database-Independent Applications
7	July	Can Digital Become an Open Software Company?
8	Aug.	Interbase Software—Extending the Relational Model to Handle Complex Data

Network Monitor

1990—Volume 5

#	Date	Title
10	Oct.	SystemView—Managing the World's Largest Information Systems
11	Nov.	Managing Storage—The Epoch-1 Infinite Storage Server
12	Dec.	Security Services—Standing between Chaos and Order in Distributed Environments

1991—Volume 6

#	Date	Title
1	Jan.	Portable NetWare—Extending the Red Reach
2	Feb.	The OSF DME—Open Distributed Systems Management
3	Mar.	Digital NAS—Services for the Distributed Computing Environment
4	Apr.	Sun Open Network Computing—Responding to the Challenge
5	May	PeerLogic PIPES Platform—Building Distributed Applications
6	June	Ellipse—LAN-Based OLTP Platform
7	July	IBM/Distributed Systems—Big Blue's Emerging Client/Server Architecture
8	Aug.	Name Services—Converging on a Two-Tier Model

Patricia Seybold's Computer Industry Reports

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
□ *Paradigm Shift—Patricia Seybold's Guide to the Information Revolution*	12 issues & tapes per year	$395	$407	$419
□ *Paradigm Shift—Patricia Seybold's Guide to the Information Revolution*	12 issues per year	$295	$307	$319

□ **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: ____________________

IU-0891

Send to: The Office Computing Group: 148 State Street, Boston MA 02109; Phone: 617-742-5200; FAX: 617-742-1028; MCI Mail: PSOCG

Special Research Reports from the Office Computing Group

Unix OLTP

Architectures, Vendor Strategies, and Issues — Price: $395

This report highlights:

- **OLTP Characteristics.** What is OLTP? Is "Classic OLTP" the only valid definition of OLTP? Where does "OLTP Light" fit in? And does Unix-based OLTP meet any of these definitions?
- **Comparative Architectures.** How are the designs of classic OLTP systems, fault-tolerant OLTP systems, and Unix-based OLTP systems similar? Where do they differ?
- **Standards.** The holy Grail of Unix and Unix-based OLTP is standards. What are these standards? Who creates them? Why are they important? Where are they leading the industry?
- **Vendors.** Who are the critical vendors in this industry? What are their design philosophies? What really exists for sale right now? What do users think?

A Kinder, Gentler Unix

Graphical Interface Strategies and Implementations — Price: $495

This special report features:

- **Usability Criteria.** A comprehensive list of GUI features with descriptions of their relative importance to the varying needs of different buyers.
- **Reviews.** Comprehensive reviews of the leading products from both hardware and software vendors.
- **The Unix Factor.** Considering Unix as the basis of open systems and its opportunity for taking the desktop.
- **Look and Feel.** A discussion of the issues of Look and Feel and other conventions—which are standards, and which are straitjackets?

Unix Relational Database Management

Vendor Strategies, DBMSs, and Applications Development Tools — Price: $595

This Special Report explores the product and marketing strategies of Informix, Ingres, Oracle, Sybase, Progress, and Unify. It takes the reader on a hands-on tour of the end-user and application development facilities of each product.

For more information: Call 1-800-826-2424 or 617-742-5200.

♻ Printed on recycled paper.