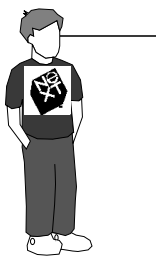


SCaNeWS

Southern California NeXT™ Users Group Newsletter

VOLUME 1, NUMBER 2
FEBRUARY 1991

SCaNeWS ©1991
All Rights Reserved

This issue has evolved into one of comparisons. Inside you'll find articles comparing the Mac and NeXT environments, IBM and NeXT implementations of NeXTstep 1.0, and Sun and NeXT benchmarks. In addition, we're delighted to include the code for a database class by one of our members, Robert Thille, accompanied by an article explaining how to use it. You'll also find a FrameMaker alert and several other items, including a press release and a few significant pickings off the net (both regular features of *SCaNeWS*).

FEBRUARY INTRO

Mike Mahoney
mahoney@beach.csulb.edu

The January 24 meeting at CSU Long Beach was attended by about 50 people, many we hadn't seen previously. The meeting featured an IBM RS/6000 running NeXTstep 1.0, demos of TouchType and the NeXTstep 2.0 user interface, and a lively discussion of "what can a NeXT do that a Mac or other machine can't". I really enjoyed the discussion part and learned a few things myself. Exchanging knowledge about how to use NeXTs better is obviously an important function of our users' group.

We have been promised a color NeXTstation at our March 28 meeting at CSU Long Beach. The meeting will probably be very well attended so get there early.

If anyone out there would like to host the April 25 or other future meeting at their location please let us know. Meetings usually occur on the last Thursday of each month. Two essential pieces of hardware are a color NeXTstation and a high-end projector (e.g. Electrohome Graphics or Barco) which can handle a NeXT's 63 kHz horizontal scan rate. An alternative to the high-end projector is a combination of a $2500 Extron board and good projector. The Extron board, which fits into a NeXTcube's slot, cuts the scan rate in half and thus increases the range of projectors which can display a NeXT's screen. (For further details on the Extron

(continued on page 2, column 2)

CONTENTS

The Databaser 2
IBM RS/6000 & NeXTstep 3
***SCaNeWS'* Role in the NeXT User Community 4**
Downloading *SCaNeWS* 4
NeXTAnswers 5
Sun and NeXT Benchmarks 5
Mac vs. NeXT Environments 6
FrameMaker 2.0 Alert 6
NeXT Scores with Campuses 7
Software Donations 8
Frequently Asked Questions 14
Computer Language Productivity Award 19
X Update 19
Creating Your Own Folder Icons 19

THE DATABASER:

A Simple Database

Robert Thille
rthille%nextoid@hub.ucsb.edu

BLACK BOOK

Robert Thille is a graduate of UC Irvine with a BS in CS. He is tired of odd jobs and is looking to settle down with a job in software development.

When I moved from my Apple //e to my NeXT I missed a few things. Wizardry was one of them, but I couldn't do much about that. However, I could do something about the lack of a simple database to keep track of my addresses and phone numbers. I wrote the Databaser class with that in mind, though it will keep track of just about anything that you can put into a Tab separated text file. It will read files of the format:

field1<tab>field2<tab>field3 <tab>field4(etc)<newline>

The class is very simple, having only twenty five methods, seven of those being outlet initialization methods used by IB[1] (Interface Builder), and three functions. There are nine IB callable action methods: add:, remove:, previous:, next:, prevByField:, nextByField:, setFields:, terminate:, and appDidInit:. The first two are for adding and removing records from the database, the next five methods deal with searching the database, terminate: and appDidInit: are shutdown and start up methods. The remaining methods are internal methods used by the class to support the external interface.

1. May be omitted in NeXTstep 2.0.

Using the class is very simple. Create a new project in IB. You need to add the Databaser class to your new application. Do this by dragging the Databaser.h file from a File Viewer to the Application window in IB. Add the Databaser.[hm] files to your project. Go to the Classes window and pull down the Operations list to Instantiate. This creates a Databaser with which you can then make connections.

Now you will need to add the objects which handle the user interface. You will need four Buttons,

(continued on page 13, column 1)

FEBRUARY INTRO

(continued from page 1, column 2)

board, see page 62 of the Fall 1990 Software and Peripherals guide.) For any meeting we'll send out the announcements, provide the goodies, arrange the speakers and bring copies of newsletters.

Future contributions to *SCaNeWS* include an intro to Objective-C for C programmers and a user interface and Interface Builder tutorial. We'll also have reviews of Diagram!, Improv, SoftPC and co-Xist (X Windows). Please let us know what else you like to see included.

In a future issue we'll also go into depth about the kind of features you get with NeXT and UNIX networking (for a start, see the Mac/NeXT comparison). Many of us at CSU Long Beach are "addicted" to "the net". Being able to easily mail application documents, folders or programs with a single drag is a great time saver and "interpersonal" communication tool. I regularly communicate with students, faculty, developers and NeXT employees via the net. I can send and receive both ASCII and application-specific files without changing their format. For example, all I have to do to mail a WriteNow (or any other) document to co-editor Lorraine is to drag its icon into a Mail.app Send window, click her address in the Addresses window, and click the Deliver button.

Having received the message Lorraine can either drag the icon into one of her folders in the File Viewer or double-click the icon to open the document within the WriteNow app. It's that simple! This kind of communication is available not only locally, but anywhere on the Internet. For example, we received TouchType and BioTRACE demo programs (from out of state) and the Databaser screen dump in a similar way through Mail.app.

Speaking of the Mail.app, our treasurer Henry Chiu has set up a picture database containing a set of SCaNned pictures of CSULB users. When you read a message from a user, a 64 x 64 pixel image of that user's face appears in the upper right corner of the mailbox window which contains the message - what fun!

IBM RS/6000 & NeXTstep:

A Comparison of the IBM and NeXT Implementations of NeXTstep 1.0

Henry Chiu and Mike Mahoney
chiu@beach.csulb.edu

IBM loaned us an IBM RS/6000 model 320 computer with 12 MB of RAM for the month of January. Since there is still a possibility that IBM will obtain the rights to offer NeXTstep 2.0 on the RS/6000 platform, we thought you might be interested in a list of differences between the IBM and NeXT implementations of NeXTstep 1.0. (The latest rumor we heard on a 2.0 deal is that Steve Jobs and IBM are still negotiating but there is no assurance that any deal will be made.) If IBM does port NeXTstep 2.0, then it will surely be quite different and better than its 1.0 port.

Here are the differences we noticed. Some are trivial and some are significant. Make your own conclusions. The NeXT machines we used for comparison had only 8 MB of RAM.

- The **NeXT symbol** in the login panel is replaced by an **AIX symbol**.
- The **login process** on the RS/6000 is two to three times faster than a 68030 NeXT and is about the same speed as a 68040 NeXT.
- A **PostScript program** we executed in Yap on the RS/6000 took about 2500ms. The same PostScript program took 8400ms on the 68030 NeXT computer and 2200ms on the 68040 NeXT.
- The **NeXTstep screen contents** (browser, dock, apps) on the IBM was the same as on the NeXT except that the Workspace Manager icon was replaced by an AIX icon. The text looked fuzzy on the IBM screen. Perhaps it was because the IBM had a color screen displaying monochrome or because the monitor had been banged around a bit.
- There is **no default database** in IBM's NeXTstep and therefore you cannot use the dwrite command to modify default values (e.g., font, size) in your working environment.
- The left **Alt key** on the RS/6000 keyboard acts like the NeXT **Command key** and the right Alt key is the actual Alt key.
- A major advantage of the IBM system is that one can press a **hot-key** and switch from NeXTstep 1.0 to **OSF/Motif running on top of X Windows**.
- IBM's NeXTstep comes with **only three NextApps**: Terminal, Edit and Interface Builder (we understand this is one of the sticking points in the IBM/NeXT negotiations for NeXTstep 2.0). For example, the IBM does not have the useful Digital Librarian or Mail.app applications. However, it does have the Info Explorer, a substitute for the Digital Librarian.
- The **Music kit and Sound kit are absent** (the RS/6000 doesn't have much sound capability without a DSP chip).
- **Interface Builder** interface (.nib) and project (IB.proj) files created by NeXT's NeXTstep 1.0 can be used by IBM's NeXTstep 1.0. However, several things need to be done before "make-ing" the application. Open the nib file and generate a new Makefile in IBM's Interface Builder and save the nib file. If the source code doesn't have any UNIX system calls or C library functions, then your program will probably compile without any problem. We found that not all C library functions were compatible in the two systems. For example, the getenv("HOME") library function call on the RS/6000 needs a different argument.

  Also, the "-" character is not allowed in filenames as the compiler will treat it as an

(continued on next page)

RS/6000 & NeXTStep

(continued from previous page)

error. If you look at the **Makefile generated by IBM**'s Interface Builder, you will see a comment line "#AIX version --- can't use '-include'.". IBM removed "-" characters from the beginning of lines in the Makefile (e.g., "-include Makefile.preamble", "-include Makefile.postamble"). Together with the Makefile, IBM's Interface Builder generates empty Makefile.preamble and Makefile.postamble files. This is one of the reasons that you need to create a new makefile when transferring apps from the NeXT to the IBM.

- IBM **Interface Builder files** will work on a NeXT running NeXTstep 1.0 but not NeXTstep 2.0. However, NeXTstep 2.0 can read 1.0 files generated by NeXT computer.
- There are **two C compilers** on the RS/6000. The standard C compiler is in /bin and the Objective-C compiler, objcc_shr, is in /usr/lpp/NextStep/usr/objcc/bin.
- **NeXTstep is installed** in /usr/lpp/NextStep and /usr/lib/NextStep on the RS/6000.

SCaNeWS' Role in the NeXT User Community

Our goals in distributing *SCaNeWS* are modest. We are geared more toward the needs of end-users, students and faculty than those of developers. Developers may find our newsletter interesting but perhaps not as technically oriented as they may like. The technical articles we include will be primarily introductory for the next few months. Among our regular features we plan to include product reviews, tutorials for basic software development as well as for some of the more complex applications, bug reports and technical alerts, updates on how NeXT is faring in the marketplace (with an emphasis on the comings and goings of third-party vendors, ever-crucial to a product's success), and pointers to the most reliable sources of NeXT information.

We refer developers and other sophisticated programmers to the *NeXT Users' Journal*, which can be downloaded from the same archive site as *SCaNeWS* (see the box "Downloading ..." on this page). We read it and love it (thanks, Erica!) and occasionally make copies available at our meetings.

Downloading SCaNeWS from an Archive Site. . .

If you missed the January issue (or any future issues) of *SCaNeWS* and have access to the Internet, you can obtain it via anonymous file transfer protocol as follows (**bold** indicates on-screen prompts, *italics* what you type):

At your local system prompt, type
ftp -n nova.cc.purdue.edu
ftp> *user anonymous <your_account@ your_address>*
(e.g. scan@beach.csulb.edu)

ftp> *binary*
ftp> *cd /pub/next/Newsletters/SCaNeWS*
ftp> *ls -l*
This will give you a listing of all the newsletters currently available. To retrieve an issue:
ftp> *get <filename>*
Repeat the last command for as many issues as you want to retrieve. To return to your local machine:
ftp> *bye*

The file is in compressed form (you can tell by the .Z extension). To uncompress it, type *uncompress <filename>*. Now the newsletter is ready to be loaded into Preview and printed.

NeXT ANSWERS

The January release of NextAnswers is now available on the Internet archive servers. We recommend that you obtain the latest release of NextAnswers and have it installed on the NeXT computers at your site.

NextAnswers should soon be available at an additional archive site:

hostname: umd5.umd.edu
IP address: 128.8.10.5
directory: NeXT/NextAnswers

The NextAnswers release is also available by email from the Purdue archive server, for those without ftp access. Send a message consisting of "help" to archive-server@cc.purdue.edu.

If the "From:" line in your message header will not be in Internet format (e.g. name @site.edu), include a line like:

pathmrd@sun.soe.clarkson.edu
or:
path wellknownsite!yoursite!
yourname@uunet.uu.net

Note that name@site.bitnet or name@site.UUCP won't work.

There is also a directory called TechSupportNotes on the archive servers, which contains longer Technical Support Notes from NeXT. There is one new TechSupportNote since the November release of NextAnswers, called 2.0EntryPoint.rtf. This is a particularly important one for programmers; it explains differences between 1.0 and 2.0. It is 65K in size.

For more information about obtaining and installing NextAnswers, see the January 1991 issue of *SCaNeWS*.

BENCHMARKS

DHRYSTONE 1.0 : OPTIMIZED, REGISTERS

Sparc 2	42075/sec	(11 secs)	SUN C, -04
NeXT 68040	28142/sec	(17 secs)	GNU C from rel 2.0
NeXT 68040	27223/sec	(18 secs)	GNU C from rel 1.0a
Sparc 1	21382/sec	(23 secs)	SUN C, -04
Sparc 1	10467/sec	(47 secs)	SUN C, no specified opt
Vax 8650	9436/sec	(52 secs)	
NeXT 68030	7884/sec	(63 secs)	GNU C, Shell
SUN 3/260	7028/sec	(71 secs)	SUN C
NeXT 68030	6866/sec	(72 secs)	SUN C, Shell
NeXT 68030	6785/sec	(73 secs)	GNU C, Terminal
NeXT 68030	5920/sec	(84 secs)	SUN C, Terminal
SUN 3/60	4964/sec	(100 secs)	SUN C
SUN 3/80	4856/sec	(102 secs)	SUN C
SUN 3/140	3794/sec	(131 secs)	SUN C
Vax 780	1712/sec	(292 secs)	
SUN 2/120	1390/sec	(359 secs)	SUN C

WHETSTONE (FLOATING POINT) BENCHMARKS

Sparc 2	125.0000/sec	(8 secs)	
NeXT 68040	38.4615/sec	(26 secs)	
Sparc 1	30.3030/sec	(33 secs)	
NeXT 68030	14.9254/sec	(67 secs)	w/68881
VAX 8650	7.0423/sec	(142 secs)	
Sun 3/140	6.7114/sec	(149 secs)	w/68881

These benchmarks were performed at ISI labs

NeXT SCaN MEETINGS

March 28 - CSULB

April 25 - ???

May 30 - CSULB

Editors' Introduction: Conrad Geiger is the NeXT Users' Group Coordinator at NeXT (invaluable to us!). In this article, he discusses the extent to which NeXT's commitment to the UNIX operating system, the unified imaging model provided by Display PostScript, and the object-oriented design paradigm renders the NeXT a superior environment for both applications users and software developers.

Mac VS. NeXT:

THE UNIX, DPS & OOP ADVANTAGES

Conrad Geiger

A Comparison of the Apple Macintosh and NeXT Environments

MultiFinder is built into Apple's standard system software as an extension of the original single-tasking Finder application. Multitasking was not designed into the original Apple OS. MultiFinder is better described as a "task switcher" than a true multitasker. In other

MultiFinder vs. Multitasking

words, MultiFinder allows several applications to be open, but only one can be active. It provides for an environment whereby the user can access one application, then switch to another application without being forced to close the original application, as was the case in the original Finder environment.

(continued on page 9, column 1)

FRAMEMAKER 2.0 ALERT

NeXT Technical Alert 91-1

Summary

The version of FrameMaker shipped with Release 2.0 doesn't display some characters properly, and has a problem with FrameMath. Also, the FrameUpdate package shipped with Release 2.0 doesn't work properly.

Description

Users of FrameMaker under Release 2.0 may notice some of these problems:

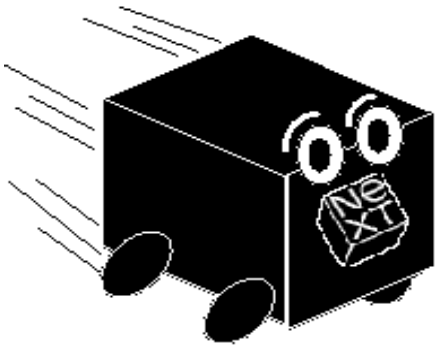
1. Some characters may not display or print properly.
2. If a FrameMaker document is printed to a NeXT machine from a non-NeXT machine, some characters may not print properly.
3. The Pop-up lists on the panels used to manipulate mathematical equations do not work properly.
4. FrameUpdate makes FrameMaker versions 2.0 and 2.0a unusable. FrameUpdate has less severe effects on other versions of FrameMaker, but its use is not recommended.
5. Printing on 68040 machines is very slow.

Most of these problems are fixed in FrameMaker 2.0d. However, printing FrameMaker documents to a NeXT machine from a non-NeXT machine will still not work. This problem will be fixed in future FrameMaker releases of FrameMaker for each non-NeXT platform.

Obtain the new release

FrameMaker 2.0d. Higher Education customers should obtain a copy from their NeXT Campus Consultants or Campus Support Center. All other customers can receive FrameMaker 2.0d by return mail from NeXT. Customers can get

(continued on page 8, column 3)



NeXT Press

from the Wall Street Journal, 1 February 1991

NeXT Scores Big With Campuses On Second Try

By G. PASCAL ZACHARY
Staff Reporter of THE WALL STREET JOURNAL

Don't count Steven Jobs out just yet. His closely held personal computer company, NeXT Inc., was written off by most competitors after its first product flopped two years ago. But now, his latest computers are selling briskly on college campuses, seeming to fulfill a goal he set five years ago when he stepped down as head of Apple Computer Inc.

Mr. Jobs, who helped to pioneer personal computing when he ran Apple, started NeXT with the idea of supplying college educators and students with an advanced computer capable of mixing huge amounts of audio, video and text. Mr. Jobs, himself a college dropout, vowed to deliver the electronic equivalent of an entire library of English literature or a sophisticated chemistry lab. But his original machine bombed because it lacked power and software application and, at $6,500, was too expensive. Mr. Jobs. who oversaw the creation of Apple's novel Macintosh computer, seems to be showing signs of his old spark. NeXT's newest models, which began rolling off the assembly line late last year, contain state-of-the-art hardware and sophisticated software. They are selling to students and faculty for as little as $3,000, less than half the price of the original machine and much less than comparable computers sold by rivals.

NeXT's latest models haven't sparked as much enthusiasm in the corporate market, which is crucial to long-term success. "We haven't felt the impact of those new products yet," said a spokesman for Businessland Inc., which is NeXT's main distributor to corporations. Still, Mr. Jobs said that half of 4,000 new models he's shipped in the past two months have gone to business. While those unit volumes are just a fraction of Apple's monthly sales, Mr. Jobs said NeXT is posting monthly sales equivalent to an annual rate of more than $100 million.

Vicki Brown, an analyst at International Data Corp., thinks NeXT is on target to sell from 36,000 to 44,000 units this year. University customers are the most eager buyers. Mr. Jobs has deep ties with academics from his days running Apple, the longtime leader in the education market. Some, who have stuck with Mr. Jobs through his recent ups and downs, say he is making good on his vow to set a new standard for academic computing. "I think he has met many of his original goals, '' said Brian Hawkins, vice president for computer services at Brown University. He said purchases of NeXT machines by Brown staff and students "have picked up considerably" in recent months. "NeXT finally has the machine it promised; that's fantastic," added Edward Barboni, vice president of information at Allegheny College. He thinks NeXT has staying power. "We've had opportunities to peek at NeXT's future products," he said, adding, "What's coming down the pipe is more good stuff."

Such praise is good news for Mr. Jobs, but he acknowledges that. parts shortages have prevented NeXT from taking full advantage of the situation. Shipments of the latest models were first held up by Motorola Inc. which makes the 68040 central processor for the machines. Then, output was slowed when the hard-disk drives used in the flagship model proved defective, forcing NeXT to switch suppliers. The limited availability

(continued on next page)

NeXTPress

(continued from previous page)

of product has cost NeXT. "People have been holding off buying because delivery times are so far out," said Chester Rice, who manages the computer store at Stanford University. Mr. Jobs acknowledged that NeXT hasn't been able to meet demand, but said the company has solved its supply problems and now is reducing its backlog.

NeXT is facing bigger obstacles selling to corporations. Mr. Jobs had bet that Businessland Inc., one of the largest computer dealers, could carry most of the sales load, but the move hasn't panned out and last September NeXT said it would be- gin looking for additional sales help. He declined to discuss plans, but he confirmed that he recently signed pp Computer Attic, a retail chain in Northern California.

Nevertheless, NeXT may continue to find big business a hard sell. NeXT computers run software that isn't compatible with programs written for any other leading brand of computer; while the number of NeXT applications is growing, only a few of the most popular programs run on the machine. As a result, the new models aren't "an instant hit with business," said Walter Feigenson, manager of special products at Ashton-Tate Inc., which sells a spreadsheet for NeXT's computer.

That's tough, because Mr. Jobs needs his machines to win wide acceptance quickly to keep pace with stalwarts such as International Business Machines Corp., the largest supplier of desktop computers, and younger, friskier rivals such as Apple and Sun Microsystems Inc., both of whom are gaining market share by the month on the strength of new models. "A lot of people are still taking a wait- and-see attitude," said Jerome Smith, director of workstations at the University of California, Berkeley, adding, "People want to see what happens to the company."

Mr. Jobs still might receive a boost from IBM, which several years ago licensed NeXT's basic software with a notion of offering it with IBM-made computers. At the time, some people speculated that IBM might "anoint" NeXT's software as a standard for a certain class of personal computers called workstations, but so far IBM hasn't released its own version of NeXT's software. James Cannavino, who heads IBM's desktop computer business, says IBM is still studying NeXT's software and has yet to poll customers on whether they would buy an IBM version.

All this adds up to a promising, but still inconclusive year for Mr. Jobs, who has plenty of cash on hand-Canon Inc. invested' $100 million in NeXT 18 months ago-but not as much patience. After Christmas, he issued a stern message to his staff, urging them to redouble their efforts and postponed the company's annual "retreat" to a seaside resort.

Mr. Jobs says he hasn't tired of the computer industry, and isn't daunted by intensifying competition. "I've spent my entire adult life making computers," he said. "I love it. It's in my blood."

(continued on page 19, column 2)

FrameMaker Alert

(continued from page 6, column 3)

the optical version by sending a blank optical disk to NeXT, or the floppy version by sending their address with the request.

For More Information

All Higher Education customers should contact a Campus Support Center to obtain a free copy of FrameMaker 2.0d. For all other customers who need to take advantage of the optical disk swap program, mail an optical disk to:

FrameMaker 2.0d Upgrade
NeXT Telebusiness
900 Chesapeake Drive
Redwood City, 94063

Please be sure to include your return address to insure that your FrameMaker 2.0d is routed correctly. Customers that have floppy disk drives can mail their return address to the above address to receive FrameMaker 2.0d on floppies.

For further information call NeXT Telebusiness at 1-800-848-6398.

SOFTWARE DONATIONS

The following software has been donated to SCaN:

Diagram! (Lighthouse Design)
Improv (Lotus)
TouchType (RightBrain)
WordPerfect (WordPerfect)

We thank the comapnies involved.

MAC VS. NeXT

(continued from page 6, column 1)

However, once the user switches to the new application, the original application goes into a hold state, and is left inactive. This prevents the user from being able to allow an application to continue processing while switching to another application to work with interactively. For example, if the user wishes to include data from a database into a report that is due that morning, he or she would typically start a query into the database in one application, and then start editing the report inside of a word processor while the database application is performing the query. This is not possible in the MultiFinder environment, since the word processor would take over, leaving the other application in a hold state.

In an environment such as NeXT, with a UNIX operating system, true multitasking is the standard. Therefore, the above scenario would be possible even to the extreme that the user can have several programs running simultaneously, giving the effect that each program is running with its own processor. This is completely transparent to end users: they don't have to think about how to do their work, they just perform the activities as comes naturally.

The design of MultiFinder is based on the concept that each application thinks it is running in a typical Finder environment. Each application thinks it has access to all of the memory it needs, and to all of the resources of the computer. MultiFinder simply keeps track of who is loaded in which part of RAM, and informs applications of how much RAM is available when they are launched. Thus, each application runs in what it considers to be an exclusive mode, or non-sharing environment.

This is true in UNIX as well; however, this is because UNIX provides a virtual environment to each application, and handles the conflicts of resources between applications in the operating system. For example, in UNIX each application thinks it has an abundance of memory available to it, regardless of how much RAM is available on the system, or how many other applications are running. This is called "virtual memory," and allows the end user to run several applications simultaneously, even though there is not enough RAM to hold all of these applications. Thus, the operating system takes care of paging to and from the disk, providing for the virtual memory environment, and making it appear as though there is enough RAM to hold everybody.

Since the MacIntosh does not support this virtual memory concept, all of the applications that the user wants to run must fit within the limits of physical RAM on the machine. This prevents the user from being able to launch several programs at once. Depending on the configuration of the machine, the user can be limited as to what combination of applications can run because of the size requirements for the applications. It is quite common to launch an application in MultiFinder, and have the system come back and say there is not enough memory left for this application. The user must then choose another running application, and close it, in order to bring up the desired application. This is a scenario that is not encountered in the NeXT environment.

Task-Switching

Another fundamental distinction between the environments is in the way they switch from one task to another. Since there is physically one processor in both the Mac and the NeXT, the CPU must give out time slices to each process to keep the appearance of multitasking applications. There are two ways of determining how often each task gets a time slice, and how long each slice lasts: preemptive multitasking, and cooperative (or non-preemptive) multitasking.

In preemptive multitasking, which is utilized by UNIX, the operating system can shift the CPU resources to another application without the current application even knowing. Therefore, a single application cannot take control of the CPU, and halt everything else on the system. With cooperative multitasking, each application determines for itself when it will give up the CPU to another program. The application must then save its environment before giving up the CPU. This means that a single application may never give up the CPU once it starts running. This is the case with many of the applications in the MultiFinder environment, which take control of the CPU and stop all background processing.

(continued on next page)

MAC VS. NeXT

(continued from previous page)

This also leaves the entire environment exposed, such that if the currently running application crashes, typically the whole system comes down. This is because there is not enough isolation between applications, as there is in the UNIX environment.

Another advantage of preemptive multitasking is the ability for the operating system to adjust the time slices when an application is blocked, waiting for a particular device that is not currently accessible. Therefore, the OS is much more efficient in its use of the CPU. In fact, with the NeXT environment, there are special DMA (Direct Memory Access) channels included in the hardware architecture, which will allow a process to access things like memory and the disk drives without the need for the CPU to be involved. This overall design dramatically increases the effective throughput of the system, adding to the efficiency of CPU utilization.

Interprocess Communication

Since the MacIntosh environment was not designed to be multitasking, it does not include facilities for interprocess communication. Therefore, the only way for two applications to communicate with one another would be through either shared memory (the "clipboard") or a shared file (the "scrapbook"). This would require a formal understanding between the two applications, that must be designed and developed within each.

The NeXT environment provides for standard interprocess communication facilities, which have been made into standard application objects (speaker/listener objects), which allow for simple and consistent access to inter-application communication. NeXT has even gone one step further with its "Services" facility, which is a common menu item object that allows for generic communication to outside applications from within an application, without the application having to know who it is talking to. This allows an application access to generic functions or services provided by other applications that may be relevant to the current task at hand.

All of this is made possible from NeXT's utilization of the Mach kernel, which supports a common mechanism for interprocess communication (Mach ports), and from the fact that NeXT is an object-oriented environment.

The most common way in the MacIntosh environment to transfer data between applications is the "clipboard". NeXT has included this support in its environment as well, giving a consistent method for a user to transfer data from one application to the next, or even within an application. In the MacIntosh environment, the clipboard is based in RAM, and therefore is limited by the space available in RAM after the other applications have taken their chunk. In the NeXT environment, the pasteboard can grow larger, due to the virtual memory environment. In addition, only the NeXT environment can support the direct dragging of an object from one application into another application running on the display.

For example, a NeXT user can drag an image directly out of an email message and into his or her word processing document to use in the report they are currently working on. This is accomplished by selecting the graphic with the mouse, and dragging over to the word processing document, and releasing. No menus needed, no "Copy" and "Paste" needed, and no switching between applications, since the environment is truly multitasking.

In the NeXT environment, applications co-exist with each other and work together to help the user solve a problem. In the Apple MacIntosh environment, the applications exist independently of one another, and the user is responsible for tying together the necessary functions to solve a problem.

Security

One deficiency of the Apple environment is security. The MacIntosh was designed to be a single-user system. This means that anyone who has access to the power switch has access to all of the data that resides on the disk. There are no inherent security features. NeXT however, utilizes the multi-user environment of UNIX, allowing for different people to have different access to the files on the system. Depending on the needs of the environment, the access can be opened up or severely

(continued on next page)

MAC VS. NeXT

(continued from previous page)

restricted to the users of the system.

File Sharing

Another advantage that NeXT has over the Apple MacIntosh is the sharing of files between machines. NeXT utilizes standard facilities, like NFS, for file sharing. This allows file systems on each machine to be accessible to any other machine on the network if so desired. Each machine on the network can in effect be a file server.

It is this capability that allows the NeXT environment to be configured such that users can log into any machine on the network, and see their own custom environment, as though they had logged into the system on their own desk. The only way to do this on the MacIntosh is to replicate the environment on every machine, for every user.

Imaging Models

The core technology of all GUI's is the imaging model: the language and paradigm used to put images on the screen and the printer. The NeXTstep window server does what no GUI has ever done before: it unifies the imaging model for the screen and the printer. This is made possible by NeXT's decision to utilize Display Postscript for the screen, and Postscript for the output language. In addition, NeXT worked in combination with Adobe to define and enhance the Display Postscript environment, in order to maintain compatibility with the evolving Postscript standard.

The Mac OS, in contrast, draws to a computer display with an entirely different set of instructions (QuickDraw) than is used to draw on the printed page (Postscript). This dichotomy causes inconsistency between what you see on the screen and what is seen on the printed page. Often it will take several print jobs from a MacIntosh to get the printed page to look as desired.

Apple has recently decided to correct this situation by converting to a unified imaging model as well. However, Apple's approach is to standardize on their own proprietary QuickDraw language. This moves Apple away from the industry standard Postscript output language. Therefore, they must continue to support Postscript in order to be compatible with the existing installed base of Apple laser printers, and the tremendous amount of Postscript based third party output devices in the industry. This means that applications must then be able to support the industry standard Postscript, as well as Apple's own QuickDraw.

With the NeXT unified imaging model, the developer only has to write one set of instructions to display the application's data, and that will take care of the display on the screen, as well as output to a hardcopy device. This reduces the amount of application code, and the time to develop the application. This also keeps the application consistent with the defacto industry standard output language.

Development Environment

The standard application that people think of when it comes to doing development in the Apple MacIntosh environment is HyperCard. HyperCard has been successful on the MacIntosh for a couple of reasons. First, it was bundled with every system. Second, it introduced the concept of an authoring system where programming is an assembly process, rather than strictly code generation. However, going beyond the simplicity of a HyperCard application, leaves a user on his or her own to develop from the ground up. HyperCard does not scale with the application, and is not extensible to allow for the creation of complex custom applications. It is limited to its basic elements of cards, stacks, buttons, fields, and backgrounds. The programming language, HyperScript, is very limited in its capabilities, and falls far short of a flexible programming language.

The NeXTstep environment includes tools for the assembly of applications as well. However, these tools will work for complex custom development as well as for simple assembly. They work in conjunction with standard higher-level programming languages, such as Objective-C, C++, and ANSI C. The NeXTstep Interface Builder, which allows for the interactive assembly of an application, works in combination with the user's choice of programming languages, and is completely extensible to whatever degree of

(continued on next page)

MAC VS. NeXT

(continued from previous page)

complexity is necessary for the application. The user is never left to start from the ground up to develop the application.

The other advantage the NeXTstep environment has over HyperCard is that it is truly object-oriented. NeXT is committed to the object-oriented paradigm, and has designed the environment from the ground up to be consistent with this. The end result is a set of tools and object kits that directly support the object-oriented paradigm. It is this implementation that gives the user complete flexibility through the subclassing of existing objects, and complete inheritance from the existing toolkits. This is not supported in HyperCard because of its lack of a true object-orientation. The user is relegated to the existing 5 entities provided by HyperCard.

Hypercard also lacks polymorphism.[1] Polymorphism underlies the ability in the NeXT environment to have multiple objects with the exact same method names implemented within each. Due to the runtime binding provided by Objective-C, an application can send a message to an ambiguous object, referencing it through a variable, that is bound at runtime to an actual or physical object instance. This gives the developer much more flexibility in the development process and structure of the application. The developer is not forced to know who or what he or she will be talking to, and calling out a specific function based on the appropriate object. This is one major difference between programming in a true object-oriented environment, versus utilizing a function library to access pseudo-objects.

HyperCard applications are not truly standalone applications. They are actually programs that run within the actual HyperCard application, and are typically interpreted. This contributes to Hypercard programs being extremely slow. It is not very realistic to do a sizeable project in Hypercard. It works well for small programs like an address book, etc., but for anything more complicated the performance (i.e. speed) becomes a major issue.

The NeXT development environment is much more robust. Developers have Interface Builder to interactively create the application. They have the Application Kit for accessing standard objects, and the Sound and Music Kits for accessing the sound capabilities of the NeXT environment. Because of the object-oriented environment, and Interface Builder's dynamic palette capability, they will have access to other custom object kits or palettes in the future, such as the Database Kit.

The environment is fully customizable by the developer, and objects can continue to be reused across applications. The Interface Builder environment also includes a project management facility that is lacking in the MacIntosh environment. In general, NeXT provides a complete development environment that is well integrated with the system, as opposed to a simple application such as HyperCard.

The Macintosh employs a primitive development environment for complex programming when compared to the NeXT. Apple now produces a "Macintosh Programmer's Workshop" for the Mac development community. Unfortunately, it is an altogether new environment deviating from the GUI that is so familiar to Mac users. MPW does support C++, an object-oriented version of the C language, but since the OS code and ROM functions were originally written in assembly and Pascal language, much software "glue" is needed for interface between the developer's code and the underlying OS.

To combat this inconsistency, Apple has advocated a new environment know as MacApp. MacApp is described as both a stand-alone application shell and a set of reusable libraries upon which to base your program. MacApp is written in a superset of Pascal called ObjectPascal. MacApp does help to rectify some of the difficulties found in developing for the Mac, but it still suffers from the foundational problems that the Mac OS was not developed using the object-oriented paradigm and that ObjectPascal is not a popular language with programmers in general.

Editors' Note:

In all fairness we should mention that the Mac's System Software 7.0 improves the Mac in several

(continued on next page)

1. polymorphism: In object-oriented programming, the ability of different objects to respond, each in their own way, to the same message

MAC VS. NeXT

(continued from previous page)

important areas. File sharing, virtual memory, multitasking, and outline fonts are supported in varying levels. However, it still ain't UNIX, NFS (network file system) and Display PostScript. On the other hand, why doesn't Apple push UNIX (and for that matter, Display PostScript)? While our local UNIX experts found ways to kill A/UX (Apple's implementation of UNIX) at a recent Apple demo, it was great to see the Mac interface running as a process under UNIX. (We wonder how long it'll take before we can run the Mac Interface in a NeXTstep window, or even NeXTstep on top of A/UX?) Let's face it, the best reasons to buy a Mac are its great variety of software and its installed base, not its superior technology.

THE DATABASER

(continued from page 2, column 1)

two Forms, and a Matrix of check-box Buttons. Drag these from the Palettes window. Drag one of the handles while holding down the Alternate key until you have as many cells as you have fields in your data. This is the dataForm, the other form is the keyForm. I have six: Name, Address, City, State, Zip, Phone.

The Palettes window doesn't have a matrix of check-boxes, so you need to drag a single check-box Button to your window then Alternate-drag like you did with the Form. Again, you want as many check-boxes as you have fields. I named them the same as the fields, in the same order. On both matrixes you should have "Cell tags match positions" turned on. The buttons should be named Add, Remove, Next, and Previous. See the graphic on page 20 for the assembled window.

Now we will make the connections to the DatabaserInstance we already put into the Application. First, some general things about making connections: you hold down Control and drag from one object to another to make a connection, then in the Inspector Panel select the action and outlets you wish to connect, then press the Connect button.

We need to connect the buttons to the DatabaserInstance so that we can add, remove and search records. Control-drag from the button in the main window to the DatabaserInstance in the Application window, select the appropriate action (add, remove, etc.) and press the Connect button. Do that for all the buttons.

Connect the Next and Previous buttons to their respective By-Field actions. This lets you search individual fields in the records. Then connect the "nextText" outlet of the forms to each other. This will allow you to Tab between the two forms.

Connect the target of the keyForm to the Next Button with action performClick:. This allows you to hit return to go to the next record.

Connect the target of the dataForm to the Form with the single cell with action selectText:. This allows you to hit return in the dataForm and be taken to the keyForm to search for records.

Connect the check-box Matrix to the setFields: action of the DatabaserInstance.

Connect the Quit item in the main menu to the terminate: action of the DatabaserInstance (you may need to disconnect it first).

Connect the Delegate outlet of the Application to the DatabaserInstance.

All that is left is to connect the DatabaserInstance to the various buttons and forms in the window. Simply drag from the DatabaserInstance to each of the Forms and to each of the buttons. Which outlet to use for each item should be easy to determine by its name. You do not connect the dataStorage outlet, that is used internally, and you need to be sure that you connect to the actual Forms, not the FormCells within.

Save the Application. Then with a text editor, open DBParams.h. This has some parameters for the Databaser object, and a function. The parameters are:

NUMFIELDS: set this to the number of fields you have in each record.

MAXLINELEN: this is the length of the longest line in your data plus some padding for safety.

PERMS: the permissions for the file, if it doesn't exist

DESC: the description for the Storage object within the Databaser object, the number here should be the same as the MAX-

(continued on page 20, column 1)

Editors' Note: The selection criteria for Usenet postings are based solely on their interest. SCaN makes no claim, explicit or implied, as to the accuracy of the information contained in these excerpts. We also assume that people who post on Usenet will enjoy seeing their words in print.

FREQUENTLY ASKED QUESTIONS

Pascal Chesnais
John Garnett
Erik Kay
next-faq@media.mit.edu

with thanks to Mark Adler (madler@pooh.caltech.edu) for answering the unanswered questions

The information contained in this post is a compilation of answers posted by participants in the comp.sys.next USENET group. No guarantees are made regarding the correctness of these responses. Note: I am not affiliated with any of the companies mentioned in this posting.

---

0. Where can I find additional questions and answers regarding the NeXT computer? (this question is really never asked but really should be)

Get NextAnswers for Digital Librarian from one of the many ftp sites that have NeXT software. NextAnswers is a series of questions and answers that the NeXT technical support has compiled. They tend to clarify issues not covered or poorly covered in the online documentation. If you have a question that is not covered in NextAnswers you can submit it to ask_next@next.com. They may not answer the question immediately.

If you don't have ftp access, you can obtain NextAnswers by sending an optical disk to:

NextAnswers Technical Support
NeXT Computer, Inc.
900 Chesapeake Dr.
Redwood City, CA 94603

Another good source of information is the archives of previously posted notes from the comp.sys.next newsgroup. All articles sent to comp.sys.next are archived on brownvm.brown.edu. Send a note to listserv@brownvm.brown.edu containing the command "INFO ?" on a single line (by itself). The list of articles is called NEXT-L.

---

1. How may I contact NeXT Computer, Inc.?

NeXT Computer, Inc.
900 Chesapeake Drive
Redwood City, CA 94063
Phone: 1-800-848-NeXT
(Redwood City #)
1-408-437-0400
(BusinessLand)

---

2. Is there a mail order company that sells NeXT-related peripherals?

Yes, its called NeXTconnection.

Address: 9 Mill Street,
Marlow, NH 03456
Phone: 1-800-800-NeXT

---

3. Is there a NeXT specific magazine?

Yes. There are also many NeXT specific newsletters, many of which are available via ftp from the NeXT specific ftp sites. The Boston Computer Society NeXT User group also produces a NeXT newsletter called "What's NeXT?." BCS may be reached at 1 Center Plaza, Boston, MA, 02108.

The first magazine is called NeXUS and is published bimonthly for $36/year. Contact Alfonso Guerra at {emory|gatech}!nanovx!nexus.

The second is called NeXTWORLD and is published six times a year ($23.95).

NeXTWORLD
501 Second Street
San Francisco, CA 94107
415-978-3196 (fax)
NeXTworld email address:
nextwrld!ddinucci@
uunet.uu.net

If you have any concerns about the NeXT computer that you would like to see appear in the "Letters to the Editor" section of the first issue of NeXTWORLD magazine, you may

(continued on next page)

OFF THE NET

(continued from previous page)

submit them to one of the following addresses:

NeXTWORLD
attn: Letters to the Editor
501 Second Street
San Francisco, CA 94107

Or electronically to: nextwrld!dlavin@uunet.uu.net, with a subject field of "Letters to the Editor"

There is a third called NeXT on Campus which deals with NeXT developments in academia. Call NeXT at 1-800-848-NeXT for subscription information.

---

4. What are the names of some of the ftp sites that have NeXT-related files?

There are too many to list them all, so are here are just a few:

cs.orst.edu
sonata.cc.purdue.edu
nova.cc.purdue.edu
aeneas.mit.edu (MIT GNU)
princeton.edu (music)
sumex.stanford.edu
cs.ubc.ca
umd5.umd.edu
ftphost.cac.washington.edu
ftp.ncsa.uiuc.edu.

See NextAnswers for more sitenames.

---

5. If I am not on the Internet, how can I get to the ftp sites via email?

Nova.cc.purdue.edu is configured as an email archive server. This means you can upload and download files via email. Send mail to: archive-server@nova.cc.purdue with the subject line "help" and you will get a complete description of this service.

If you submit material, you may want to send mail to lindley@cc.purdue.edu with a description of the material you are submitting.

---

6. What is the claimed performance of the new 040 NeXT machines?

NeXT advertizes the new 040 machines at 15 MIPS and 2 MFLOPS.

---

7. Where can I get the most recent version of Kermit?

The source for version 5A(163) (30 Nov 90) is available via ftp from watsun.cc.columbia.edu in the directory named kermit/sw. Get the file ckaaaa.hlp to get started and see what files are required. Note that this version of Kermit does have a NeXT specific compile option. A binary only version is available via ftp from cs.orst.edu as kermit5a.tar.Z. This file is currently stored in the /pub/next/sources directory but may be moved to /pub/next/binaries at some point in the future.

---

8. How much free space is available on the NeXTstation and NeXTcube?

As shipped, the 105MB NeXTstation has approximately 25.2 MB free. Note: The swap space is by default configured to 16MB with a low water mark of 20MB (the system does not attempt to reduce the size of the swapfile space until the swapfile grows past the low water mark).
The extended release takes 179MB, including a 16MB swap file, and not including Mathematica or Sybase (this is from a virgin install of 2.0 extended from an optical).

---

9. What software is bundled with the new 040 NeXT machines?

Both the 105MB and 200MB systems come with Release 2.0 preinstalled. The systems with larger hard disks have Release 2.0 Extended preinstalled. The price of all of the new systems includes the cost of the software license. Note that the price of the 040 upgrade for 030 machines does not include the cost of the software license.

RELEASE 2.0 includes:

End User Applications
Workspace Manager™
NeXT Mail™
WriteNow®Word Processor
Digital Webster™ (9th Collegiate Dictionary®and Collegiate Thesaurus®)
Digital Librarian™
Edit
Mathematica®:for higher education customers only
DataViz/Bridge™
Installer
FaxReader
Preferences
Preview for PostScript
PrintManager

Developer Tools
VT100™ Terminal Emulator [based on Stuart]

(continued on next page)

OFF THE NET

(continued from previous page)

System Admin Applications

BuildDisk
InstallTablet
Mail Manager
NetInfo Manager
NetManager
Printer Tester
User Manager
Installer

RELEASE 2.0 (extended) adds:

End User Applications

Oxford® Dictionary of Quotations
William Shakespeare: The Complete Works (for Digital Librarian)
TEX™ Document Processing System (Radical Eye Software)
Improv™ (Improv is only free if 040 NeXT machine or upgrade ordered before end of March 1991)

Developer Tools

Interface Builder™
Objective-C® Language Compiler
C++ Language Compiler
Objective-C Class Definitions, 56001 DSP Tools
GNU Emacs
GNU Debugger
BUG-56™ Debugger (Ariel)
Malloc Debugger
AppInspector™
PostScript Tools
Application Kit™
Music Kit™
Sound Kit™
On-line technical documentation

---

10. What type of microphones work with the NeXT?

Some NeXT owners use the RadioShack (Realistic) Tie Clip Microphone ($19.95) cat 33-1052. Others have successfully used WalMart (~ $6) and Sony brands (~ $60).

---

11. How do I connect a modem to the NeXT?

In short, for the 030 NeXT you need a Mac II serial modem cable. Note that some Mac II cables will not allow successful dial-in (because of incompatible pin wiring). Any Mac II cable should work for dial-out. The Businessland part number for a NeXT modem cable is reportedly 235739. Note that no Mac cable will allow hardware flow control. For those 040 owners wishing flow control, you may buy a cable from NeXTconnection or NeXT or make one yourself using the information in the zs(4) man page.

Most use tip or kermit to control the modem. SLIP and/or UUCP may also be used, but are more complicated to set up and require the remote machine to also have SLIP and/or UUCP (respectively).

For more information, it is highly recommended that you obtain and read the documentation from NeXT in their TechSupportNotes series called SerialPortDoc.wn. This document is available from most FTP sites that carry NextAnswers. Also, try to obtain the comp.sys.next note by Mark Adler entitled "How to connect a modem" (reposted 5 Jan 91) [Adler's note does contain info on the changes required to connect a modem to the 040 NeXTs]

---

12. Where can I obtain a NeXT version of X-Windows?

There is a X11R3 version named XNeXT developed by MIT. This version is available in binary only and does not work under Release 2.0 of the NeXT OS. XNeXT is a 1-bit black and white server. XNeXT is available from most ftp sites that have NeXT software.

McGill University (der Mouse) has produced a beta version of an X11R4 server named mouse-X available via ftp from 132.206.1.1. cd to X/XNeXT and fetch one of the files with names beginning with "distribution". This version only uses NeXTstep for keyboard and mouse events. This version has problems getting events under Release 2.0. McGill is reportedly waiting on some necessary technical information to be provided by NeXT Computer, Inc.

Pencom Software of Austin, TX has announced and has a working alpha version of X11R4 called "co-Xist". A 4-bit beta version is scheduled for February 1991 and a released version for March 1991. Color is planned in a future release. Cost has been announced at $149 educational, $249 commercial. Co-Xist runs inside of a NeXTstep window. Contact Pam O'Neal at pensoft!pam@cs.utexas.edu.

---

13. Where can I obtain a NeXT version of SLIP?

There is no publicly available version of SLIP for the NeXT; however, versions are under

(continued on next page)

OFF THE NET

(continued from previous page)

development by several sources.

---

14. What type of memory may be installed in a NeXT?

Use Mac II-style SIMMS, 100ns or faster. SIMMS must be installed in groups of four. The NeXT can use either 1MB or 4MB SIMMS. Most people are buying 80ns SIMMS in order to be more sure of future compatibility. Also, note that the low-profile vertically-mounted 4MB SIMMS are easier to install than the horizontally mounted 4MB SIMMS because of the small height clearance above the SIMM slots. It is possible to install the horizontally mounted SIMMS but you will be required to slide the CPU board and the hard drive in simultaneously. Note that parity (9-bit) RAMS can be used in both the 030 NeXT and the 040 NeXT. Parity and non-parity SIMMS can be mixed in both the 030 and the 040 (however, the boot on 040 boxes will generate a warning). Only the 040 NeXT can use the parity memory to detect parity errors (parity and non-parity may not be mixed if you wish to have parity errors detected).

---

15. What can be done about older 030 NeXTs that have a fan that turns in the "wrong" direction?

The fan on older 030 NeXTs sucks air out of the back of the cube which means that it draws unfiltered air in through the optical disk on the front of the cube. This causes optical disks to succumb to dust must sooner than cubes with the new-style fan which turns in the opposite direction.

According to NeXT, it is now okay to reverse the fan direction in older cubes [do not reverse polarity, just reverse the whole fan assembly]. Apparently, the original concern was that people would install third party hardware cards which when installed with the reversed fan, would overheat the cube. It seems that not many third party cards have been installed. You should also obtain the cleaning kit and dust filter from NeXT.

---

16. Are there any alternative sources for the SCSI-II to SCSI-I cable required to attach external SCSI devices to the 040 NeXTs?

Yes. This cable is the same as the one used by Sun SparcStations and DecStation 5000's (but not DecStation 3100's, which use 68-pin micro rather than the 50-pin micro connector used on NeXT 040, Suns and DecStation 5000).

---

17. What disk drives will work with the NeXT?

An internal or external hard drive has to have a SCSI interface and, according to hardware.586 in NextAnswers, must support:

1. "Inquiry" and "Read Capacity" commands
2. Conformity to the ANSI SCSI Rev. 17B or greater
3. 10 byte read and write command blocks

The drive must be formatted with either 512 or 1024 byte sectors. The latter will give you more storage on the same drive, and possibly some performance improvement.

Most any SCSI hard drives you buy nowadays (new) will conform, except for drives intended for the Macintosh, which have a slightly different firmware to handle Apple's nonstandard use of SCSI. Look for the hard drives in the weekly newsletters for the Unix/workstation world, instead of *MacWeek*. Just make sure they have a reasonable return policy.

Chapter 12 of the Network and System Administration manual covers the gory details of adding a drive. Note that under 2.0, /etc /disktab entries are created automagically.

In the cube, you can put a full height five inch hard drive. The station comes with a 3.5 inch hard drive, which I suppose can be replaced. You can usually find a SCSI box/power supply/centronics connector package for around $200 to put an external drive in (also budget a can of flat black spray paint). Either the cube or the station can take an external SCSI drive. It is also possible to make that the boot drive if desired.

A particularly good deal that people have been taking advantage of is the Fujitsu 2263SA: a 670M full height drive for around $1400. It has a five-year warranty and a 16ms average access time. A 1.4G Fujitsu is also available for around $2200. Fast Access (708-530-7749) has these.

(continued on next page)

OFF THE NET

(continued from previous page)

As for floppies, I am currently using a PLI external floppy drive that works just honky-dory. I originally wanted to get an internal floppy, since that would be cheaper (my 040 board already has a floppy controller on it), and possibly more compatible. However, I gave up trying to get the necessary mounting hardware from NeXT, which involves a new faceplate and new bays for the hard and optical drives.

---

18. What fax modems will work with the NeXT?

The Everex/Abaton InterfaxNX will. NextConnection will be selling it if they ever get them in stock. I hear they're black. It is a combination 2400 bps MNP 5 data modem and a 9600 bps group III, class 2 fax modem. (Group III refers to the level of fax-to-fax communication, and Class 2 refers to the command set used by the computer to talk to the fax modem. It's that class 2 that prevents any 'ole fax modem from working with the NeXT. All fax modems for PC's and Mac's are class 1.)

There are other suppliers of fax modems for the NeXT as well: call NeXT and ask for their Software and Peripherals Catalogue that lists third party hardware and software suppliers.

---

19. Can I run DOS software on the NeXT?

Yes, there is a product called SoftPC that has been announced for the NeXT and publicly demonstrated on the NeXT. Contact Insignia Solutions Inc. at 408-522-7600.

In addition, NeXT machines equipped with 2.88Mbyte floppy drives are able to read and write DOS diskettes (3-1/2" variety!) at 1.44Mb or 720Kb.

---

20. What comes with the 040 Upgrade?

An anti-static disposable bracelet, installation guide, new OD cable, OD filter, 040 board, a simm extractor tool (i.e. it comes with no RAM, which is pulled from your 030 and put on the new board), Fed-Ex return slip and registration card.

There are two mail-back cards: one for Lotus Improv and NeXTworld, and the other for Mathematica 2.0 and Sybase (eventually). This may change, of course, and may be different for non-educational users. The 040 upgrade requires the 2.0 upgrade.

---

21. What are some good references on Objective C?

There is an introductory tutorial for Objective C on the NeXT by Gerrit Huizenga available for anonymous ftp from nova.cc.purde.edu in the directory /pub/next/docs. Two versions are available: Objc.frame.Z or Objcframe.ps.Z.

For a more general introduction to Object-Oriented Programming, see Brad Cox's book *Object-Oriented Programming: An Evolutionary Approach*, Massachusetts: Addison-Wesley, 1986.

---

22. What alternative printers (laser or otherwise) may be used with the NeXT?

Anything you can hook up to the serial port. If it's a Postscript printer, then you can make it the system printer. Otherwise, you can just use it as a dumb ASCII output device. People have coerced Imagewriters to produce Postscript output.

---

23. How may I attach more than two serial ports to the NeXT?

You will have to buy a third party SCSI serial port box. They already exist for other workstations, and Uninet (800-433-6784) has drivers for the NeXT for their serial port extender. It also has Centronics parallel printer ports. Don't expect it to be cheap.

---

24. Can I run MacIntosh programs on my NeXT?

No. There is a third party company working on reading and writing of Mac diskettes.

---

25. Can I run tcsh on my NeXT?

If you are simply looking for emacs-like command editing you can do this with the csh distributed by NeXT by typing:

```
set editmode=emacs
```

In addition it is possible to bind control keys to commands with a .bindings file. Users should seek the csh(1) man page for more information.

(continued on next page)

OFF THE NET

(continued from previous page)

26. What fonts can I use on my NeXT?

Properly packaged Type 1 or 3 Postscript fonts will work on the NeXT, although certain conversions may be necessary. Freeware and shareware fonts are available on sonata.cc.purdue.edu in pub/next/misc. Each file unpacks into its own directory. Within each directory is a ReadMe.rtf and a Makefile. See the ReadMe.rtf for more font descriptions and installation instructions. (You may also find comments in the Makefile of interest.)

These packages were prepared by Doug Brenner (dbrenner@umaxc.weeg.uiowa.edu).

---

28. How much does the NeXT cost?

A retail price list may be obtained by contacting NeXT at the number or address mentioned in question 1. Educational prices vary depending upon the university. The typical educational discount is between twenty and thirty percent. NeXT offers registered developers a significant discount.

---

29. Where can I buy a NeXT?

From NeXT, Businessland, Computer Attic (in California), and possibly from your university if you are a student. If your university does not offer the NeXT, you may be able to buy from a "hub" university if the hub program is available in your state (contact NeXT for more info).

---

30. How can I save my WriteNow (or other printable) documents to a Postscript file?

It's easy. Just select PRINT from the main WriteNow menu, then select SAVE from the resulting print panel.

NeXTPRESS

(continued from page 8, column 3)

COMPUTER LANGUAGE PRODUCTIVITY AWARD

On February 12, 1991, NeXT went to an award ceremony put on by Computer Languages Magazine at the International Software Developers' Conference in Santa Clara.

The award category was "Integrated Development Environments." Other categories included such things as compilers, debuggers, CASE tools and "Special".

The nominees in the Integrated Development Environments category were:
Smalltalk V, ParcPlace Systems
MacApp, Apple Computer
Saber-C, Saber Software
NeXTstep 2.0 (Extended), NeXT
... and the winner was ...
NeXTstep 2.0!

X UPDATE

from: Dany Guindi
pensoft!dany@cs.utexas.edu
Pencom Software
512-343-1111
9050 Capitol of Texas Highway
North, Suite 300
Austin, TX 78759

A lot of people have asked the status of co-Xist™ for the colour machine, and I finally have something I can say... NeXT was kind enough to lend us a NeXTstation colour to try it out, we took our binaries and executed them on the colour machine and had a 2 bit greyscale X server running inmediately. The NeXTstation colour was also quite fast!

Needless to say, now we go into heavy development to get colour X before we have to return the loaner!

CREATING YOUR OWN FOLDER ICONS

One neat thing a user mentioned at the January meeting is that you can create your own icons for folders (i.e. directories) and have them show up in the File Viewer. Launch the Icon app (in /NextDeveloper/Demos), open up an existing application icon (e.g. /NextDeveloper/Examples /BusyBox/BusyBoxApp.tiff), change it to whatever you like using Icon's tools and save it as the file *.dir.tiff* in the folder to be identified by the icon.

You can also create an "open" icon for that folder (which appears when the folder "opens" to accept a file you drag into it). To set up an "open" icon, go through the same process as above except save it as the file *.opendir.tiff* in the folder.

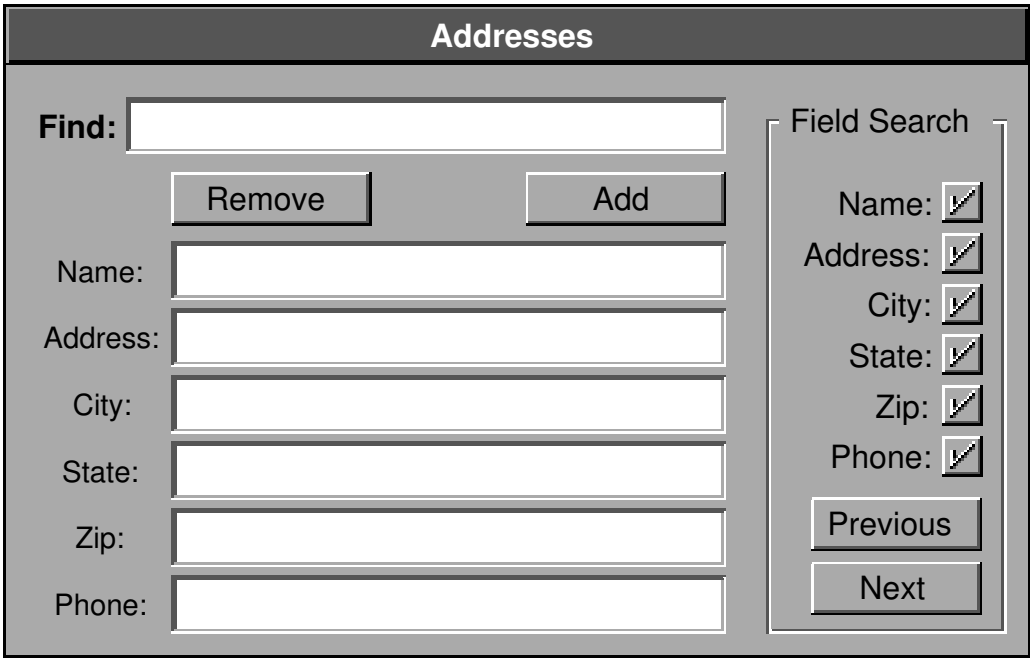
THE DATABASER

(continued from page 13, column 3)

LINELEN (macro substitution doesn't work in quote marks).

Then there is a function which returns a string. This is the name of the file to use when the Databaser loads and saves the data. If there is a '%s' in the string it is replaced with the users home directory. So to put your addresses in your home directory in a file called BlackBook, you would put the string "%s/BlackBook" in the function instead of the string which is there. Save the file. Go back to IB and Make the Application. You should have a functioning database!

If you use this class, please give me credit in the Info... Box. Thanks...



ASSEMBLED ADDRESSES WINDOW

DATABASER.H

```
/* Copyright 1989, 1990, 1991 Robert Patrick Thille */

/* Generated by Interface Builder */

#import <objc/Object.h>

@interface Databaser:Object
{
    id   keyForm;
    id   nextButton;
    id   dataForm;
    id   removeButton;
    id   addButton;
    id   previousButton;
    id   dataStorage;
    int   numrec;
    int   currec;
```

(continued on next page)

DATABASER.H

(continued from previous page)

```
  int   fields;
  char *datafilepath;
}

- setKeyForm:anObject;[1]
- setNextButton:anObject;
- setDataForm:anObject;
- setRemoveButton:anObject;
- setAddButton:anObject;
- setPreviousButton:anObject;

- setDataStorage:anObject;
- updatedataStorage;

- terminate:sender;
- appDidInit:sender;

- initbase;

- add:sender;
- remove:sender;

- previous:sender;
- next:sender;

- setFields:sender;
- (int)getFields;
- prevByField:sender;
- nextByField:sender;

- search:(BOOL)forward;
- enable;
- show:(int)record;
- (char *)stringFromStream:(NXStream *)fileStream
  ok:(int *)success :(char *)stringBuffer;

- search:(BOOL)forward inFields:(int)fields;
- getFields:(int)fields in:(STR)record:(char *)retstr;

@end
```

DATABASER.M

```
/* Copyright 1989, 1990, 1991 Robert Patrick Thille */
/* Generated by Interface Builder */

#import "Databaser.h"
#import <appkit/Application.h>
```

(continued on next page)

1. Editors' Note: The italicized methods are outlet initialization methods and may be omitted in NeXTstep 2.0.

DATABASER.M

(continued from previous page)

```
#import <appkit/Panel.h>
#import <appkit/Form.h>
#import <appkit/Matrix.h>
#import <appkit/Cell.h>
#import <appkit/publicWraps.h>
#import <objc/Storage.h>
#import <libc.h>
#import <sys/fcntl.h>
#import <ctype.h>

@implementation Databaser

#define ALLFIELDS 0xFFFFFF  // only good for 24 fields

#import "DBParams.h"

BOOL keyinstring(char key[], char string[])
{
   int i = 0, j = 0;
   char c, *temp;

   if (string == NULL)
      return NO;

   // convert the key to lower case;
   temp = key;
   while(*temp != '\0') {
      if (isupper(*temp))
         *temp = tolower(*temp);
      temp++;
   }

   while (string[i] != '\0' && key[j] != '\0') {
      c = isupper(string[i]) ? tolower(string[i]) : string[i];
      if (c == key[j]) {
         i++;
         j++;
      }
      else {
         i = i - j + 1;
         j = 0;
      }
   }
   if (key[j] == '\0')
      return YES;
   else
      return NO;
}
```

(continued on next page)

DATABASER.M

(continued from previous page)

```
void getfield(char field[], STR fields, int num)
// this gets the field number 'num' from 'fields' and puts it in 'field'
{
   int index;

   while (num > 0 && *fields != '\0') {
      if (*fields == '\t')
         num--;
      fields++;
   }
   index = 0;
   while (*fields != '\0' && *fields != '\t' && *fields != '\n'
        && index < MAXLINELEN) {
      *field++ = *fields++;
      index++;
   }
   *field = '\0';
}

- (char *)stringFromStream:(NXStream *)fileStream
      ok:(int *)success :(char *)stringBuffer
/* reads a newline-terminated string from a stream */
{
   int    nextChar, characterNumber = 0;

   while ((nextChar = NXGetc(fileStream)) != '\n') {
     /* make sure we haven't reached the end of the file */
     if (nextChar == EOF) {
      *success = 0;
      stringBuffer[characterNumber] = '\0';
      return stringBuffer;
     }
     stringBuffer[characterNumber++] = (char)nextChar;
   }
   stringBuffer[characterNumber] = '\0';
   *success = 1;

   /* create space for the string, copy it, and return the new string */
   return stringBuffer;
}

- updatedataStorage
{
   // save the current record before we quit or move to another record.
   // fill buf[] with the fields of the data form separated by '\t'
   int index;
   STR temp;
   char buf[MAXLINELEN];
```

(continued on next page)

DATABASER.M

(continued from previous page)

```
   if (!numrec)
      return;
   temp = buf;
   for(index = 0; index < NUMFIELDS; index++) {
      strcpy(temp,[dataForm stringValueAt:index]);
      while(*temp++ != '\0');
      *(temp-1) = '\t';
   }
   *(temp-1) = '\0';
   [dataStorage removeAt:currec];
   numrec--;
   con
   {
      int left = 0, right = numrec - 1;

      while(left <= right) {
         index = (left + right) / 2;
         if (strcmp(buf, [dataStorage elementAt:index]) < 0)
            right = index -1;
         else
            left = index+1;
      }
      [dataStorage insert:buf at:left];
      currec = left;
      numrec++;
   }
}
- initbase
// this method reads in the data from the file returned from getfilename()
// which should be declared external to this class
{
   const char  *homedir = NXHomeDirectory();  // users home directory
   int  scanresult, fd, success = 1;
   char stringBuffer[1024];
   NXStream  *filestream;

   numrec = 0;
   // setup datafilepath with the full path to the data file
   datafilepath = (char *)malloc(strlen(homedir)+strlen(getfilename()));
   sprintf(datafilepath, getfilename(), homedir);

   // open the data file and attach it to a stream
   fd = open(datafilepath, O_RDONLY);
   if (fd < 0) {// file is not there, or cannot be read
      dataStorage = [Storage newCount:0 elementSize:MAXLINELEN
        description:DESC];
      [self enable];
      return self;
   }
   filestream = NXOpenFile(fd, NX_READONLY);
```

(continued on next page)

DATABASER.M

(continued from previous page)

```
   // Create a new storage object, and read records from the file
   dataStorage = [Storage newCount:0 elementSize:MAXLINELEN
      description:DESC];
   while(success) {
      [self stringFromStream:filestream ok: &success :stringBuffer];
      if (success) {
         [dataStorage addElement:(void *)stringBuffer];
         numrec++;
      }
   }
   NXClose(filestream);
   close(fd);
   fields = ALLFIELDS;
   [self show:0];
   [self enable];
   [[keyForm window] makeKeyAndOrderFront:self];
   [keyForm selectTextAt:0];

    return self;
}
- setKeyForm:anObject[1]
{
    keyForm = anObject;
    return self;
}

- setNextButton:anObject
{
    nextButton = anObject;
    return self;
}

- setDataForm:anObject
{
    dataForm = anObject;
    return self;
}

- setRemoveButton:anObject
{
    removeButton = anObject;
    return self;
}

- setAddButton:anObject
{
    addButton = anObject;
    return self;
}
```

(continued on next page)

1. Editors' Note: The italicized methods are outlet initialization methods and may be omitted in NeXTstep 2.0.

DATABASER.M

(continued from previous page)

```
- setPreviousButton:anObject
{
    previousButton = anObject;
    return self;
}

- setDataStorage:anObject
{
    dataStorage = anObject;
    return self;
}

- terminate:sender
{
   int  scanresult, fd, counter, success = 1;
   NXStream  *filestream;
   struct stat stbuf;

   [self updatedataStorage];
   // rewrite the file, to set its length to zero
   fd = creat(datafilepath, PERMS);

   // open the data file and attach it to a stream
   if (fd >= 0) {    // file opened successfully
      filestream = NXOpenFile(fd, NX_WRITEONLY);
      for(counter = 0; counter < numrec; counter++)
         NXPrintf(filestream, "%s\n", [dataStorage elementAt:counter]);
      NXClose(filestream);
      close(fd);
   }

   [NXApp terminate:self];
    return self;
}

- appDidInit:sender
{
   [self initbase];
}


- add:sender
// this adds a blank record to the database
{
   int index;
   STR temp;
   char buf[MAXLINELEN];

   [self updatedataStorage];
```

(continued on next page)

DATABASER.M

(continued from previous page)

```
   // clear the dataForm for fresh input, select the top field and add a
   // blank record to the dataStorage
   for(index = 0; index < NUMFIELDS; index++)
      [dataForm setStringValue:"" at:index];
   [dataStorage insert:"" at:++currec];
   numrec++;
   [self enable];
   [dataForm selectText:sender];

    return self;
}

- remove:sender
// this removes the current record from the database.
{
   [dataStorage removeAt:currec--];
   numrec--;
   [self enable];
   if(currec < 0)
      currec = 0;
   [self show:currec];
    return self;
}

- previous:sender
// this searches backwards for the next record with the search criteria
{
   [self search:NO];
   return self;
}

- next:sender
// this searches for the next record with the search criteria
{
   [self search:YES];
   return self;
}

- search:(BOOL)forward
{
   [self search:forward inFields:ALLFIELDS];
}

- prevByField:sender
{
   [self search:NO inFields:[self getFields]];
}
```

(continued on next page)

DATABASER.M

(continued from previous page)

```
- nextByField:sender
{
   [self search:YES inFields:[self getFields]];
}
- getFields:(int)fieldmap in:(STR)record:(char*)retstr
// uses fields as a bitmap and AND's out fields in record and returns in retstr
{
   int index, bit;
   char temp[MAXLINELEN];

   retstr[0] = '\0';
   for(index =0, bit =1; index < NUMFIELDS; index++, bit *=2)
      if ( fieldmap & bit) {
         getfield(temp,record,index);
         strcat(retstr,temp);
      }
}

- search:(BOOL)forward inFields:(int)fieldmap
{
   char  tempstr[MAXLINELEN], tempstr3[MAXLINELEN], *tempstr2 = tempstr;
   int index, offset;
   BOOL done = NO;
   BOOL invert = NO;
   BOOL found;


   [self updatedataStorage];

   if (forward)
      offset = 1;
   else
      offset = numrec-1;
   strcpy(tempstr, [keyForm stringValueAt:0]);

   if (tempstr[0] == '!') {
      tempstr2++;
      invert = YES;
   }

   index = currec;

   while(!done) {
      index=(index+offset) % numrec;
      [self getFields:fieldmap
         in:(STR)[dataStorage elementAt:index]:tempstr3];
      found = keyinstring(tempstr2,tempstr3);
      done = (invert ? !found : found ) || (index == currec);
   }
```

(continued on next page)

DATABASER.M

(continued from previous page)

```
   if (index == currec) {
      [[keyForm setStringValue:"TEXT NOT FOUND" at:0] selectTextAt:0];
   }
   else {
      [self show:index];
   }
     return self;
}

- setFields:sender
{
   int tag, temp, inc;
   id cell;

   for(tag=temp=0, inc=1; cell=[sender findCellWithTag:tag]; tag++, inc*=2) {
      temp += inc * [cell intValue];
   }
   fields = temp;
}

- (int)getFields
{
   return fields;
}

- enable
{
   switch(numrec) {
   case 0:    // Don't let the user do anything but add a record
      [previousButton setEnabled:NO];
      [nextButton setEnabled:NO];
      [removeButton setEnabled:NO];
      [keyForm setEnabled:NO];
      [dataForm setEnabled:NO];
      break;
   case 1:    // Dont let the user search if there is only one record
      [previousButton setEnabled:NO];
      [nextButton setEnabled:NO];
      [removeButton setEnabled:YES];
      [keyForm setEnabled:NO];
      [dataForm setEnabled:YES];
      break;
   default:   // The user can search, add, or remove records.
      [previousButton setEnabled:YES];
      [nextButton setEnabled:YES];
      [removeButton setEnabled:YES];
      [keyForm setEnabled:YES];
      [dataForm setEnabled:YES];
      break;
   }
```

(continued on next page)

DATABASER.M

(continued from previous page)

```
   return self;
}
- show:(int)record
// this shows the record number record
{
   STR tempstr;
   char temp[MAXLINELEN];
   int index;

   if (!numrec) {
      for(index = 0; index < NUMFIELDS; index++)
         [dataForm setStringValue:"" at:index];
      [self enable];
      return self;
   }
   tempstr = (STR)[dataStorage elementAt:record];
   for(index =0; index < NUMFIELDS; index++) {
      getfield(temp, tempstr,index);
      [dataForm setStringValue:temp at:index];
   }
   currec = record;
   [keyForm selectTextAt:0];
   return self;
}
@end
```


SCaNeWS CONTRIBUTORS

Mark Adler
Henry Chiu
Conrad Geiger
Mike Mahoney
Robert Thille
off the net:
Pascal Chesnais
John Garnett
Erik Kay
Editors:
Mike Mahoney & Lorraine Rapp

If you would like to submit an article to *SCaNeWS*, contact the editors.

SCaNeWS is composed using FrameMaker™

Articles are copyrighted by their authors.

SCaN OFFICERS

PRESIDENT: Michael K. Mahoney
Computer Engineering/Science Dept
CSULB, Long Beach, CA 90840
(213) 985-1550
mahoney@beach.csulb.edu

VICE-PRESIDENT: Bob Desharnais
(213) 343-2056
KCBQRAD@CALSTATE.BITNET

SECRETARY: Lorraine Rapp
rapp@beach.csulb.edu

TREASURER: Henry Chiu
chiu@beach.csulb.edu

If you have any comments or questions about SCaN or *SCaNeWS*, contact one of the above or scan@beach.csulb.edu