Patricia Seybold's

UNIX IN THE OFFICE

VOL. 3, NO. 2
ISSN: 0887-3054
FEBRUARY 1988

INSIDE

PRODUCTS • TRENDS • ISSUES • ANALYSIS

Progress

An Elegant Environment for Developers

By Judith R. Davis

PROGRESS (from Progress Software Corporation) is marching to a somewhat different drummer than are competitors such as Unify, Informix, Oracle, and Ingres. Progress is primarily marketed as a fourth-generation applications development language that just happens to have a good relational database management system (RDBMS) as its underlying foundation. Progress is not SQL based, and it hasn't yet announced a "star" architecture for supporting heterogeneous networks (although both developments are well underway). In these respects, it is most like Unify. Yet it differs from Unify and*(continued on page 3)*

• E D I T O R I A L •

Life after Unix

Visions of Computing in the '90s

By Patricia B. Seybold

IT IS PREMATURE to sound the death knell for Unix, particularly since Unix is just coming into its stride in the commercial marketplace. Over the next few months, we'll witness a swelling in the ranks as a number of major system suppliers unfurl their Unix strategies. In the late '80s and early '90s, proprietary operating systems will diminish in their strategic significance and in their ability to attract new software. We see IBM, Digital, and Apple holding on to their respective operating systems as strategic weapons (while hedging their bets with Unix). But we see Prime, Unisys, Hewlett-Packard, Wang, Data General, and others shifting their resources to the Unix camp.

What is the underlying significance behind the massing of the forces towards Unix? It demonstrates the power of software developers—the most valuable resources the industry has today. Developers are no longer interested in writing for proprietary platforms because doing so limits their options. Vendors whose systems do not have mass market appeal can no longer attract exciting new applications. They have no choice but to throw in the towel and join the march towards Unix.

The Posix standards (discussed in Vol. 3, No. 1) offer a bit of solace to vendors and users who need to straddle operating systems. By moving towards a standard application interface, the software industry may eventually become operating system-independent.

But the current Unix procession indicates another movement: a shift away from three-tiered, hierarchical computing architectures towards a two-tiered, distributed network computing environment. The two tiers—workstation and servers—will subsume today's current minis, mainframes, PCs, and workstations. We are moving inexorably towards a network computing model that more closely resembles today's Apollo and Sun systems than IBM's traditional, mainframe-centered world.

So, while the Unix tactic offers temporary survival for a number of existing computer suppliers, the more significant transition they will need to weather is the move to truly distributed, peer-to-peer networks with the concomitant distributed databases and processing technology required to achieve secure yet flexible systems.

FUTURE DIRECTIONS. We see a world in which personal workstations are tailored to each individual's working style and duties. The information the user gathers, assimilates, and analyzes can be provided automatically by pre-programmed agents that he or she can customize at any time. Information may arrive in the form of video clips, voice comments, graphs with live underlying data and formulas, text, data, or programs. The kinds of analytical tools available to the user will include three-dimensional representations, expert systems models, and dynamic modeling techniques. The information he or she will be able to obtain may be predetermined or dynamically accessed from other resources around the world.

Perhaps more important will be the kind of interactions each of us will be able to have with groups of colleagues, suppliers, and customers through this supportive networked environment. Many of us have worked for years in time-sharing environments. We have swapped mail messages and even joined in computer conferences on local and wide area networks. But the kind of collegial sharing of knowledge that will be possible in the new era of distributed computing cannot be directly extrapolated from those experiences.

Personally tailored knowledge bases will extend their tentacles across these distributed networks, hooking onto existing databases and applications, turning data into information, and enabling people to see and to establish new relationships among live, constantly shifting views of reality. These knowledge bases, representing different perspectives and expertise, will intersect and impact with each other, making new synergistic connections possible among people within and across organizations. ◐

Patricia Seybold's Office Computing Group

Editor-in-Chief PATRICIA B. SEYBOLD
Managing Editor RONNI T. MARSHAK
Senior Editor MICHAEL D. MILLIKIN
Senior Editor JUDITH S. HURWITZ
Assistant Editor DAVID S. MARSHAK
Staff Writers LAURE BROWN, FATIMA MAHDI
Sales RICHARD ALLSBROOK JR.
Associate Editor JUDITH R. DAVIS
33 Summit Road, Lexington, Massachusetts 02173
Telephone: (617) 861-3926
Contributing Editor GARY J. NUTT
3450 22nd Street, Boulder, Colorado 80302
Telephone: (303) 444-0341
Customer Service DEBORAH A. HAY

148 State Street, Suite 612, Boston, Massachusetts 02109 Telephone: (617) 742-5200 FAX: 1-617-742-1028

Patricia Seybold's Office Computing Report (ISSN 0887-3062) is published monthly for $385 (US), $397 (Canada), and $409 (Foreign) per year by Patricia Seybold's Office Computing Group, 148 State Street, Suite 612, Boston, MA 02109. Application to mail at second-class postage rates is pending at Boston, MA and additional mailing offices. POSTMASTER: Send address changes to The Office Computing Group, 148 State Street, Suite 612, Boston, MA 02109.

•PROGRESS•

(continued from page 1) from all of the other DBMSs we have reviewed in its simplicity of design.

Progress does not come with a variety of modules for the user/developer to select from, mixing and matching to meet application and user requirements. There is no separate forms generator or report writer. Progress is Progress. The applications development environment is the focal point for everything you do in Progress, not a separate arena you walk into after you have learned how to create databases, forms, reports, and queries.

We believe that Progress creates the most elegant development environment of all the DBMS products we have reviewed so far.

Progress is designed to tightly integrate the fourth-generation language (4GL) with the DBMS, and thus create a total environment for developing complete applications. Based on our own experience and on discussions with professional developers, we believe that Progress creates the most elegant development environment of all the DBMS products we have reviewed so far.

What is Progress like from an end-user perspective? What do the developers like about the product and what improvements would they like to see? What does Progress Software need to do to ensure its success in an increasingly competitive and rapidly changing market? As we did with Unify, Informix-SQL, Ingres, and Oracle, we set out to answer these questions about Progress.

Company Background

Progress Software Corporation was founded as Data Language Corporation in December 1981 by Joseph W. Alsop, president; Charles A. (Chip) Ziering, vice president of development; and Clyde Kessel, project manager. With a background in mainframe DBMSs, their objective was to bring mainframe DBMS functionality to the mini and micro environments. Product development began in the summer of 1982, and the Progress RDBMS and application development language was introduced in August 1984.

The company is privately held, with 40 percent owned by venture capital participants and 60 percent by its founders and employees. Revenues for the 1987 fiscal year (December 1 through November 30) came in at $6.5 million, well above 1986 revenues of $4.7 million. And the company projects 1988 revenues at $12 million. Since the introduction of Progress in 1984, the company has been profitable and claims to have a compound annual growth rate of over 40 percent.

The company recently changed its name to Progress Software Corporation to reflect the success and recognition of its product. At the same time, the company moved its headquarters down the road from Billerica to Bedford, Massachusetts and into a much larger office. Progress Software employs about 85 people, 60 in the United States and 25 overseas.

Product Overview

Progress Software develops and markets the Progress fourth-generation language (4GL) and RDBMS software. The company describes Progress as having five major components:

- Relational database manager for storing the data. The Progress DBMS stores data in a variable length format. It includes automatic concurrency control and transaction integrity (crash recovery or roll-back capability), and roll-forward recovery.

- Data dictionary that stores all information about the database structure, including the names of tables, descriptions of fields, index definitions, and access restrictions. You can use the data dictionary to change the database structure (adding or deleting fields, indexes, or tables) without having to unload and reload the actual data.

- Applications development language (the Progress 4GL). The company states that its 4GL is "complete" in that the developer does not have to resort to any 3GL in an application. This was confirmed by the developers we interviewed.

- Procedure editor for building, testing, executing, and modifying Progress procedures.

- Screen and report formatter to determine how the data is displayed. Progress includes an extensive set of defaults for this and also allows the user to directly specify formats, including multiple overlapping windows.

PACKAGING. Progress Software has chosen not to modularize its product to the same extent as do competitors like Oracle Corporation and Relational Technology. The Progress DBMS and 4GL are packaged together as a single product that the company refers to as "full Progress." The only subsets of Progress available are the run-time versions aimed at the professional application developer. The Progress family consists of:

- Full Progress. All DBMS and applications development capabilities.

- Run-Time Progress. Users can only run an existing Progress application; they cannot develop new applications or modify database structure. This is the typical run-time version provided by all DBMS vendors.

- Query/Run-Time Progress. With this, users can run an application the same way they do with Run-Time Progress. However, in addition, users have the full capability of the Progress

language to write any procedure that extracts data from the database (queries and reports). They just cannot write new applications that modify the data or the database structure. This product provides a nice middle ground for value-added resellers (VARs) in particular, enabling them to give their customers not only a custom-developed application, but also ad hoc development flexibility at a lower cost (and risk) than they would get from providing a complete version of Progress. None of the DBMSs we have reviewed to date provide this type of product.

- Developer's Toolkit. A set of tools assists the developer in packaging and distributing a Progress application. One advantage of Progress over some of its competitors is that the Developer's Toolkit includes a utility for compiling Progress procedures. Thus, a developer who creates an application on an AT and wants to port it to a Pyramid doesn't have to buy full Progress for the Pyramid. The Developer's Toolkit (at half the cost) provides what is necessary.

In the VAX/VMS environment, it may be difficult for Progress to overcome the disadvantages of both its late start and smaller footprint as a DBMS vendor.

Test Drive. Another unusual offering from Progress Software is Progress Test Drive. This is a demo version of Progress, but it, in fact, includes all of the Progress capabilities and the full documentation. The one limitation is that the user can only use a given database four times. The user can, however, continue testing the product by creating a new database. Test Drive costs a nominal $95 and has been a very successful marketing tool, allowing the company to introduce prospective customers to the power of Progress at a very low price.

PLATFORMS. Progress is written in C and was developed in the 68000-based Unix environment. It was first introduced on three Unix systems—Wicat, Fortune, and Cadmus—and now runs on a wide variety of Unix platforms (see the accompanying list).

Progress arrived in the PC world early on, in the spring of 1985. This has been a significant advantage for developers, who can develop a Progress application in a single-user DOS environment on the AT and then port it up to any other Progress platform, including multiuser systems. Many VARs chose Progress because of its early availability on the PC AT.

Progress Software has also kept pace with its competitors in the LAN market. LAN Progress was introduced for Unix LANs (Berkeley 4.2 and derivatives, such as Ultrix) in January 1987, and for DOS LANs in April 1987.

Digital's VAX. The most recent port of Progress was announced last November for Digital's VAX under VMS. Entry into this phenomenally successful market was a must for Progress Software, and several of its larger competitors have a huge headstart here. Both Ingres and Oracle have established significant positions in the VAX/VMS environment, and it may be difficult for Progress to overcome the disadvantages of both its late start and smaller footprint as a DBMS vendor. Marketing as well as the timely introduction of enhanced capabilities, such as networking, will be very important factors in how successful Progress Software ultimately becomes.

CTOS/BTOS. Progress Software also announced at the end of last year that it is porting Progress to the Convergent Technologies CTOS and Unisys BTOS operating systems, to be available this spring. Chip Ziering, vice president of development for Progress Software, described CTOS/BTOS as "a market with a real lack of application development systems. While CTOS/BTOS is not as widespread as Unix, there are significant pockets of use, particularly in the federal government."

VERSION 4. Progress Software introduced Version 4 of Progress last November, enhancing the product with a number of functionality and performance improvements. New functionality includes:

- Roll-forward recovery. Progress has always had exceptional roll-back recovery facilities. Version 4 adds roll-forward recovery, where a log of committed transactions (the after-image file) is kept on a separate disk. This log is used to reconstruct the database from a backup copy if the database disk is irretrievably damaged or the database becomes corrupted. Roll-forward recovery requires the use of two disks.
- Screen overlays. Multiple windows can appear simultaneously on the screen, and now the windows can be overlapped.
- Temporary files. Progress has the ability to define a file (table) in RAM to hold information temporarily. This temporary table can be manipulated as if it were a stored database file.
- High-level calls. Progress has the ability to call non-Progress programs (3GL programs written in C, Fortran, etc.) from within Progress procedures. The 3GL code is compiled and linked with the Progress modules.
- Compiler listing. The compiler listing is helpful for debugging, providing information such as the scope of a transaction.
- Background color. Progress now provides the ability to control the background color of forms displayed on terminals.
- Stack labels for columns. You can define multi-line (stacked)

Progress runs on the following hardware platforms:

Altos (Xenix)
Arete (System V)
AT&T 7300, 3B series (System V)
CCI Power series (System V)
Convergent Technologies (CT/IX)
DEC (VMS and Ultrix)
Fortune 32:16, Formula (FOR:PRO)
HP 9000 series (HP-UX)
Honeywell (System V)
IBM PC and compatibles (DOS and Xenix)
IBM PC LANs
(Netbios-compatible, DOS and Xenix)
IBM PC RT (AIX)
ICL Clan (DNIX)
Icon (MP/UX)
Intel (Xenix)
Motorola 8000 (System V)
NCR Tower (System V)
NEC Astra (ASTR-1X)
Plexus (System V)
Pyramid (OSX 3.0)
Sequent Balance (Dynix)
Sun (SunOS 3.0)
Unisys (System V)

Progress Pricing (Effective December, 1987)

Machine Class (examples)	Full Progress Price	Machine Class (examples)	Full Progress Price
A (DOS)	$1,000	H (AT&T 3B15, NCR Tower 32/800)	$15,000
B (Xenix/286, AT&T 7300)	$1,800	I (CCI Power 6/32 S, DEC VAX 8200/8250)	$20,000
C (Xenix/386, Altos 686/886, Sun-2)	$3,000	J (DEC VAX 8300/8350)	$25,000
D (DEC MicroVAX 2000, Plexus P-20)	$4,500	K (DEC VAX 8500/8530, Pyramid 98x)	$30,000
E (AT&T 3B/500, NCR Tower 32/600)	$6,500	L (DEC VAX 8550 through 8700, Unisys 7000/40)	$45,000
F (DEC MicroVAX II, Honeywell X-40)	$9,000	M (DEC VAX 8800, Pyramid 9830)	$60,000
G (Arete 1200/1600 with 2 APs)	$12,000	N (Pyramid 9840)	$80,000

Two additional high-end classes (O—$100,000 and P—$125,000) have been established for future ports of Progress on larger machines.

The Progress Query/Run-Time System and Developer's Toolkit each cost approximately 50% of the full Progress price. (If the Toolkit is purchased with full Progress for the same machine, the Toolkit costs only 25% of the full Progress price.)

The Progress Run-Time System costs approximately 20-25% of the full Progress price.

There are quantity discounts available for full Progress, Query/Run-Time, Run-Time, and the Developer's Toolkit.

Test Drive costs $95.

Annual maintenance is 15% of the list price. Under a maintenance contract, there is also a per update charge for media, documentation, handling, and shipping.

labels for columns on screen forms and reports, e.g.,

Sales
Region

Several performance enhancements were also added.

- Reduced amount of synchronous input/output (I/O). In Version 3, Progress writes a committed transaction directly to both the database and the before-image file, bypassing the Unix buffer cache. (The before-image file is kept in case the system crashes and an incomplete transaction must be rolled back.) In Version 4, the synchronous (direct) write to the database was changed to use the Unix buffer cache, and additional information on committed transactions is stored in the before-image file to enable the database to be restored if the system crashes.

- A new option for production environments speeds up subroutine calling.

- Faster field scan. Because Progress stores data in variable length records, it must refer to field pointers to locate the appropriate field(s) in a record. Version 4 adds an index pointer at every 16th field in the record, enabling it to search through records with many fields much more quickly than in previous versions.

- Improved buffer scan. Progress now uses a hash table algorithm to better manage the blocks of records stored in the buffer pool.

- The memory management involved in executing procedures was improved to reduce the overhead of calling subprocedures.

INSTALLED BASE. The company states that over 10,000 copies of Progress are in use with over 50,000 users.

Marketing Strategy

Progress is designed primarily for the applications developer. Progress Software sells to VARs, independent software developers, MIS professionals, and analysts and consultants who develop business and commercial applications.

The company positions the product as "a premier application development system," according to Joseph Cordo, manager of Market Development and Services, "We emphasize that Progress is a 4GL with which you can develop complete applications (without resorting to 3GL code), not a database management system," Cordo stated. Progress is intended to appeal to anyone interested in developing commercial business applications. Cordo said, "Several large corporations have chosen Progress because of its applications development environment, for its ability to provide solutions to business problems," not because of a requirement for SQL or distributed database capability. These include NCR Corporation, Bell South, and Marriott Corporation. They are asking the question: "What is the most productive way to prototype an application, get it up and running, and then ensure performance?"

As Progress sells more to MIS departments, it must address the need for SQL, enhance its presence in the VAX/VMS environment, improve networking, and provide access to other databases.

Two other very important strengths the company stresses are an underlying DBMS that is high performance and transaction oriented, and the portability of Progress applications among different operating environments.

CHANNELS. Progress is distributed through several channels. In the United States, Progress is distributed by over 300 VARs, several original equipment manufacturers (OEMs) such as Fortune and Sequent, and cooperative marketing partners (including Altos, Arete, Convergent Technologies, Fortune, Motorola, NEC, Sanyo/Icon, and Sequent). Cooperative marketing partners are companies that recognize the opportunity to sell more hardware if they train their sales forces on Progress as a solution to business problems. Progress Software agrees to train the sales force; provide collateral materials, handbooks, videos, and demo software; and cooperate on trade show activity and seminars.

The company also has four direct sales offices: in Boston, San Diego, Washington, DC, and Atlanta. The Atlanta office is new, evidence that the company is beefing up its direct sales force.

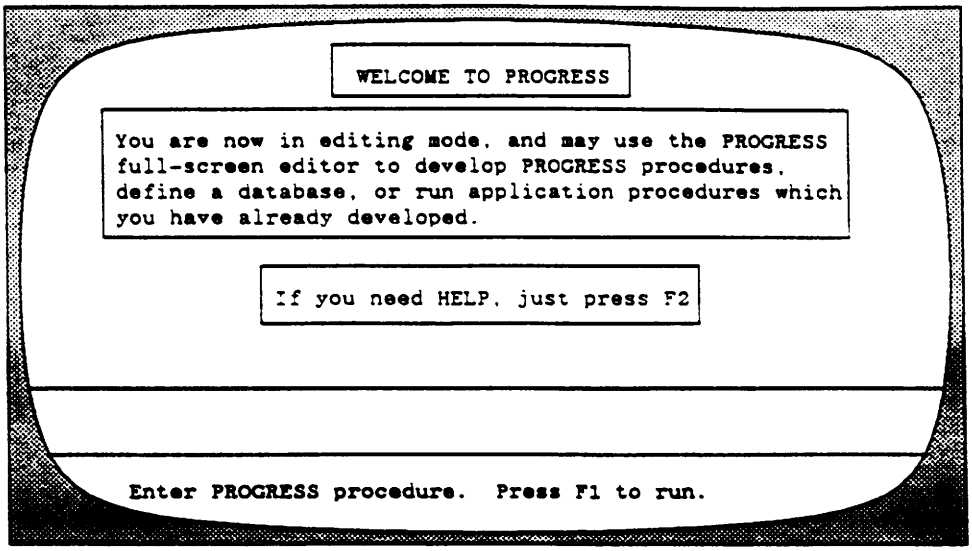
Internationally, the company uses a combination of wholly-owned subsidiaries, distributors, and over 200 VARs. Direct representation through subsidiaries is a relatively new approach. Last fall, Progress Software acquired one of its independent distributors and established a network of six European subsidiaries.

THE FUTURE. Progress Software believes that it has wide acceptance in the VAR marketplace and with a number of large corporations as well. As it becomes more involved in selling to the internal MIS department, the company realizes that it has to address the need for SQL, enhance its presence in Digital's VAX/VMS environment, improve networking, and provide access to non-Progress databases. Development in all of these areas is well underway (see "Future Directions" below).

But the company emphasizes its continuing commitment to the VAR market. Its roots are there, and, while it will strengthen its major account focus, it will still actively recruit VARs. VARs

have been very instrumental not only in distributing the product, but in introducing Progress within the corporate environment as well. In several cases, Progress Software has closed large deals with corporate customers that were generated by alert resellers. The company has a strong compensation plan in place for these resellers. In addition, the Progress sales managers work closely with VARs in following up with potential corporate customers.

According to Cordo, the company has more than doubled its sales and marketing staff in the past year, and the marketing budget is growing by 30 percent annually. "Sales and marketing is a key area of investment. We're at a critical stage right now, with revenue projections for this year of $12 million. We can continue our high growth or stagnate. To stay on our current growth path, we not only have to invest in the right product development, but also in marketing and sales."



This is the Welcome screen in Progress. You are now in the Progress editor. The horizontal lines enclose the editing area, which can expand to fill the entire screen.

THE COMPETITION. In the Unix market, Progress comes up against Informix and Unify most often, Ingres and Oracle less frequently. However, Ingres and Oracle are increasing their interest in Unix, and Progress does compete against them on larger systems.

In the PC world, the company mentioned Data Access's Dataflex and Ashton-Tate's dBase III, although Progress does not really compete head-to-head with dBase since the two companies' channels of distribution are different. Progress Software sells more to organizations looking to deploy applications developed on the PC into multiuser environments.

In the newer (for Progress) VAX/VMS environment, it's still too early to tell, but the company tends to bump into Cognos, Oracle, Software AG, Informix, and, of course, Digital's own products (Rdb and Datatrieve).

USING PROGRESS: THE END-USER'S PERSPECTIVE

We used Progress on both the NCR Tower under Unix (Version 3.2J) and the IBM PC AT under DOS (Version 4.2A), since Version 4 was not yet available for the Tower. We found Progress virtually identical in the two environments (except, of course, for the differences between the two versions). Unlike some of the previous DBMSs we have reviewed, Progress is Progress is Progress, regardless of the operating system.

SYSTEM REQUIREMENTS. The Unix version of Progress requires a minimum of 2MB of memory and 2.5MB disk storage. The DOS version requires 512K of memory, but 640K is recommended.

DATABASE PARAMETERS. You can have up to 1,023 tables and 1,023 indexes per database in Progress. The maximum record size is 2,000 bytes, and the number of fields (columns) per table is limited only by the maximum record size. Progress uses a variable-length format to store the information in each field, optimizing the use of storage space. The company also pointed out that using variable-length records typically enables Progress to store more data than DBMSs that use fixed-length records. A single database can span physical disk drives with the use of a multi-volume feature.

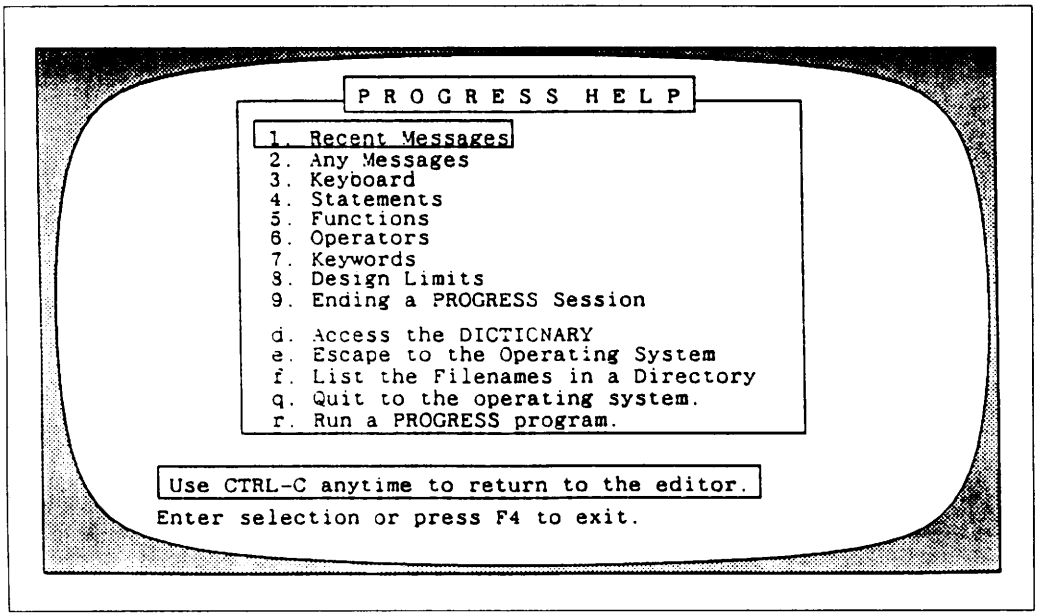
User Interface

The Progress approach has two major steps:

- Define the database using the data dictionary.
- Create procedures with the procedure editor to access and manipulate the database and to develop applications.

In the Progress environment, the procedure editor is "home base," so to speak. This is where you hang out in Progress, unless you are working in the data dictionary. You start in the procedure editor, and use it to access everything in Progress (including the data dictionary).

Progress has a menu system for both the data dictionary and help. Creating, modifying, and deleting tables and fields is done by selecting items from menus and completing screen forms. The menus are nicely implemented, allowing you to highlight



When you ask for help in Progress, you get this full-screen menu. You can select an item by highlighting it or by selecting the number or letter preceding it.

Statements within procedures can take up more than one line. The only requirement is that the entire statement end with a period (as if you were writing sentences). As do most procedural languages, Progress recommends that you indent nested statements and loops within procedures. This formatting convention makes procedures easier to read and understand. A nice touch in Progress is the fact that when you indent a line in a procedure, Progress continues to indent subsequent lines until you manually move to a different margin. This saves keystrokes in lengthy procedures.

the choice or enter the number or letter of the item. All other Progress functions and operations are done by entering statements in the procedure editor and running the procedure. This includes adding, updating, and deleting records, querying the database, and generating reports, as well as developing complex applications.

Using the Procedure Editor. The procedure editor is fully integrated with the data dictionary and the language compiler. It verifies the existence of tables and fields and checks your syntax. One of the best features of the procedure editor is its level of interaction with the user. If you make an error, it positions the cursor over the error and provides an error message. At this point, you can correct the error and then immediately rerun the procedure. You can move, delete, and copy text within a single procedure or between procedures. A single keystroke lets you test a Progress procedure (F1 for "run") or retrieve the last procedure you ran (F7).

You can also easily access operating system commands from within the procedure editor. Our one criticism here is that when we tried this initially, we couldn't remember how to return to the procedure editor and could not readily find the answer in the documentation.

You use function keys to save and retrieve procedures. The only problem we had with this was getting the same prompt whether we pressed F5 ("get") or F6 ("save"): "Type the name of the file: or blank for temporary file". In case we forget whether we pushed the right function key, we would like a hint in the prompt of which one we actually did press. That way, we won't inadvertently overwrite something on the disk or add something unnecessary to the contents of the procedure editor.

Progress is flexible in creating procedures. Procedures are stored as ASCII files and can be written using other editors.

HELP. Progress provides on-line help, but it is not context sensitive. It is, rather, a fixed menu of items from which you choose what you want to know. And there is no on-line help for the data dictionary function. Help appears to be designed primarily for assisting in the procedure debugging process. If you commit an error in a procedure, Progress will point to the line with the error and give you one or more error messages. If you still aren't sure what is wrong, you can press Help (F2) at this point. The first item on the Help menu (which is helpfully highlighted) is "recent messages". Selecting this will display each error message and give additional information on what you might have done wrong.

We did not always find Help as helpful as we would have liked. For example, if you select an item from the list of Progress statements, all you get is a one-line guide to the syntax of the command. No explanation. No examples. The functions are better—they include an explanation, but they also lack examples. End users love examples. They are sometimes much easier to relate to than generic syntax.

A minor point: Help is accessed with the F2 key consistently across all Progress platforms, but, in the PC environment, F1 has become the standard (de facto) position for Help.

We would also like to see a prompt on the screen at all times reminding us how to get Help. A prompt is displayed when you first enter Progress, but it disappears as soon as you clear the Welcome screen. Why not add a prompt (or soft-key display) to the line at the bottom of the screen? It would then read, "Enter Progress procedure. Press F1 to run. Press F2 for help."

One small inconsistency: When you exit from the Help function with the Escape key, the Help menu disappears. However, when you leave Help using the ^Break sequence, the Help menu remains on the screen, which is confusing since it is no longer active or accessible unless you press F2 again.

One more small point, if you will bear with us. The list of

keyboard functions on the PC version lists functions for F13 and F14, but these keys do not exist on the PC.

DOCUMENTATION. Progress documentation includes a tutorial, a reference manual, a programming handbook, a "pocket" Progress for quick reference, and two other separate documents on using multivolume databases and roll-forward recovery. You also get the Progress "Test Drive" manual, an abbreviated tutorial that gives you a quick but comprehensive first look at Progress. These booklets are all PC-sized and bound so they open flat, and they come in a handy carrying case (referred to as the "lunch box" at Progress). All that's missing is a frame or box for storing the manuals on the desk for easy access.

The manuals for Progress are clear and well written (including the Programming Handbook), even from the end-user perspective. They are full of examples and make good use of graphics. The writing style is straightforward and doesn't get hung up on terminology. In fact, the documentation does an excellent job of explaining terms and why Progress does something the way it does. One interesting point about the documentation is that it uses the term "file" rather than "table," an unusual approach for an RDBMS.

Because Progress looks the same on all systems, the documentation includes instructions for all environments: VAX/VMS, Unix, or DOS. That is, documentation comes in just one version, no matter what system you have. In addition, the documentation provides helpful hints about the operating system, such as the fact that Unix is case-sensitive whereas DOS is not, how long filenames can be, etc.

The Progress Test Drive is a good place to start working with Progress. It gives you the option to start at various places in the Test Drive manual depending on your primary interest: building your own application right now, seeing a sample application in action, starting from the beginning, etc. The next step for the end user is to go through the tutorial. Progress provides an extensive demo database for this.

Applications Development. One of the most unusual aspects of the documentation is its emphasis on application development. Before you even enter data into a newly created database, Progress has you developing a sample application to facilitate the process. This approach, because it makes application development a natural part of the process, tends to get you involved and comfortable before you realize what you are doing and have time to get nervous about writing programs. Progress makes creating procedures seem the only reasonable and easy thing to do.

Creating a Database

In Progress, the first step is creating a database. You do this by copying an existing database and modifying it, or by copying an "empty" database that Progress provides. At the Unix prompt, you merely enter the "prodb" command:

prodb [database name] empty

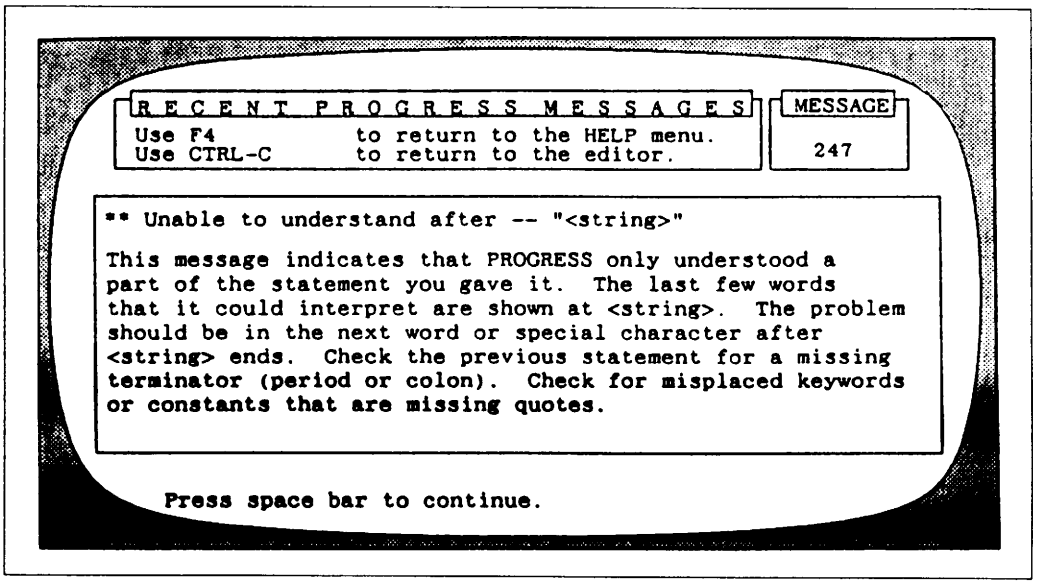
to create a new empty database. The company recommends that you create one or more separate working directories for your own databases. Once the database is created, you enter Progress with the "pro" command:

pro [database name]

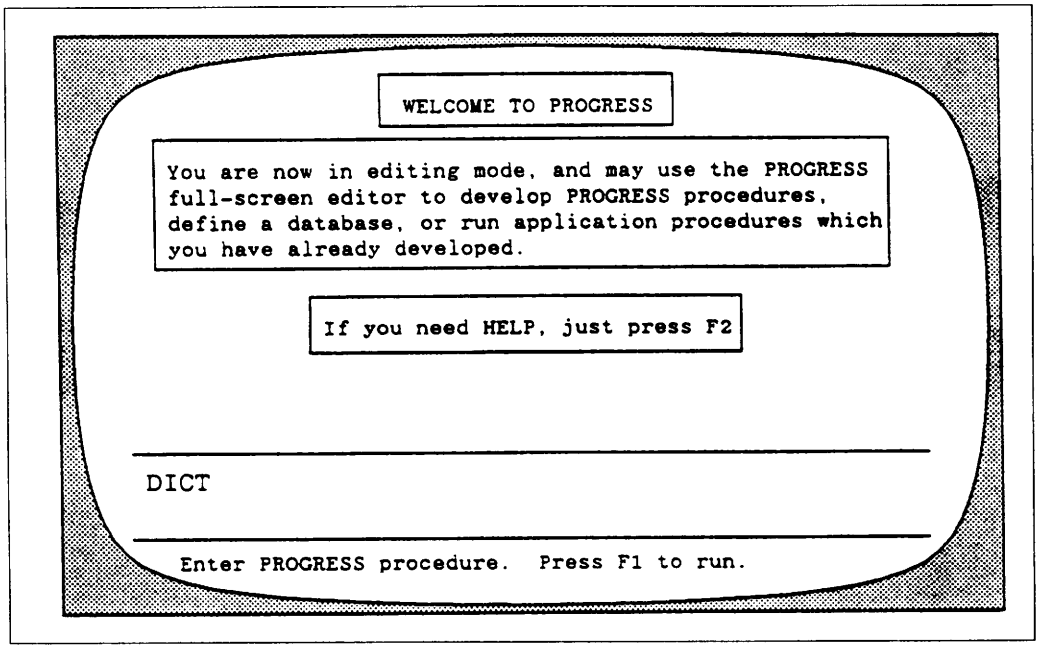
to start Progress using the new database. Therefore, Progress only operates on one database at a time. To use or create another database, you have to exit from Progress first. We would rather see the ability to change databases from within Progress.

A number of start-up options are available at this point. For example, you can specify a date and numeric format other than the standard defaults. These formats will apply throughout your Progress session.

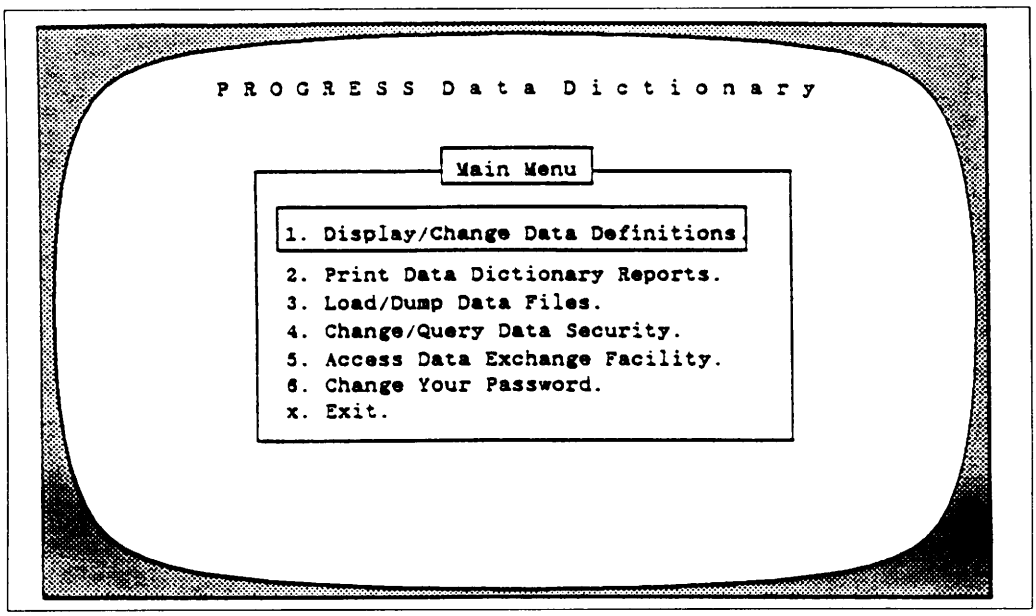
CREATING TABLES. You define new tables and alter existing tables in the data dictionary. (By the way, the Progress data dictionary is itself a Progress application.) First, you get a main menu where you select "Display/Change Data Definitions". Next, you get a screen listing the existing tables in the database.



A sample Help screen.



You enter the data dictionary from the procedure editor with the "dictionary" command (or from the Help menu).



Here is the main menu for the Progress data dictionary.

You can select one of these or create a new table name.

You can also enter optional information about the table: a description, a validation expression (a logical expression that must be true before Progress will delete a record from the file), and a message to be displayed if the validation test fails. An example of a validation test would be not allowing a user to delete a customer master record if orders for the customer are still outstanding. Complex validation criteria can be created in an "include" file and the "include" file referenced as the validation expression. Thus Progress, like Unify and some of the newer DBMSs, allows you to include referential integrity information in the data dictionary; it does not have to be built in at the application level.

We think this is a very important factor in increasing developer productivity. It also provides a level of integrity and security for the database; the database administrator is no longer dependent on the competence and thoroughness of the developer to preserve referential integrity.

DEFINING FIELDS. After you create a table, you get a menu of options and enter information about each field in the table. The only required information is the field name and data type. You do not have to indicate a field length because Progress stores the data in each field as a variable length.

Data Types. Progress supports five data types:

- Character. The size of a character field is limited only by the maximum record size of 2,000 characters (although the display limit for a character field on a screen form is 255).
- Integer. +/- 2,147,483,647.
- Decimal. Fifty digits are the maximum, with up to 10 decimal places. Progress includes an optional European format for commas and decimal points (the format can be changed as a start-up parameter when you first enter Progress and is in effect for the entire Progress session).
- Logical. True/false.
- Date. The date field has a default format of mm/dd/yy and handles dates from 1/1/32768 B.C. through 1/1/32767 A.D. Like the decimal field, alternative formats can be specified for dates. Progress handles a wide variety of date arithmetic and functions.

Arrays of any data type are also supported. An array field contains multiple elements. For example, you could define "ship-month" as an array field containing 12 elements, each element indicating the number of items shipped in a single month.

Null Values. Progress provides tools to handle null values (a special way of identifying "missing information" in a field) for those users who care about the effects of null values. In Progress, entering "?" in a field signifies an "unknown value," or null value. Therefore, a developer, when creating a database, can set the initial value in every field as "?", designating that this field contains missing information until specific data is entered to replace "?".

Otherwise, if a field is just left blank, it may be interpreted as having a default value other than an unknown value. (The only exception is a date field, where the default is an unknown value.)

Other Field Parameters. At this point, Progress fills in default values for the remaining field characteristics and allows you to change them if you want. They include:

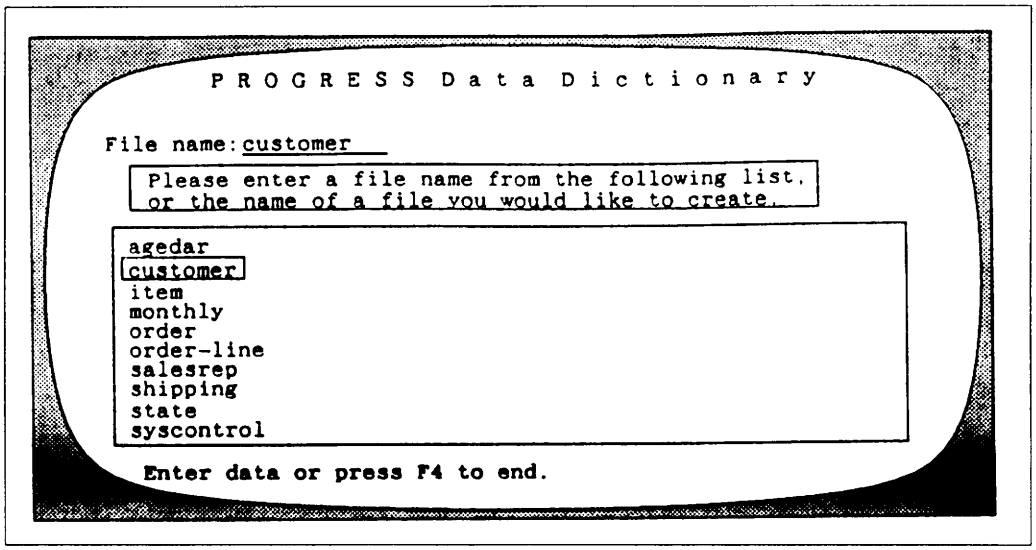
- Format. A description of how the field will be displayed on screens and reports. Progress provides a default—for example, x(8) for 8 characters is the default for character fields—and allows you a wide range of format options—all caps, a phone number displayed as (999) 999-9999, etc. Logical fields default to a yes/no format, but you can specify other designations for field contents as appropriate, such as true/false, paid/not paid, shipped/waiting.

- Label. A customized field label is provided for screens (the field name is the default).

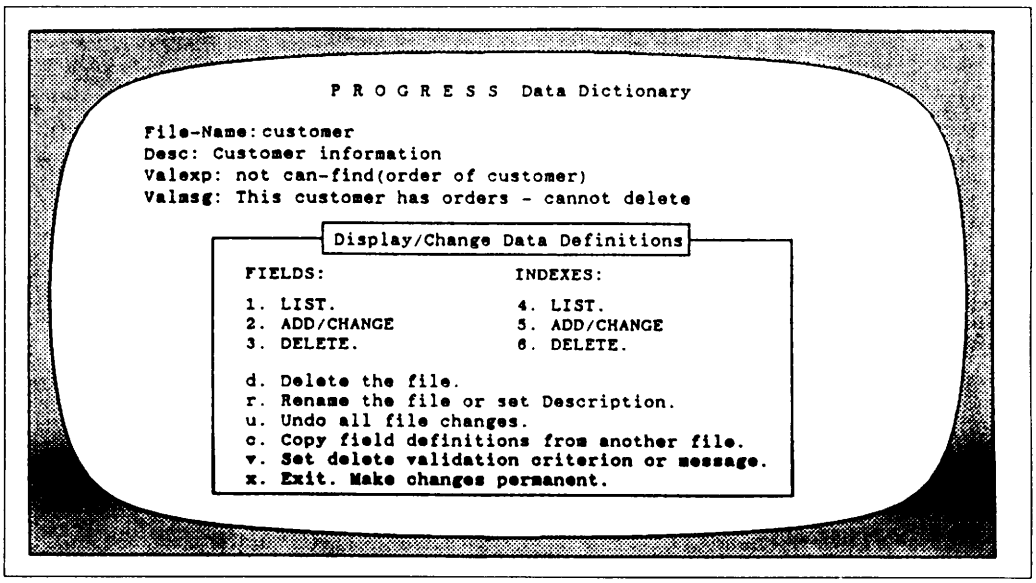
- Column label. A customized heading is provided for a column of data on a report or listing (again, the default is the field name). Labels can be overridden in procedures where appropriate.

- Initial value. A default initial value is provided for the field when a record is first created; options include functions such as "today", which will enter today's date into a field. The default initial values are a blank string for character fields, zero for decimal and integer fields, no (false) for logical fields, and "?" designating an "unknown value" for a date field.

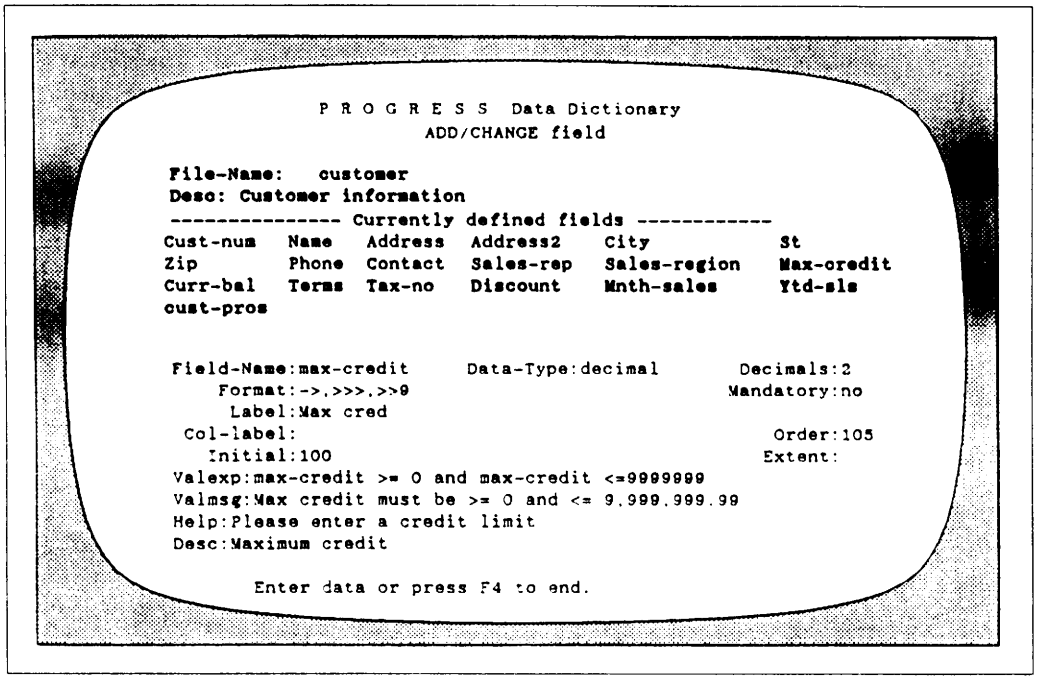
- Validation expression. The option is given to define a validation test for data entered in the field. We would have liked more information and examples in the documentation on creating validation expressions, particularly those where you



You choose the file (table) name from a list, or enter a new filename.



Here is the menu for defining fields and indexes for the selected table. There are also options for changing the table definition.



On this screen, you add or change field definitions.

validate the existence of data in another table (e.g., the customer number entered on an order form must exist in the customer file).

- Validation message. The validation message is displayed if the data entered does not meet the validation test.

- Help. A customized Help message is displayed at the bottom of the screen whenever data is to be entered in the field.

- Description. Text is provided to clearly describe the field and/or its purpose; this is for application documentation purposes only.

- Mandatory. A mandatory field cannot have an unknown value in it. We found the term "mandatory" confusing because you can avoid entering data even if the field is defined as mandatory. To get a true mandatory field that requires the user to input data, you define a field as mandatory with an initial value of "?".

- Order. The order in which the fields are displayed on the screen on default forms and reports is provided. The first field you define is assigned an order of "10," the second "20," and so on. If you wish to then create a third field but have it appear between the first two, you can either change all three numbers, or simply give the third field a number between 10 and 20.

- Extent. The extent default applies to array fields, indicating the number of elements in the array.

Progress has done a nice job of structuring the data dictionary menus and screens. We found the process relatively intuitive and easy to work with. We especially like the fact that Progress has made defining tables and fields comprehensive and yet not overwhelming.

Modifying the Database Structure

Modifying the database structure in Progress is much easier than it is in any of the other databases we have reviewed to date. To modify a field, you merely go into the data dictionary function and change whatever field is necessary. However, you cannot change a data type or the array extent of a field. To do either, you would have to create a new field, copy the data into the new field, and delete the old field.

You can also add and delete fields in the table. None of these modifications requires the user to unload the data first and load it back in after the database is modified.

Progress allows you to change the table specifications (such as validation test and message) as well as the table name. You can also delete tables (as long as the validation test isn't violated).

Only one person can change the data dictionary at one time, and, on a multiuser system, no one can access the database while changes are being made. Additional data dictionary functions include printing data dictionary reports, creating indexes, loading and dumping data files, specifying data security, accessing the data exchange facility, and changing your password.

Designing Screen Forms

Progress uses "frames," or windows, to display data, and automatically provides a default frame format if one isn't specified. To do this, Progress follows a comprehensive and well-documented set of default options. It is also possible to define custom frames within a procedure using what Progress calls "frame phrases." In these, the user can specify overall frame characteristics as well as those that affect individual fields in the display. Screen forms can contain multiple frames (windows) and frames can be overlaid on others. Progress also supports color and other video attributes.

DEFINING FRAMES. Overall frame characteristics define the way the entire frame is displayed and include the abilities to:

- Position the frame on the screen (center it, place it in a specific column and/or row, overlay it on another frame)

- Define the number of columns in which fields are displayed in the frame
- Define the number of records to display in a frame (single or multiple records)
- Specify video attributes
- Suppress the box around the frame
- Define the way labels are displayed (none, side labels versus columnar layout)
- Specify a frame that scrolls a specified number of lines at a time rather than a page at a time
- Put a title on the frame

For individual fields, many format phrase options either override or augment what has been defined in the data dictionary. You can change the format, the label, the help message, and the validation criteria for a field, among other things. You can also define variables that are not part of the database.

Screen formats are defined within procedures using the frame phrase. Another option is to define a frame using a "form" statement. This is helpful when the frame definition is quite complicated or if you will be using the frame repeatedly in one or more procedures. The form statement defines the frame and names it. The procedure can then invoke the frame definition whenever appropriate. This is more akin to the standard forms-design process in other DBMSs, where you design a specific form, store it, and call it up when needed.

Progress also provides space at the bottom of the screen for displaying application-specific messages.

FRAME BEHAVIOR. Frames can be viewed at any time, hidden, and overlaid. Again, Progress has defaults for these functions that the developer can override.

Data Entry and Editing

To add, update, and delete records in a database, you use the appropriate Progress commands (insert, update, and delete) in the procedure editor. And the Progress documentation tells you how to create simple procedures to perform each of these functions for multiple records. This is what we referred to earlier as the emphasis on applications development. You are creating procedures before you know it, and the documentation is clear enough to make it seem relatively easy. For example, to add a single record to a customer file, you might enter:

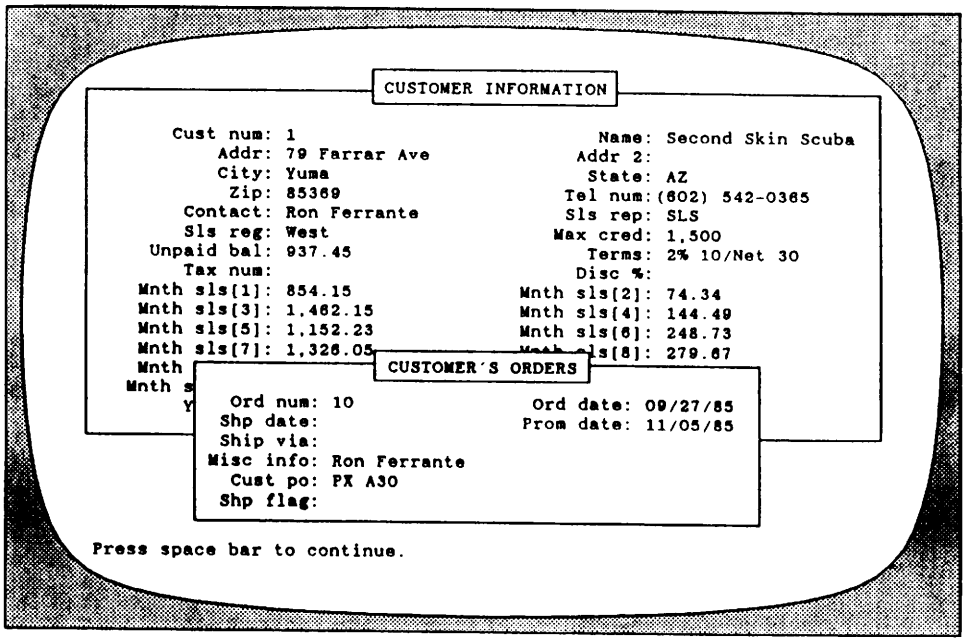
```
insert customer with 2 columns.
```

The procedure to repeat this for many new records is simply:

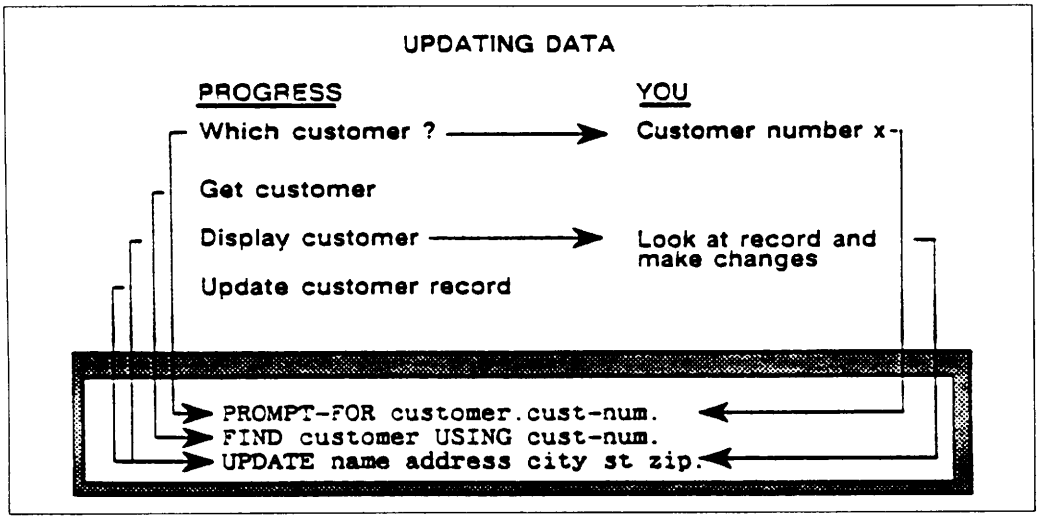
```
repeat:
  insert customer with 2 columns.
end.
```

You store this procedure and recall it any time you want to add records. (You do have to remember what you named the proce-

```
FOR EACH customer:
  DISPLAY customer WITH 2 COLUMNS TITLE "CUSTOMER INFORMATION".
  FOR EACH order OF customer:
    DISPLAY order-num odate sdate pdate shp-via misc-info
            cust-po shipped WITH 2 COLUMNS 1 DOWN OVERLAY
            TITLE "CUSTOMER'S ORDERS" ROW 14 COLUMN 10.
  END.
END.
```



A sample procedure to define a screen form in Progress. This procedure defines two frames, one of which overlays the other. The customer information is displayed first. When the user presses the space bar, the order window pops up, displaying each order for the customer, one at a time.



Here is a simple Progress procedure for updating a record. You could make this an iterative process to update more than one customer at a time by adding a "repeat:" loop around the procedure.

dure, which is not always easy.)

Updates and deletes are similar. While we found this straightforward to work with, it would be helpful to further "automate" this for the end user, giving us the option to specify a table name and get a strip menu for "add," "update," and "delete" with appropriate options to use a custom form, etc. This is a query-by-forms (QBF) type of function, which Progress does not have at the present time. However, the company plans to implement QBF with its Fast Track offering. Fast Track, which has not yet been announced, is a menu-driven, what-you-see-is-what-you-get (WYSIWYG) applications development environment (see "Future Directions" below).

Indexing

You use the data dictionary to create indexes for a table. You name the index, indicate if this index is the primary index used to access information in the table, indicate whether it is a unique index, and name the fields on which to build the index. Each field can be ascending or descending. You can also allow the use of abbreviations (i.e., the first few characters) in a character field to search for a partial match against an index. Other data types require an exact match.

You must have one primary index for each table. If you don't specify one, Progress will automatically provide a default primary index built on the internal record ID. You can also define secondary indexes up to a maximum of 1,023 indexes for the entire database. Progress uses B-tree indexing and allows indexing on multiple fields (up to 10 fields with a maximum index key of 126 characters). You can change the name of an index and delete it. You cannot change the components of an index once it is created; you would have to create a new index.

Once you define an index, Progress uses it automatically, selecting the most efficient path to the data.

Queries

Progress queries are generated from procedures that generally use the "for each" and "display" statements. For example,

```
for each customer where curr-
bal >= 1400:
   display name phone curr-
bal sales-rep.
end.
```

In this case, Progress will use its default formatting parameters to display the resulting data. The user could also add custom-formatting specs using the "frame" phrase. Progress is case-insensitive when typing in procedure statements or table/field names.

It is interesting to note that you will not find the word "query" anywhere in the Progress documentation. Queries in Progress terminology are essentially screen forms for displaying data in the database.

We found it amazingly easy in Progress to relate tables and display related data on the screen. Displaying information from three files (e.g., line item information for each order for each customer) in three separate windows took only a couple of minutes to generate and refine. And no forms design was necessary, although the display could easily have been spruced up. It is mind-boggling to imagine doing the same thing with SQL and some other forms-design products we have used.

What Progress lacks is a user-friendly QBF function. We also could not find a way to display aggregates (e.g., totals) without listing all the fields we wanted to see.

Progress has chosen to align column headings with the default data format: It left-justifies column headings over character fields and right-justifies headings over numeric fields. We would prefer consistency in handling headings independent of the data type in the column. Even better would be the option to align headings as appropriate—right, left, or center.

You cannot move backward through records retrieved by a Progress query without writing a non-trivial program to do this. Progress lacks a built-in function to go back to the previous record. The only way to see previous records is to rerun the query. From the end-user perspective, this is a serious flexibility limitation. The company stated that its Fast Track product will provide this "browsing" capability in its QBF function.

Generating Reports

A report in Progress is simply another query with control breaks,

aggregates, and fancier formatting. Report capabilities include:

- Defining control breaks for subgroups
- Specifying aggregates (total, average, count, maximum, minimum, sub-total, sub-average, etc.)
- Sorting the records on multiple levels
- Calculations across fields/variables/constants
- Specifying the output destination (screen, printer, ASCII file) and printer control codes
- Defining running headers and footers

As the report gets more complex and customized, so does the procedure required. If you want to override the Progress default layouts, there are a number of ways to do this, but you must use the procedural language (for example, to specify data formatting and layout). This starts to become intimidating to the end user. Progress is very powerful, but we would like to see some friendlier end-user tools in the area of report writing.

What we like about Progress is that report writing is essentially an extension of writing queries, not a separate module. And both queries and reports are generated the same way—by writing procedures in the procedure editor.

One aspect of writing reports we don't like: If a field-display format is not large enough to fit a column total, Progress displays "?????" where the total would be. To fix this, you must either change the field format in the data dictionary or add a format statement to the procedure. If there is space on the screen, we would like to see Progress handle this problem automatically, or at least prompt us that the problem exists and ask for a new format for the column.

Applications Development

The tutorial documentation on writing procedures is excellent, explaining in great detail how to write procedures and what is actually happening in the database as each statement in a procedure executes. The tutorial also explains how to compile and maintain procedures and what this means in terms of performance. Progress automatically compiles a procedure before running it for the first time, and you can run one procedure from another. The tutorial describes this process, explaining how to set up subprocedures and include files, and pass data (called shared variables) and arguments among procedures to build flexibility into an application. The documentation also covers putting the "finishing touches" on an application, such as providing context-sensitive Help and preparing the application for use.

In addition to the capabilities already described, other highlights of the Progress 4GL include:

- High-level file and record manipulation commands
- Automatic error-handling and recovery
- Automatic lock-handling for concurrent file access and the ability to override this
- Ability to call routines written in C (new with Version 4)
- Ability to provide context-sensitive, custom Help frames, in addition to individual Help messages for field input

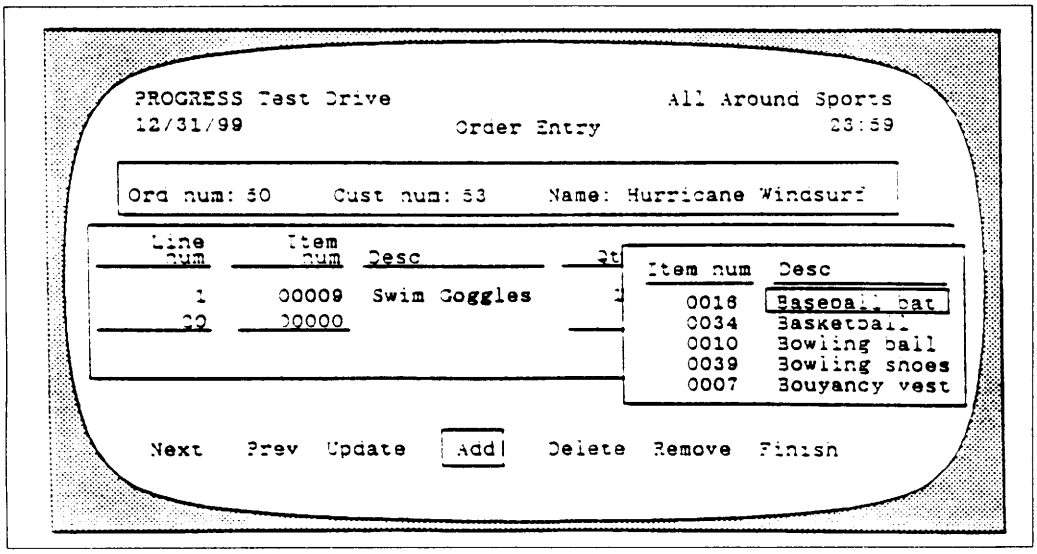
While Progress is not really an end-user development tool for the average user, there is an elegance and ease surrounding much of the Progress functionality that is impressive. Much of

```
FOR EACH salesrep BREAK BY slsrgn BY slsname:
  DISPLAY slsname slsrgn slsquota (TOTAL BY slsrgn)
          WITH TITLE "Sales Quotas by Region".
END.
```

Sales Quotas by Region

Name	Region	Yearly Quota	
Kintake, Ali I.	Central	$400,000	
		$400,000	TOTAL
Brawn, Bubba B.	East	$110,000	
Ewing, Rick A.	East	$125,000	
		$235,000	TOTAL
Pitt, Dirk J.	West	$400,000	
Retton, John Boy	West	$666,000	
		$1,066,000	TOTAL
		$17,011,000	TOTAL

Here is a simple procedure to generate a report. In this case, Progress is using its default specifications to lay out the report format.



A sample Progress application displaying multiple overlapping windows and a strip menu across the bottom of the screen. In this case, when you are in the item number field in the process of entering a new order, if you press Help, you get a frame listing the existing item numbers. You can scroll through, select one, and have the information transferred back to the line item frame.

this has to do with the close integration of the procedure editor and the data dictionary, the interactive nature of the procedure editor, and the fact that Progress essentially provides a single environment to work in. Unfortunately, some basic development steps, such as creating an application menu, require writing what looks like a relatively complex set of procedures. Yet, other functions that appear to be complex, such as relating three files and displaying information from each in multiple windows on a single screen, are incredibly easy. You don't even need to design a custom form to get meaningful results in Progress. The default formats are often quite serviceable.

A more automated applications generator tool, incorporating a "what you see is what you get" approach, would dramatically improve the accessibility of Progress's power and functionality for the end user. This is coming with the Fast Track product, and we encourage Progress Software to expand Fast Track for end-user applications development. It is a shame to limit the use of such a seamless development environment to professional developers and expert users.

COMPLETE ENVIRONMENT. Progress Software claims that Progress is a complete applications development environment, and that the developer will never need to code anything in a 3GL such as C, Cobol, or Fortran. Our interviews with experienced developers bear out that claim.

Data Integrity

One of the real strengths of Progress is its data integrity facilities. We have already mentioned the ability to include data validation and referential integrity in the data dictionary. In addition, Progress provides automatic crash recovery. If the system crashes or a user abnormally ends a Progress session in the middle of a transaction, Progress automatically rolls back any uncommitted transactions when the system is restarted. Progress returns the database to the condition it was in prior to the last (uncompleted) transaction for each user.

Progress, with Version 4, now also supports roll-forward recovery. Completed transactions are recorded in an after-image file, which is stored on a separate disk from the database. If the database disk or database becomes corrupted or damaged, the after-image file can be run against a backup copy of the database to restore the database to the point where it was damaged.

Database Security

In the Progress data dictionary, you can define user IDs and passwords to ensure that only authorized users can access the database. You can also specify separate read, write, create, and delete permissions at the table and field levels. Restricting access to specific records in a table can only be accomplished with a procedure.

Within an application, the developer can create an "activities permission" file, a table listing each procedure and the names of users authorized to run the procedure. Security can also be built in at the operating system level in Unix (and VMS) for databases and object code.

THE DEVELOPER'S PERSPECTIVE

We also discussed Progress with five developers who have used the product for at least one and one-half years, and some for up to three years. Two of the developers represent organizations that are using Progress for internal development, and three are VARs. All are quite happy with Progress. Three major strengths were cited across the board, and all had additional positives to support their satisfaction with the product. Progress has been used to develop a wide variety of applications, including Complex MRP (manufacturing resource planning) systems.

Strengths

PORTABILITY. All of the developers stressed the portability of Progress applications among platforms as a major benefit.

Applications can be developed on one platform and ported to another simply by recompiling the code—DOS to/from Unix to/from VMS to/from LANs and single user to multiuser. One developer stated that Progress's portability is "probably as good as you can get." Another stated that Progress applications can be ported "easily, with no known problems. We have not encountered any source code incompatibility issues at all." A third developer added that, in some cases where the chip is the same, you don't even need to recompile the code.

Another perspective here is the fact that, for some other DBMS products (e.g., Oracle), the user must be careful because one development tool may be available under one operating system but not under another. With Progress, these differences are minimal.

APPLICATION DEVELOPMENT. All of the developers also like the ease and speed with which applications can be developed and prototyped with Progress. This was a major factor in the selection of Progress by all of these organizations. Characteristics of Progress that facilitate the development process are the close integration of the DBMS with the 4GL, the interactive data dictionary, and the fact that the query language is the development language. Progress is also a complete 4GL, "as advertised." All of the developers stated that they have not had to resort to writing 3GL code to complete an application. One developer described Progress as a "total environment."

Integration. Several of the developers praised the level of integration between the DBMS and the development language. "It is all one product. This is simpler from both a purchasing point of view and from an applications development perspective. The coding and running of an application are all done in one environment." Another developer put it this way: "The Progress software is very consistent. When we looked at other products, like Informix and Unify, they seemed disjointed. That's one of the reasons Progress was so exciting." A third stated that the overriding consideration in selecting Progress was

Progress Database

Feature	Progress
Underlying file structure	Proprietary
Database parameters	
Tables/database	1,023
Records/database	No limit
Fields/record	No limit
Record size	2KB
User interface	Commands in procedure editor; full screen menus in data dictionary/Help
Menu bypass	No
Contextual help	No, unless custom help designed for an application
Tutorial	Yes
Data types	
Character	Yes (limited by record size)
Numeric	Yes (integer and decimal)
Currency	No (handled through format specification)
Date	Yes
Time	No (Progress has limited tools to support time values in a field)
Combination (links between 2 fields)	No
Binary	No
Variable length text	Yes
Support for Arrays	Yes; can define any data type as an array
Field attributes	
Case conversion	Yes
Default value	Yes
Required field	Yes
Acceptable values	Yes
Verification (enter data twice)	No
Formatting of data	Yes
Calculated fields	No
Display only (no entry/update)	Yes
Prompt (for data entry)	Yes
Error message	Yes
Customized help	Yes
Screen forms	
Default form generator	Yes
Customized	Yes
Multiple tables/form	Yes (no linit)
Multiple screens/form	Yes
Embedded processing (if-then-else logic, display aggregates)	Yes

Progress Database

Feature	Progress
Query-By-Forms	Doesn't have QBF yet. Queries written with procedure editor. All of these features of QBF are included in the Progress 4GL.
Exact match	
Relational operators	
Ranges	
List of values	
Wildcards	
Maximum/minimum values	
Print query results	
Pass results to Report Writer	
SQL	Progress doesn't have SQL yet; the following items are answered based on the capabilities of the Progress 4GL
Standard SQL statements:	
Data definition language (DDL)	Yes
Data manipulation language (DML)	Yes
Query language	Yes
Extensions to SQL:	
Commit/roll-back transactions	Yes
Execute operating system commands	Yes
Load/unload data to/from ASCII file	Yes
Additional data definition statements	Yes
(alter, drop table, etc.)	Yes
Can be embedded in C/Cobol programs	No
Can create a new table with query results	Yes
Stored queries	Yes
Case-insensitive (e.g., field names)	Yes
Optimizer	Yes
Can call C routines	Yes
How to create SQL queries	Proprietary query language (4GL); queries created with procedure editor
Report Writer	
Nonprocedural	Yes
Default report generator	Yes
Interactive report generator using screen forms	No
Interactive debugging	Progress 4GL, ASCII file, program
Input source	Yes
Multiple tables	Yes
Page formatting	Yes
Headers and footers	Yes
Data formatting	Yes
Sort data	Yes (10 levels)
Aggregate functions	Yes
Logical processing (if-then-else logic)	Yes
User variables	Yes
Prompt for input variables at run-time	Yes

the "seamless integration between the database and the 4GL. We can very rapidly develop a complex application."

Storing integrity rules in the data dictionary is another major benefit of Progress.

Training. One developer described Progress as an "easy move for internal programmers. People today who have been writing 3GL code and are comfortable with a procedural language will migrate easily and quickly to Progress. Progress is just an extension of the 3GL."

Productivity. One organization stated that eight developers were able to write the equivalent of 750,000 lines of Cobol code in Progress in one year. It estimated the productivity increase with Progress as approximately 8 or 10:1 over a 3GL, and about 2:1 over competitive products like Unify and Oracle (due primarily to the fact that Progress is integrated as one product).

Another developer rewrote a Basic application with Progress, replacing 250,000 lines of code with 30,000 lines with no need to use C. This person's estimate was a fivefold increase in productivity with Progress.

Functionality. One developer, an experienced mainframe database administrator, described Progress as having "most of the functionality of a large mainframe database." Progress has evolved substantially over the past three years, implementing tighter and tighter integration between the application and the data dictionary. The dictionary has become much more active, thus saving a lot of coding effort on the part of the developer. This is particularly true in a procedural language, where you may use a lot of redundant coding. "Progress is so powerful, I see no limitations in its capability. There is nothing that Progress can't do that a mainframe DBMS can do. And remember that here we are comparing a package costing hundreds of thousands of dollars on the mainframe to a $1,800 program on an AT under Xenix."

Another point is that, with Progress, it is easy to exit to the operating system and return. This increases your feeling of seamlessness and easy access to different functions when you work with Progress.

DATABASE INTEGRITY. All of the developers cited Progress's crash recovery capabilities. As one VAR stated, "The crash recovery facility is of major importance to our customers. If you turn the machine off in the middle of a transaction, when the system comes back up, it will automatically roll back the interrupted transaction and restore the database to its previous condition." No one mentioned any instances of integrity problems in Progress. One person (a VAR) said that his company's customers have "never lost any data. Progress's integrity is truly excellent."

COMPRESSED DATA STORAGE. Because Progress uses variable-length records, storing only the data that is actually entered in each field, it requires less disk space for storage than other DBMSs. Particularly for VARs, this can enable more features and functions to be built into an application without sacrificing efficiency and performance.

SUPPORT. Three of the developers consider the level of support provided by the company as a strength. "Support is very good in terms of problem resolution and maintenance."

DOCUMENTATION. The quality and completeness of the documentation was mentioned by two developers. One described the documentation as "excellent and exhaustive, including current knowledge of known bugs."

CREDIBILITY AND QUALITY. One developer praised Progress Software's "truth in advertising" approach, saying "most of the information in their literature really is true." The developer described this strength as one of the key factors that sets Progress apart from its competitors. Another developer perceives Progress Software as a very quality-oriented company, ensuring that its products are high quality. "New releases work well and do not contain many errors. The company does a good job of eliminating most bugs up front before releasing the product." A third developer said that Progress has a "low bug rate."

Progress Database

Feature	Progress
B-tree indexing	
Max. no. indexes	1,023
Max. no. fields/index	10
Max. size of index key	126
Order options	Ascend/decend
Unique index	Yes
Database security	
Login password	Yes
Database-level access	Yes
Table-level access	Yes
Record-level access	No (must be built in at application level)
Field-level access	Yes
Access by time of day	No (must be built in at application level)
Access by location (workstation)	No (must be built in at application level)
Ability to customize standard menus	Yes
Ability to design application menus	Yes
Default menu generator	No
Custom help	Yes
Ability to create views of database	Yes (only in forms)
File access methods	
Hash indexes	No
Links (explicit table relationships)	No
B-trees	Yes
Buffered sequential	Yes
Transactions	
Logging	Yes
Commit/roll-back transaction	Yes
Roll forward	Yes (version 4 only)
Referential integrity	Part of data dictionary
Concurrency control—locking levels:	
Database	No (done through security)
Table	No
Record	Yes
Raw input/output	No
Database can span multiple physical devices (disks)	Yes
Networked version for IBM PC	Yes

PERFORMANCE. As with other DBMSs we have reviewed, the performance sword can cut two ways, being both a strength (it's good enough generally) and an area for improvement (it could always be better and faster). According to the developers, Progress generally performs very well. One developer stated that his organization has developed a complex decision-support system that is operating in an online transaction-processing (OLTP) environment (multiuser, lots of contention, users making rapid decisions, over 50 defined relations, and 100 external indexes), and that Progress handles this quite well.

Another developer stated that Progress is distinguished from products like Oracle and Ingres in the micro/mini Unix environment by its performance. Progress has been optimized to run well on these systems, unlike Oracle and Ingres, which originated on mainframe-size systems. "Progress performs much better than Oracle and Ingres on these smaller systems. Progress marries the best of the two worlds: It has optimized performance on the micro/minicomputer while incorporating the features necessary to do data processing applications, such as record-locking and transaction recovery. And the recovery capabilities set Progress apart from most of its PC-oriented competitors."

Progress has been optimized to run well on micro/mini Unix systems, unlike Oracle and Ingres, which originated on mainframe-size systems.

Weaknesses

None of the developers had any major negatives to say about Progress, and no single area of improvement was mentioned by all five.

PERFORMANCE. As we stated above, users are always looking for better performance. The one area cited for improvement by two developers was a need for better performance under multiuser conditions. One stated that, on one of his organization's applications, "There can be up to 60 active operators. However, on this application and platform, Progress begins to max out at about 40 active users. At this point, performance begins to degrade dramatically." He added that Progress Software needs to develop a multi-threaded, serverless architecture and to better manage contention on larger systems. "Progress is lagging against the competition here. The question is whether the company is keeping up with technology in the Unix environment with regard to performance."

The company is currently working on this issue and intends to introduce such an architecture in a future release.

INDEX RECOVERY. The biggest issue with one of the developers is the lack of an easy way to recover a damaged index, particularly if it is the primary index. Progress Software has rectified this in Version 4, adding a utility that rebuilds indexes without requiring that the database be unloaded and reloaded. In this developer's view, however, the need to upgrade to solve the problem can get expensive for an organization like his with many installations of Progress.

And the developer stated that while Progress Software has been generally responsive; it has not communicated strongly enough the need to upgrade to a new version in order to solve the problem. "The company will state that there is potential for index corruption, but doesn't go far enough in describing the fix to the customer. Even an upgrade doesn't automatically solve the problem, since you would still have to re-create the database under the new version."

END-USER TOOLS. One of the internal developers would like to see Progress Software introduce more end-user tools to make applications easier for the nonprofessional to generate. The VAR market has been the company's primary market until recently, and more end-user tools will facilitate the sales process to corporate users. One option suggested (strongly) by the developer is for the company to extend its Fast Track product even more for the end user. (We agree.)

PRECISION. One developer thinks that Progress's floating point performance could be improved. "There are not many scientific applications written in Progress (or any other 4GL, for that matter). Progress, with its 10 digits to the right of the decimal in calculations, carries too much precision around. If you are manipulating lots of decimal numbers, performance slows down."

COMPANY SIZE. Two perspectives on the size of the company were expressed. One is that the company is still relatively small, and that it is important that it develop, introduce, and market the right products at the right time to survive. One developer (a VAR) summed it up this way: "Our only hesitation is, now that we're tied to Progress, if the company falls back in the market for any reason, it will affect us." Another, also a VAR, is not overly concerned about the size of the company. "In fact, being a smaller company can be a plus rather than a minus in terms of knowing your customers and servicing them well."

SQL. According to one internal developer, Progress Software has to address the whole question of SQL. "It may not be all that important today, but the advantage will come when people start developing toolsets that will be able to front-end an SQL database. SQL has many disadvantages, but if it evolves as a standard database access language, it will be very valuable."

Another internal developer stated that SQL is only important for organizations that closely follow IBM.

Future Directions

SQL. Chip Ziering, vice president of development at Progress Software, indicates that SQL is coming. The company has a lot of SQL up and running, and is close to completing its implementation of Level 1 of ANSI-standard SQL. This includes queries (the "select" statement), the data definition language (DDL—i.e., the "create table" statement), and the data manipulation language (DML—the "insert," "update," and "delete" statements). The company is currently working on the ability to define views, update views, and grant access privileges.

Progress is structuring its 4GL to accept SQL statements as well as Progress commands. A developer will be able to use one or the other, or to mix and match both types of commands in a single procedure. In addition, it will be possible to include Progress-type clauses in SQL statements to dress up the format and display of query results. For example,

```
select * from customer with 2 columns.
```

The "with 2 columns" clause is from the Progress 4GL and will format the retrieved data as desired, avoiding a wrap-around problem if there are many fields in each record. The company is also extending SQL to include current Progress capabilities such as defining array fields.

Essentially, Progress Software is implementing SQL because it is a requirement "check-off" item, particularly in the corporate MIS environment. Ziering indicated that the company has not heard its VARs demanding SQL to the same extent as internal corporate users are. The company stated that implementing SQL has not been as difficult as anticipated. Since the Progress language was built to go far beyond SQL, the necessary internal database access structures were already developed. It was more a matter of mapping SQL statements to the appropriate internal structures.

One of the problems for the company is timing its introduction of SQL. The addition of SQL involves a major new release of Progress, and the company just introduced Version 4 in November. Normally, major releases are a year apart. The company also has to complete other enhancements that will be rolled out with SQL. Ziering emphasized the company's desire to get SQL out quickly; it will be ready for beta testing in the next few months.

What will Progress Software's next step be with regard to SQL? The company plans to continue parallel development of both Level 2 SQL functionality and embedded SQL (the ability to embed SQL statements in 3GL programs and directly access a Progress database), with more emphasis on the embedded SQL.

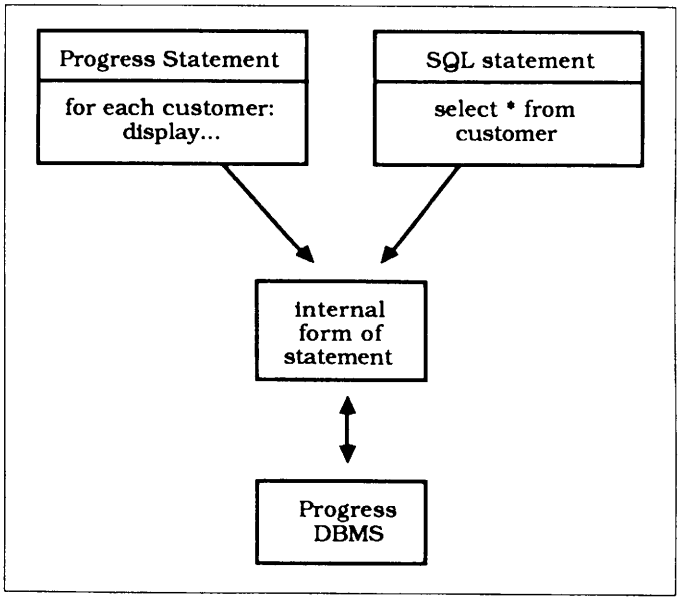
NETWORKING. Another strong area of development within Progress Software is splitting the Progress 4GL from the database management system. This is necessary so that the company can move in three important directions to implement networked databases, a major strategic goal.

First, Progress could accommodate applications requiring distributed databases (e.g., with the database itself spread among two or more systems).

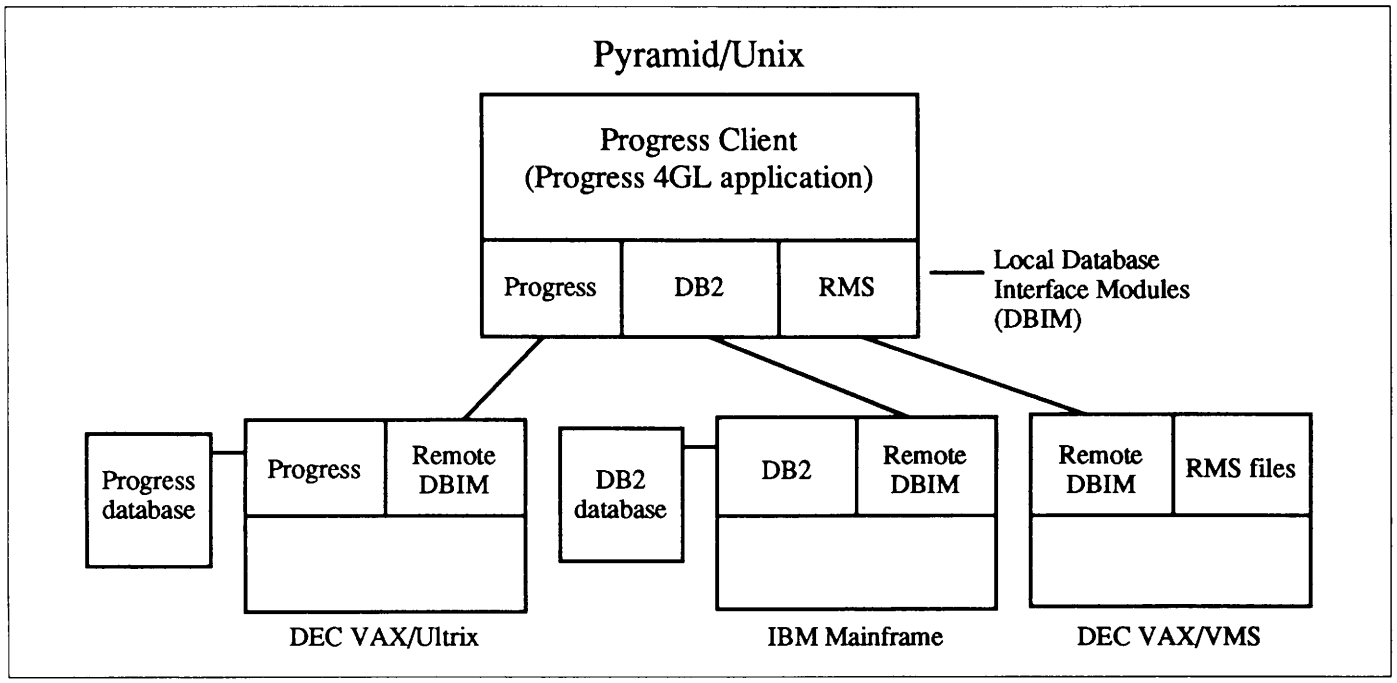
Second is the importance of running transparently across heterogeneous networks. Progress already runs on PC- and Unix-based (Berkeley 4.2) LANs with a separate front end and back end (i.e., the application/user interface runs on the workstation and the DBMS runs on the network server). However, this is restricted to LANs where the hardware and operating system are homogeneous (e.g., all PCs, all MicroVAXs, etc.). Progress must be able to run on a multivendor network. An example would be PCs on a LAN under DOS accessing data on a Pyramid under Unix. The company states that it now has Progress running successfully under these conditions using a TCP/IP-based network.

The third direction is access to heterogeneous DBMSs. By separating out the 4GL and building what Progress Software calls "database interface modules," the Progress 4GL could be used to access non-Progress databases such as IBM's DB2 and Digital's RMS and Rdb. Progress Software currently has a project underway to enable a customer to use Progress to access Total databases (Total is a DBMS from Cincom) and to work toward accessing Oracle databases as well. The key here is to set up the data dictionary so that it knows where everything is located and how to get it. Then the applications programmers and end users don't have to know where the data resides; they can access it transparently as if it were sitting on their local systems.

These developments would allow Progress to provide the application development environment for the user regardless of the DBMS where the data is stored. A database interface module, or gateway, is required for each "foreign" database. As Ziering put it, this "federated database" approach "allows each area within an organization to design its database in the most productive manner for its applications, and then use Progress to access



Progress is currently working on incorporating SQL statements into the Progress 4GL.



Progress Software intends to support both heterogeneous networking and access to heterogeneous DBMSs. This involves developing a Database Interface Module for each "foreign" database, a piece of which will run on both the client and server systems.

the multiple databases." Progress Software fully intends to implement two-way access where possible, not just the ability to retrieve data from foreign databases. For example, in addition to a QBF facility across different databases, Progress plans to include updates across systems.

Access to RMS files in the Digital environment and DB2 files on the IBM mainframe are high on the company's priority list. While Progress now runs on the Digital VAX under VMS, the company does not see its product being ported to the IBM mainframe in the near future. According to Ziering, "Progress's highly interactive user interface doesn't fit in an environment with block-mode terminals. The idea is not to run on the mainframe, but to access data there." The company is looking at the 9370 (VM and Unix under VM) as a potential platform for the future.

All three of these strategic directions are underway in parallel development efforts within the company.

APPLICATIONS DEVELOPMENT. Not content to rest on its laurels with regard to its application development tools, the company is beta-testing Progress Fast Track, a more visual and intuitive means of developing applications than the current 4GL. Fast Track is itself a Progress application, written in the Progress 4GL and layered on top of Progress.

Fast Track is a set of WYSIWYG application development tools that includes the following:

- Menu generator and editor for developing full-screen and menus (strip menu capability to be added in a later release)
- Screen painter
- Report writer
- Query-by-forms (QBF) facility

Fast Track is menu driven and automatically generates Progress 4GL code. The initial version of Fast Track is designed for the application developer, although Ziering acknowledged that the developer will not be able to complete 100 percent of all complex applications within Fast Track. This product also begins to push in the direction of end-user application development. The company wants to offer development tools, in particular the report writer, that can be passed on from the developer to the end user. Even with the first version of Fast Track, the company expects the report writer to be an end-user tool. And the addition of a QBF capability will provide more menu-driven assistance for the end user to add, update, and delete records.

Fast Track will be introduced as a layered product requiring Version 4 of Progress.

Editor. Other enhancements coming up for applications development are extensions to the procedure editor, such as global search and replace. The company's goal here is to develop a truly full-function, full-screen editor.

USER INTERFACE. Progress Software is pursuing three possible avenues for improving the ease of use of Progress for the end user. One idea is to extend Fast Track over time. Another is to develop a separate product, a pfs:File kind of tool, that is more limited in capability than the Progress 4GL but much easier for end-user development of simple applications.

A third direction is to work on developing a windowing,

graphics-oriented environment. The key here is to develop a model inside the Progress 4GL that can work on top of the appropriate windowing software (e.g., the Macintosh, the OS/2 Presentation Manager, Sun's SunView, X-Window, DECwindows), and allow the developer to put together an application that will run regardless of the platform. This is the approach that Relational Technology and Sun Microsystems are taking in their joint development project (see our news coverage in Vol. 3, No. 1).

TRANSACTION PROCESSING. The company has a project underway to develop a serverless, multi-threaded architecture for Progress. This is working in-house, and the company plans to release it in conjunction with its release of SQL.

OS/2. Progress Software is actively working on porting Progress to OS/2, and it plans to announce availability as soon as possible now that OS/2 has been released.

FOREIGN LANGUAGE SUPPORT. Progress Software has already taken several steps to deal gracefully with foreign languages. Progress users can change the default formats for dates and numbers without changing the application. This is part of starting up Progress and is independent of the application. Progress error messages are stored in a separate file and can be translated into other languages. The company believes that the end users of an application need to be addressed in their native languages.

The company describes current plans to extend its support for foreign alphabets by adding a full representation of foreign character sets, uppercase rules, and collating sequences. This will enable users in non-English-speaking countries to use their full character sets in entering data and to have indexing and sorting performed correctly.

Summary

Progress Software appears to have an excellent technical foundation and market reputation on which to build. And being relatively late to market can sometimes have a silver lining if the product has the architectural foundation and functionality on which to overlay more advanced development tools. The company is also working closely with NCR in a number of strategic development areas. Such a partnership with a major corporation that is also intimately involved in the industry should help Progress Software.

The ultimate questions for Progress Software are:

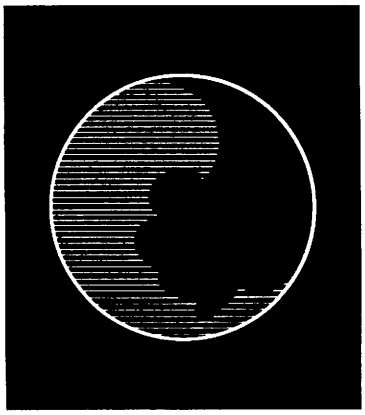
- Can it develop and introduce the right products at the right time and create the perception that it intends to bring state-of-the-art technology and tools to market on a timely basis?
- Can it successfully market the product, differentiating Progress's considerable benefits and capabilities from those of its larger competitors?
- Can it gracefully handle its next transition period as it approaches $12 million in revenues and a staff of 100+ employees? This is a question every growing company has to face.

In working with Progress, we perceive that it recognizes where it needs to improve both the product and the organization.

In working with the company, we perceive that it does recognize where it needs to improve both the product and the organization. This is certainly obvious in the developments that are already well underway. The competition will state that Progress has missed the boat by not conforming to SQL earlier. We are not convinced of this for three reasons: We are not fans of SQL as a language and would much rather use the Progress 4GL for applications development; SQL has not yet become a full standard; and we think Progress Software will not need as much time as its competitors think it will to implement SQL within the Progress environment.

What the company does need to focus on is improved end-user tools: a QBF facility, a visual (point-and-pick) query facility, a wonderfully easy report writer (can't someone manage to do this?), and an applications generator for menus, etc.

The company has an impressively full plate in front of it. The pressure is on to introduce heterogeneous networking, distributed databases, SQL, access to non-Progress databases, and OLTP capability within a year. If the people at Progress Software can do all this—plus make its environment friendlier for the end user, maintain the perception of a seamless and integrated development environment, and maintain the quality of its products—we will take our hats off to them. It's quite a challenge, and we hope Progress Software meets it. ◐



Patricia Seybold's ***Office Computing Group*** *Presents*

The Second Annual Technology Forum

"Distributed Network Computing: A Journey into the Future"

A Seminar about the Networked Distributed Computing Environment

April 27 – 29, 1988
Hyatt Regency Hotel,
Cambridge,
Massachusetts

The Second Annual Technology Forum takes an in-depth look at the emerging world of Distributed Network Computing. We are entering an era where the network is becoming the computer. We no longer have neat distinctions between personal computers, minicomputers, and mainframes. Technology is emerging that allows software to be distributed over a variety of processors—creating a virtual supercomputer.

Speakers will address topics including:

- ❑ What effect will NCS have on the distributed computing evolution?
- ❑ Will Digital commandeer the marketplace with DECnet System Services (DSS)?
- ❑ How are Unix vendors differentiating themselves?
- ❑ What will the role of OS/2 LAN Manager be in distributed computing?
- ❑ What will the typical Distributed Network look like?
- ❑ What will it take to manage this new breed of networks?
- ❑ What will the impact be on traditional data processing departments?
- ❑ What types of applications are emerging to take advantage of distributed computing?
- ❑ How will distributed computing affect the development of CSCW applications?

To reserve your space at the 1988 Technology Forum, simply fill out the order form below or call us at (617)742-5200.

☑ **PLEASE REGISTER ME**

Discounts available for two or more attendees from the same company. For information, call Gerri Cotton (617) 742-5200

☐ My check is enclosed for $795.
(Please make checks payable to Patricia Seybold's Office Computing Group)

☐ My purchase order number is _____

☐ Charge to my MasterCard/VISA (circle one)

Card # _____ Expiration Date _____

Signature _____

Name _____

Title _____

Company _____

Address _____

City, State, Zip _____

Area Code, Phone Number _____

Cancellation Policy:

Should a registrant be unable to attend the Forum, the Forum office will refund the full registration if notified before March 28. Cancellations from March 28 to April 10 are subject to a $50 service charge. There will be no refunds as of April 11. Substitutions may be made at any time.

Please return this form filled out to:
The Technology Forum
Patricia Seybold's Office Computing Group
148 State Street, Suite 612
Boston, Massachusetts 02109
(617) 742-5200

•TUTORIAL•

Network Protocols: Part 2

Implementing the mid-layer protocols of OSI.

By Gary J. Nutt

In our last tutorial (Vol. 2, No. 12), we talked about several different aspects of the International Standards Organization (ISO) Open Systems Interconnect (OSI) model for networks. The local area network (LAN) technology is typical of the trend toward open architectures. LAN interconnection systems are systems intended to support a wide variety of different machines manufactured by different vendors. Standards are mandatory for the ultimate acceptance of LAN technology by the users. Because of its technological lead, Transmission Control Protocol/Internet Protocol (TCP/IP) has become the de facto protocol for developing additional systems-level software, such as file server subsystems, remote window systems, etc.

Apollo, Digital, Wang, and Xerox, among others, were early to recognize the strategic importance of good LAN technology and moved forward with products that were ahead of the LAN standard efforts. IBM also recognized the need for good LANs, but appeared to have decided to stay with Systems Network Architecture (SNA)-compatible LANs as long as possible (e.g., 3270 versions of SNA), then to push for the standardized token-ring LAN. The pioneering efforts were rewarded in the classic manner; these LAN products have had to change when standards were established. The Xerox Ethernet products (published as an open product specification jointly with Digital and Intel in 1980) were the least affected by the resulting standards, but the products did have to be modified slightly in order to comply with the IEEE 802 Committee recommendation. The Apollo Domain network, WangNet, and DECnet have all changed considerably over the last few years, moving toward standardized lower layers.

Despite the plethora of excellent technical solutions and the number of viable options that are compatible with the ISO OSI standards, the number of solid product directions is small. At the physical layer, twisted-pair solutions dominate because of the tremendous problems in rewiring sites to accommodate higher bandwidth solutions. Whenever it is possible to install new media, coaxial cable dominates, either as broadband or baseband signaling media. Generally, the coaxial cable installations use the IEEE 802 broadband token-ring approach or the IEEE 802 baseband Ethernet approach. Various IBM products support both approaches, although their preferred approach appears to be the token ring. Meanwhile, the mainstream of the Unix community—including both System V and Berkeley Software Distribution (BSD)—has focused on the Ethernet technology.

The first generation of LAN hardware products implemented the physical and data-link layers of the ISO OSI architecture. These hardware products had to be accompanied by network and transport layer software in order for the hardware to be at all useful. The physical-layer hardware has two parts: the physical interconnection medium and the device for transmitting and receiving signals on the medium. In the case of twisted-pair media, the signaling device amounts to a modem; for broadband and baseband media, the device can be thought of as a cable tap. In order to get high data-transfer rates from the signaling media, the "transceivers" had to be very carefully manufactured. As a result, LAN taps were (and are) relatively expensive.

The remainder of the physical layer and all of the data-link layer were implemented in an integrated circuit. Intel, AMD, and others built sophisticated LAN (Ethernet) chips in the early 1980s. The original crop of LAN controllers contained little more function than that implemented by the LAN chip (i.e., the physical and data-link layer logic). In some cases, these controllers were built by new LAN companies for "standardized" buses (e.g., Digital Unibus, IBM PC and AT bus, Motorola VME bus, and Intel Multibus). Manufacturers that did not incorporate one of these buses in their equipment were forced to build their own controllers for their proprietary buses. This new market stimulated a few new companies such as Unger-

man-Bass, 3Com, and Bridge Technology.

The hardware products required that the "system integrator" provide the network- and transport-layer protocols as software in order for the LAN to be useful. The amount of software to implement the mid-layer protocols (network and transport layers) is not trivial, particularly if the network layer actually implements the IP protocol (see Vol. 2, No. 12). The mid-layer protocol software is large; should it be implemented as a driver? Perhaps partly as a driver, with the remainder as non-system library programs. How should it be divided? To complicate matters, most computer vendors did not have the expertise to quickly implement the mid-layer protocols. This caused a few small software vendors to create generic versions of the network and transport layers and become OEMs of the software for the system integrators—which, in most cases, were computer manufacturers attempting to connect their hardware into a standardized LAN. In this case, the manufacturer was required to port the generic mid-layer protocol software into his systems environment.

Unix vendors were almost the only vendors to accomplish this task. The generic mid-layer protocol code was generally written in C, the native tongue of Unix. Secondly, the Unix kernel supports the development of drivers for foreign devices without modifying the kernel (see Vol. 2, No. 8, pp. 16-17). When drivers got too big, functionality could be moved out of kernel space into user space as libraries. (As discussed in Vol. 2, No. 10, AT&T System V.3 provides a better solution, using streams to support the mid-layer protocols.)

Since the mid-layer protocols were implemented in software, and since the software was difficult to get into production, TCP/IP was in a perfect position to become firmly established as the protocol of choice. TCP/IP had been developed for use on the ARPAnet and was relatively mature. Furthermore, the BSD of Unix included TCP/IP in the standard release! It was easy to forget about the ISO OSI mid-layer protocols, at least in the short term.

Essentially, the important interfaces to a multivendor LAN had become the bus (in order to obtain an off-the-shelf controller board) and Unix driver code for the controller. TCP/IP was free with that environment. Although it was possible to implement these bottom four protocols in another operating system and bus environment, it was much more costly.

However, the PC user, more than almost any other user, needs the LAN. Without it, the PC remains isolated or, at best, interconnected using low-speed serial communication lines. The LAN protocols needed to be implemented in the PC world.

The standard PC processor is an Intel 8808; the low-end PS/2 processor is not substantially larger. These CPUs are marginal for supporting Unix-style operating systems, particularly in conjunction with mid-layer protocol software. Unix was also not attractive to the PC user who had a large investment in DOS application programs and the data that goes with them.

The obvious solution was to move the mid-layer protocol onto the controller board. This would satisfy the DOS marketplace, and it would also appeal to the Unix marketplace, since it would unload the CPU and eliminate the headaches associated with complex drivers, libraries, etc. In late 1986 and early 1987, the leading LAN vendors announced controller products that contained the mid-layer protocols. These boards include the same basic LAN chip that the earlier boards contained, but they also incorporate a CPU and memory to execute the mid-layer protocol. The mid-layer protocol software for the on-board CPU is stored in ROM. To use the controller, it is necessary to have a suitable bus interface and to be able to write a relatively simple device driver for the controller. The hardware interface is still the bus, but the software interface is now a TCP interface, an ISO OSI transport layer interface, or perhaps an SNA interface. It will probably be possible to provide all of these software interfaces with the same controller board but with different on-board ROM. The LAN vendors now own the mid-layer implementation problem. The question is, which protocols will be important?

The intelligent LAN controller boards are the wave of the future. The solution simplifies the systems integration problem and allows the vendor and application writer to concentrate on network applications such as electronic mail, file servers, distributed databases, etc. These solutions are less dependent upon Unix technology than previous ones, but Unix-based systems should already have established themselves as the environment for such application software. TCP/IP network software will continue to dominate for a while, since many of the network software applications depend upon it; eventually (when Fortran dies), TCP/IP will be displaced by full ISO OSI and SNA protocols. ◐

NEWS ANALYSIS

PRODUCTS • TRENDS • ISSUES • ANALYSIS

• INSIDE •

• MICROSOFT •

A Tale of Two Companies

PC or Mac? The word processor world has been divided while the question rages. Microsoft Word has been the top high-end word processor for the Macintosh environment; indeed, Mac Word, as it is sometimes called, validated the Mac as a word processing platform for many skeptics. Back on the PC ranch, WordPerfect has left all competitors in the dust. WordPerfect Version 4.2 is a corporate favorite, providing exceptional functionality and customer support. (The support issue is significant when Microsoft's difficulties with Word 3.0 for the PC are considered. The initial release was buggy, and, even though the problems have been fixed, the company is still taking heat for it.)

STIFF COMPETITION. Both Microsoft and WordPerfect are poised for growth. And they are eying the same (and each other's) terrain. WordPerfect for the Mac, announced at Fall Comdex 1987, is going to provide stiff competition on the computer "for the rest of us." Preliminary reports indicate that users are pleased as punch with beta versions (which WordPerfect is willingly handing out) and are itching to get the finished product.

Microsoft continues to add functionality to the PC Word product, and, rumor has it, a new PC-based word processing package is on the drawing board.

ENTERING THE FRAY. Where to go? The obvious next step for these competitors is to port to Unix, and they have done just that.

Word joins Multiplan as Microsoft's business application offering in Xenix (from, of course, Microsoft). The Xenix version brings all the advanced functionality of Word 3.0 for the PC and is completely file-compatible with the MS-DOS version of Word. And, since the DOS version can transfer files to and from Mac Word ... well, you get the picture.

Microsoft Word for Xenix will sell for $595, with additional manual sets at $50 each. The product is available today.

WordPerfect is following close on Microsoft's heels. The Unix version of WordPerfect has been announced and is in beta-testing now. We were impressed with our preliminary look. And, since WordPerfect is winning or projected to win on the other fronts, we anticipate that it will do well in the Unix market.

So what's a software developer to do? We like the capabilities of Microsoft Word. But, in the PC environment, its interface remains too enigmatic, and Mac Word falls short of the emerging leaders in the sophistication of its composition capabilities. The Xenix version makes no new promises, just file compatibility. Microsoft needs to give Word (all versions) a goose! Fix the minor deficiencies and put to rest, once and for all, its reputation for slow and buggy delivery.

WordPerfect seems to be on the glory road. Apparently, it can do no wrong. The challenge is not to get too cocky, but to continue to improve and update the products.

CONCLUSION. Competition is healthy, not only for the companies involved, but for the Unix/Xenix word processing market in general. Having the leaders in word processing software fight for your terrain only promises new and improved capabilities and top notch products. For too long, Unix has been a word processing wasteland. But Microsoft Word and WordPerfect, plus other notables including Innovative Software and Samna, are making the Unix environment as legitimate an environment for word processing as any other. ◐

—R. Marshak

•NCR•

New Towers, LAN Support Introduced

NCR (Dayton, Ohio) is trying to maintain its position as a leading traditional Unix hot-box vendor by making a major move to broaden the range and increasing the power of its popular Tower 32 products. At the same time, the company is keeping an eye on the growing PC LAN market by introducing software support for local area networking.

NEW TOWERS. NCR is beefing up the Tower 32 line at both ends as well as in the mid-range. The company is adding four new offerings (32/200, 32/450, 32/650, 32/850) to its current 32-bit Tower family, more than doubling its customers' options.

At the low end, the Tower 32/200 supports up to four terminals. It is aimed at the PC LAN market, claiming higher performance and greater cost-effectiveness than those of a competitive PC LAN. The 32/200 will list at $5,445 and will ship in April.

In the mid-range, NCR is upgrading its 32/400 and 32/600 with faster processors and better disk I/O performance. Processor speed has been increased from 16.7 MHz to 25 MHz, while disk I/O performance has been doubled by using a High Performance Mass Storage Controller. The 32/450 supports up to 32 terminals, and the 32/650 supports up to 64 (compared to 16 and 48 for the 32/400 and 32/600). List price for the 32/450 is $15,565, and, for the 32/650, $24,915. NCR will begin volume deliveries of both systems in June 1988.

At the high end, the 32/850 is built on the 32/800's distributed architecture. It supports up to six 25 MHz 68020-based Application Processors (APs), each with 8MB of on-board system memory expandable to 16MB. The 32/850 architecture also provides for a separate file processor for high-performance I/O processing; up to 64MB of system memory for each AP; 40KB cache memory and a 68882 floating-point coprocessor on each AP; and an integrated disk capacity of 1.9GB and an external disk capacity of 20GB. The 32/850 supports up to 512 terminals (compared to 128 for the 32/800). The list price for the 32/850 is $106,175. NCR will begin shipping it in the third quarter of 1988.

According to NCR, these new systems provide a hardware and software upgrade path for current NCR Tower users. All of the Tower 32s will become standardized on Unix 5.3 (SVID compliant) beginning with the 32/200, with the others migrating from Unix 5.2 to 5.3 during 1988.

LAN SUPPORT. While concentrating on expanding and enhancing its Tower line, NCR is not ignoring the burgeoning LAN movement. The company already has a line of 80286 and 80386 PC-compatibles. It sees these being networked in MS-DOS and OS/2 networks, with its 386 machines as servers in this environment.

NCR believes that it can provide better price/performance LAN alternatives using its towers as servers, and to this end, is introducing the NCR Tower File Server. The NCR Tower File Server provides a transparent file system between the workstations running DOS and Unix. The server supports MS-Net and DOS 3.1 file-locking. The next release will be Advanced Program-to-Program Communications (APPC) compatible.

NCR is offering industry-standard connectivity solutions. The LAN product family will include Token-Ring connections, with Token-Ring NETBIOS support available. TCP/IP Ethernet connectivity will also be forthcoming.

In addition to file service, the NCR File Server will provide print-spooling. Future plans include electronic mail that will link into Unix mail with X.400 gateways. The File Server will ship in the third quarter and is priced at $1,230.

Though providing a DOS solution with its PC-compatible line, NCR does not foresee providing OS/2 server applications capabilities on the Tower platform. It sees OS/2 as essentially an Intel-bound operating system, not readily transportable to Motorola-based systems. Besides, in NCR's view, Unix is a much more appropriate system for host-based applications.

ARE HOT BOXES HOT? NCR was able to leverage its hot boxes and standard Unix strategy into a successful turnaround for the company. Now that this approach has become the norm, NCR must continue to differentiate itself from the pack. The temptation is to concentrate on pushing bigger and better iron, letting others do the integration and add the value. NCR claims to be aware of this pitfall and points to its software and new LAN offerings. While these are certainly steps in the right direction, they are no more than steps and only indicate a direction. In order to succeed, NCR will have to follow these steps with more substance—more value-added in its standard software packages, better integration, and more services to its LAN offerings. Only then will NCR be able to remain as hot as its boxes. ◐ *—D. Marshak*

•OFFICE SYSTEMS•

Moving to VMS

Each year, Unix vendors seem to play musical chairs with their office systems. One year, one vendor is in the hot seat; the next year, it's somebody else. Well, this year it's Uniplex. And while Uniplex is busy with its office system success, a few other Unix vendors are investigating other markets. Last year, when most vendors hooked into the large installed base of DOS, Uniplex didn't. Instead, the company released only its word processing component for DOS. Now, we're finding VMS versions of Q-Office (from Quadratron) and Alis (from Applix), but still nothing from Uniplex. So far, the folks at

Uniplex have no plans for a DOS or VMS version. They say they're future oriented and they consider DOS a somewhat obsolete system. But let's face it: Unix is hardly a new environment, and it's going to be a while before DOS or VMS disappear.

Porting to VMS systems is a strategic next step for Quadratron and Applix, especially since Uniplex is the vendor to beat. Q-Office and Alis have already been running on the VAX under Ultrix, Digital's Unix derivative, but no more than 10 percent of the VAXs run on Ultrix or Unix. So the two companies are making profitable moves here. They're opening a whole new market, one that Uniplex hasn't tapped into.

QUADRATRON. Quadratron's position as *the* Unix office solution has toppled since Uniplex hit the scene. By tailoring its product for the DOS and VMS communities, the company has an opportunity to gain status in another market.

VMS versions of Q-One and Q-Office+ provide file compatibility while operating on Digital's VAX- and MicroVAX-based computers. The programs support VAX-specific environments, including installation, file-system management, and DECmail.

DOS versions give PC users the same capabilities offered on Unix, Xenix, and VMS systems. The DOS/VMS interface is identical to the Unix version, which is—in a way—too bad; we've seen better (see "Unix Office Roundup," Vol. 2, No. 12). Files, functions, and keystrokes are compatible across the board. DOS/VMS users can also link to Unix with what Quadratron calls the PC Connection. It allows PCs to use a Unix or Xenix machine as a master file server.

DOS and VMS versions of Q-Office sell on a per-CPU price schedule.

APPLIX. A VMS version of Applix's Unix office product, Alis, will be available in March. The integrated windowing package will include word processing with on-screen fonts, graphics, spreadsheet, data management, electronic mail, and scheduling. It provides file compatibility between different operating systems and hardware types.

The VMS version of Alis has the same components found in the Unix version: word processing, free-hand drawing, business graphics, spreadsheets, and database management. Alis's components are not great, but that might change by the time the VMS version is ready to ship. Applix is working on a major upgrade of Alis. Alis's best feature, its graphic interface, which includes live links between components to create compound documents, will be there. Like Q-Office, Alis also lets you link DOS and VMS to Unix. Its information-sharing feature lets you exchange data with different types of computers, as long as both computers share the same network.

The company is also looking into VMSmail and All-In-1 for the next version. Prices for the Alis VMS version start at $3,000 and depend on the system configuration.

CONCLUSION. Porting to VMS is a logical step for Unix vendors like Quadratron and Applix, who want to expand their markets, but there's another side to the coin. While these companies are taking advantage of DOS and VMS, such traditional DOS products as WordPerfect and Microsoft Word are becoming available for Unix systems. So, it seems that the three platforms worth hooking into are DOS, VMS, and Unix. It will be interesting to see what develops with OS/2.

What about Uniplex? Future-oriented or not, it seems to us that Uniplex already has its hands full with the success of its Unix office solution.

—L. Brown

•COMPUTER-BASED TRAINING•

Help under Unix

Software programs ought to be so intuitive that the user need not train at all. But, in this age of complicated databases and baroque data processing systems, this is not realistic. Computer-Based Training (CBT) is a convenient, interactive, effective, and comparatively inexpensive solution to the training issue. Many Unix vendors already offer CBT with their products, and most of the others will be forced to upgrade their systems to include it if they wish to bid on the AFCAC 251 request for proposal (RFP). All the preparation for AFCAC bidding is bound to make CBT a very visible issue in the coming months.

PLAYING IN THE BIG LEAGUE. For all its size, the U.S. Air Force is just another user with all of a user's hesitancies about installing a new computer system: Assuming the whole thing actually works, how do we get all our people trained and productive on the new system without a horrendous loss of time and money? From a close reading of the AFCAC 251 RFP, it is obvious that the Air Force considers CBT to be an important part of its user training. It is requiring CBT for all programs and systems as well as for systems administration tasks. In doing this, the Air Force is only asking for what any user would ask for, if he thought he had the money and the clout to get it.

Cheers and Jeers. The AFCAC bid will be a thinly disguised blessing for all users in the Unix environment. Vendors who are preparing their products for the AFCAC bid—and, because of the money involved, there will be a lot of them—will be forced to provide comprehensive CBT courses. These courses will, in turn, probably become available to other Unix users. To put it bluntly, the vendors will be providing the kind of training support for their products that they ought to have been providing all along. What can we say but "Thank you, Uncle Sam!"

AUTHOLOGY. One company hoping to profit from the new concern over CBT is Computer Enhanced Interactive Training (CEIT) of San Jose, California. CEIT is offering a DOS-based CBT-course authoring tool called Au-

thology.

Authoring programs are generally high-level language programs designed for users who may not be computer experts. For ease of use, the Authology interface relies on context-sensitive, online help, a windows-like environment, and menus and dialogue boxes for text entry and selection. The spreadsheet-like format of the application-building tool (see "Author" below) is an intuitive way to organize information, and it aids in visualizing and planning the course.

Authology requires that the trainer write his courses under a DOS operating system. But because Authology was written in C, a course written using Authology can be delivered on either a DOS or a Unix system. This allows third-party training firms that are not accustomed to writing for Unix systems to write Unix CBT courses in the DOS environment. Courses for students who are learning on a DOS machine can be enhanced with graphics, animation, video, and music; Unix courses are offered in text only. The ability to present courses in DOS and Unix will be a selling point for Authology with the AFCAC vendors, because the project includes support for DOS PCs as well as the Unix-based systems.

The cost for Authology is figured on a site license basis and depends on the number of machines being used for training. CEIT provides its own training in Authology, of course, and learning to use the text-only portion takes two days.

Authology comes in three parts:

- Author, the application-building tool for the trainer
- Presenter, the run-time program required on the trainee systems
- The Student Score Reporter (SSR), the evaluation program for the trainer

Author. "Author" is the program the trainer uses to write the course on his or her PC. It is based on a spreadsheet program, with each lesson as a separate spreadsheet. Along the left column are the names of the pieces of the presentation in the order they will be shown to the student, for example, "introduction, video sample, text sample, questions samples, etc." Along the top row are the names of the descriptions, qualifiers, conditions, etc. that fully describe each piece of the presentation. These include, among others, whether the piece is text, graphics, or video; where to find the piece; how much of the piece to show; and whether it's shown all the time or only when the student gives a certain response to a question. This information is selected through the use of menus, so the trainer need not remember all the options in any category. The trainer can use the spreadsheet as an overview tool to review a summary of the entire lesson with all of its branches in one glance.

Authology allows unlimited branching so that the trainer can change the path of the trainee through the lesson depending on the responses to the questions posed. If the trainee already seems to know all the answers, he or she can be sent ahead to more advanced sections of the lesson or to a completely different lesson. If he or she is having difficulty with the material, the trainee can be sent into a loop that reinforces one concept and then returns to the same place in the lesson, or to a slower, more detailed path through the lesson. Branching enables the trainer to tailor the course to many types of people and to make it as interactive as he or she wishes.

Among its other useful features, Authology contains a forms editor that will appeal to vendors of data processing systems. In a CBT lesson, the trainer can create a replica of a form that a data-entry operator would be expected to use. The trainee is prompted to fill in the form and the course gives feedback on each field of the mock form. If the trainee types in information incorrectly, the program can respond in a variety of ways to correct and reinforce the right procedure. For example, the course can show the trainee how the information should be correctly entered in the field, give a lesson on what that field means, and/or prompt the trainee to try again.

Presenter. "Presenter" is a separate standalone program required to run any of the courses written under Author. Besides simply running the course, Presenter also keeps score of the student's responses and evaluates these responses in terms of the overall goals of the course, called the lesson objectives, which the trainer determined when writing the lesson. The information on questions and objective performance can be saved on a disk or a network file server.

SSR. The Student Score Reporter (SSR) allows the trainer to call up and evaluate all the information on student performance gathered by the Presenter from all the students who have taken any one course. The performances can be compared by question, by objective, or by lesson, and can be represented graphically if the trainer desires. The trainer can see, for example, which lesson programs work better than others, which questions no one seems to understand, which objectives most people are failing, and which areas will need more reinforcement.

LOOKING TO THE FUTURE. Authology seems to be a full-featured program, but—graphics, please. Although many dumb terminals have connected to Unix systems, the percentage is dropping as graphic workstations come down in price. On top of that, it has been proven many times that colorful, graphic presentation of material aids retention and understanding. Taking into consideration both of these facts, it would be a pity if Authology continued to limit its Unix course presentation to text only, with all of the graphics and video features its already has in its DOS version. ◐ —*F. Mahdi*

Patricia Seybold's Computer Industry Reports

Topics covered in Patricia Seybold's Computer Industry Reports in 1987:

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

Office Computing Report

1987—Volume 10

1 Jan. Predictions—Systems and Players in the 1990s *Other Features:* Using Technology to Competitive Advantage

2 Feb. Leading the Way—Will DEC stay on Its Roll? *Second Feature:* General, DEC, HP, IBM, Wang

3 Mar. Digital's All-In-1 *Second Feature:* Service and Support II: Honeywell, NBI, NCR, Prime, Xerox

4 Apr. LANs in the Real World: The Users Speak *Second Feature:* 3Com's 3 Station: A VAXmate Killer?

5 May Wang Tries a New Approach: Integrated Image Management *Second Feature:* Electronic Publishing

~~6 June Prime Information~~ **OUT OF STOCK**

7 July End-User Computing—Tools for Visionary Users

8 Aug. Will Wang Make it? Can Fred's Leadership Turn the Tide

9 Sep. Voice Messaging Systems—User Perspectives

10 Oct. IBM Office Software—Implementing A Cohesive Vision *Second Feature:* The Marriage of PC and FAX

11 Nov. User Developed Applications—They Know What They Need

12 Dec. Hewlett-Packard's New Wave

UNIX in the Office

1987—Volume 2

1 Jan. DOS/Unix Integration—Doing it Right with Multiplex *Second Feature:* It's...Super Boxes!—Disguised as Mild-Mannered Multiuser Systems

2 Feb. The Officesmith—The Keyword Is Structure *Second Feature:* Standardize or Bow Out

3 Mar. Unix Arrives!—And UniForum Is It's Coming Out Party

4 Apr. Apollo's Challenge: Bowing to Standards, Yet Remaining on Top

5 May Open Systems and Standards Fuel Sun's Fire *Second Feature:* A New Dimension: Plexus's Mixed-Mode Data Processing

6 June The Ingres Realtional DBMS—Riding the Wave

7 July Texas Instruments—Blending AI with Commercial Systems

8 Aug. Banyan Systems—A Unix Solution to the PC LAN Conundrum

9 Sep. AFCAC 251—Defining the Future Unix Office

10 Oct. The Oracle Relational DBMS—The Leader of the Pack

11 Nov. Expres Initiative—Is there an Office Use for University Communications

12 Dec. Unix Roundup—Unix-Based Integrated Office Systems

Network Monitor

1987—Volume 2

1 Jan. Wang's WangNet—A Product In Transition *Second Feature:* Broadband Technology Primer—Wang's Dual-Cable Approach

2 Feb. Sytek—Can It Make the Big Time? *Second Feature:* Fibronix Debuts FDDI LAN

3 Mar. Bridge Communications—Stalking the Competition

4 Apr. Networking the Macintosh: Apple Opens the Floodgates *Second Feature:* Hewlett-Packard's GlobalNetwork Strategy

5 May Integrated Voice and Data LANS: IEEE to Marry ISDN and LANS *Second Feature:* Fox Research's 10-NET: An Expatriate Returns

6 June Banyan Systems Incorporated—Networks for People Who Think Big

7 July Apollo's New Thrust—Shaping Computing's New Generation

8 Aug. Proteon Inc.—A Better Mousetrap Isn't Enough

~~9 Sept. Digital's DecNet~~ **OUT OF STOCK**

10 Oct. DataGeneral—Better Late Than Never

11 Nov. IBM and SNA—Meeting the Challenge?

12 Dec. Micom Interlan Inc.—Learning From Past Mistakes

Patricia Seybold's Office Computing Group: 148 State Street, Boston MA 02109; (617) 742-5200; FAX: 1-617-742-1028