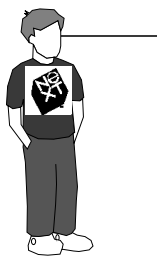


SCaNeWS

Southern California NeXT™ Users Group Newsletter

VOLUME 1, NUMBER 6
AUGUST 1991

SCaNeWS ©1991
All Rights Reserved

An August Performance

august, *adj*. - inspiring reverance or admiration; grandeur; majestic.

That's what happens to so many NeXT users, they are inspired to revere those black cubes, slabs and displays (and especially for me) the intelligence that makes that hardware do grand and majestic things.

August, *n*. - 1. the eighth month of the year;
2. the Illustrator demo month at SCaN

Yes SCaNners, we've got a great one for you this month - a NeXTWORLD 5-cuber, a product that every reviewer raves about (see page 4 for more).

SCaN is on a roll. The past two meetings have been great successes. The May 7 meeting at UCLA was attended by about 70 people and featured a demo of Visus' PaperSight and some internal NeXT apps on a NeXTdimension (ND) system. Our sincere thanx go to Allen Denison and Chet Kapoor for doing so much to make the meeting successful. The June 4 meeting at CSULB featured Steve Jobs' great 35 minute ND demo on video and a hands-on networking meeting where about 90 people got to play with the rows of networked NeXTs in our lab. There were so many people that the video was shown twice to packed rooms (which also took the pressure off the NeXT usage). The focus was on networking software such as NeXTmail, co-Xist (X Windows with Motif), Touch (a NeXTstep interface to ftp), and NewsGrazer (a NeXTstep interface to Usenet news). People seemed to enjoy the hands-on feature of the meeting so much that we'd like to plan another one at Long Beach later this year. If you stick around late enough you'll see hacks and developers do some pretty crazy things. - *Mike Mahoney*

NeXT at SIGGRAPH '91

Mike Mahoney
mahoney@csulb.edu

I just got back from the SIGGRAPH '91 conference which was held in Las Vegas the week of 7/30/91. SIGGRAPH is the annual ACM extravaganza held in various places around the US and is certainly the premier annual graphics event in the world. For example, graphics techniques such as compositing (transparency) and radiosity were presented as research papers at SIGGRAPH '84 and Pixar's Oscar winning "Tin Toy" was shown at SIGGRAPH '88. About 30,000 people attended SIGGRAPH '91 which featured tutorials, panels, refereed papers,
(continued on page 18, column 2)

CONTENTS

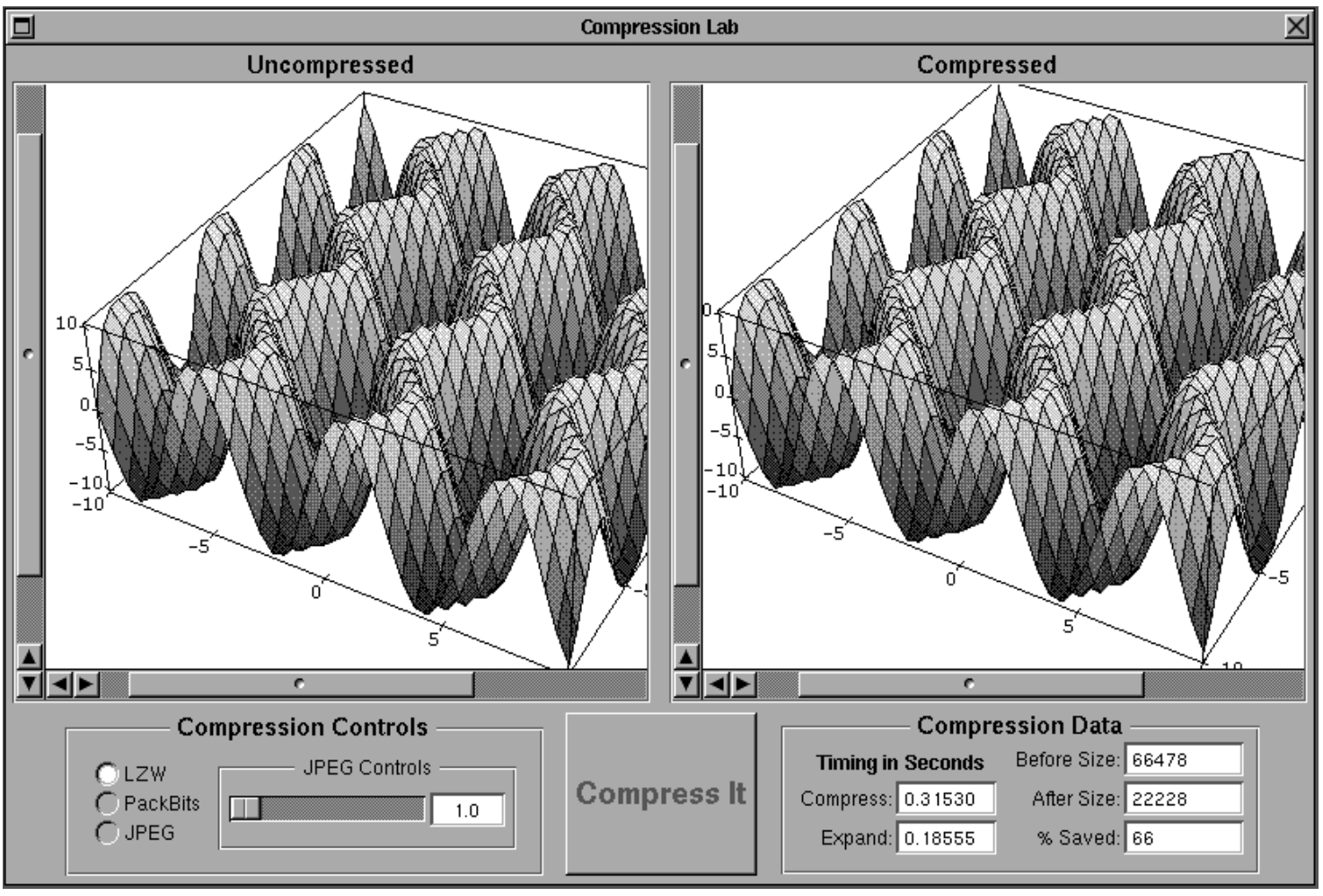
NeXT at SIGGRAPH '91 1
Image Compression 2
NeXTPress
Adobe Illustrator 3.0 4
NeXT Technical Alert 6
June NextAnswers 7
Comp.sys.next &Digital Librarian . . . 7
New NeXT MiniExamples 8
New IB Tutorial from NeXT 9
NeXT Publications. 10
NeXTstep Advantage 10
NeXTware 11
State of the NeXT Economy 16
New Mathematica Journal 17

FUN WITH IMAGE COMPRESSION

by Robert Thille (robert%nextoid@hub.ucsb.edu)
14053 Foothill Road, Santa Paula, CA 93060

This isn't really an article about the technical aspects of image compression, but rather about how to go about using it in your programs or as a user on the NeXT platform. Also, to get in-depth, you really have to dig in and take a look at the code.

At the last SCaN meeting, an engineer from NeXT demonstrated an application which let the user read in a TIFF file and play with the different image compression options built into the standard NeXT system software. You could view the result in a pane next to the original image, see how much time it took to compress it and decompress it, and how much less space the compressed version took up. I thought this would be a great tool/toy, since Icon will let you compress things with JPEG compression, but you can't tell ahead of time what they will look like. The guy from NeXT said that they wouldn't distribute the application because it wasn't up to release standards. I thought this was pretty odd considering how many times Icon has crashed on me,
(continued on next page)



CompressionLab Interface

Image Compression

(continued from previous page)

but I do understand them not wanting to have to be responsible or support "playware". So I decided to write it myself. The ultimate reinventing of the wheel.

I started out just in Interface Builder, trying to remember what the NeXT version's interface looked like and decide whether I wanted to do it the same or not. I decided to do a similar interface, using panes with scrolling views onto a dot-for-dot image of the TIFF file. By dot-for-dot, I mean that if the TIFF file came from a scanner and it was 3" by 5" at 144 dots per inch, on the screen it would be about 6" by 10", because the screen is 72 dots per inch. It isn't really, but the software pretends that it is, so things come out WYSIWYG-only a little smaller on screen.

Once I had defined my interface, I had to start building the components that didn't already exist. There were three main parts, objective-C classes, two of which might be useful in another application. The first one is CLController, which handles the messages from the interface objects, and decides what to do with them, passing them to the other parts or taking care of them itself.

The next part was NXImageView. It's something of a misnomer right now, because I really only handle bitmapped (TIFF) images in the view. It's an easy way to put a NXBitmapImageRep in a scrolling view, but it's slow to scroll and it redraws some things too much when the file is changed. It also handles reading and writing the image via the NXBitmapImageRep's existing methods.

The final part was a Timer class. It is just an object oriented interface to the MACH gettimeofday call. It keeps track of when it was started and then when you stop it, it figures out the elapsed time. It seems to have a great deal of precision, though I don't have any idea how accurate it really is. Feel free to use the class in your own programs, and the same with NXImageView. You should probably clean NXImageView up a little, if I don't beat you to it. :-)

Here's the break down on the division of work between the three main parts. The user interface, which was defined in IB, takes commands from the user and passes them to the CLController. The CLController either adjusts some of the other controls in the interface (if you changed the compression parameters), enables or disables controls (if you loaded or saved a file, or changed compression parameters), or tells the NXImageView to preform some actions. The NXImageView's job is just two main things: read and write TIFF files, and display them on the screen.

There are still lots of things that can be done with this program, but it's a starting point for learning a little about the NeXT's image compression utilities. Some enhancements I've come up with and some that Karl Kraft, my beta tester, have come up with are:

- Make it a service available in all applications
- Allow users to copy and paste images
- Give more explanation of JPEG
- Warn users if they open a previously compressed image
- Clean up the interface
- Use disk instead of memory for the compression and decompression, (Karl)
- Use multiple windows instead of a single output pane so users can compare multiple compressed images
- Allow users to add compression modules dynamically so that non-NeXT supported compression could be added (a candidate for this is LZW with horizontal differencing, this is available in Photoshop on the Mac and NeXT will read it, but not write it)
- Add other image manipulations besides just compression, ie. allow the user to adjust the histogram of the channels of the image like Scene and Icon do.

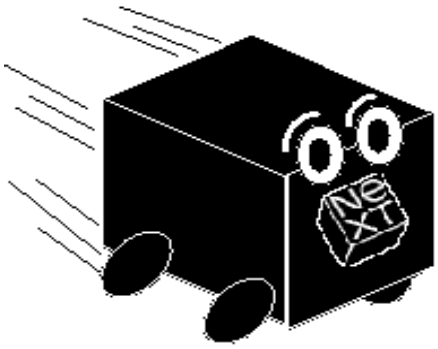
[Source code listings begin on page 20]

Next SCaN Meeting

The next SCaN meeting will be held a little early to avoid the beginning-of-the-semester crunch.

Date: Thursday, August 29
Time: 6:30 p.m.
Place: California State University, Long Beach

- A representative from Frame Technology will demonstrate FrameMaker 3.0 on the NeXT. Release 3.0 of FrameMaker will be generally available in September 1991.
- A representative from Uninet will demonstrate a SCSI expansion box



NeXT Press

Announcements from NeXT and 3rd party product suppliers about product releases, upgrades, bugs, and other matters of importance...

Adobe Illustrator 3.0 for NeXTstep is now available!

MOUNTAIN VIEW, Calif
(July 1, 1991)

Adobe Systems Incorporated today announced the shipping of Adobe Illustrator for NeXT software. The new program offers unprecedented flexibility in desktop color image manipulation, including the ability to perform real-time edits inpreview mode. Using Adobe's Postscript software technology resident on the NeXT computer, Adobe Illustrator software allows users to see colors and placement on screen exactly as they will appear in finished, printed format.

The product has a suggested retail price of $695 and is available from all authorized dealers.

"We designed this version of Adobe Illustrator to take full advantage for the Display PostScript system and the NeXT computer's extraordinary color and file handling capabilities," said Dave Pratt, vice president and general manager of Adobe's Application Products Division. "Now, artists, designers and business professionals can quickly and easily create powerful artwork, incorporating graphics and images for maximum impact."

"Graphic artists are going to go crazy for this product," said Steven P. Jobs, president and CEO of NeXT Computer, Inc. "Moving the color windows around is soooo fast, and now you can even edit in Preview mode!"

Adobe Illustrator for NeXT software offers all of the drawing, tracing, blending and other tools found in the Macintosh version of the product. The Display PostScript system and advanced color capabilities included in the NeXT computers support additional features that are available only in the NeXT version of Adobe Illustrator software; some of these are summarized below:

• Editing is available in all display models, including preview illustration and preview selection. This allows the user to evaluate instantly, and in full color, the effects of changes on the finished artwork.

• Industry standard Tag Image File Format (TIFF) files as well as Encapsulated PostScript (EPS) files can be imported, "dragged and dropped", or placed as art.

• Color or grayscale TIFF images may also be used as templates for tracing with the Autotrace tool, and each shade and hue in the color image can be traced independently.

• User-definable grids with independent horizontal and vertical snap-to-point may be overlaid on the screen to aid in precision placement of artwork.

• NeXTMail service allows TIFF, EPS or Adobe Illustrator files to be transferred to other NeXT computers on a network.

In addition to the program disk and user guide, the product package includes a tutorial guide and disk, a guide to advanced tips and techniques, the Adobe Separator program for color separations, and a disk of sample art. Adobe offers free technical support to registered users.

System Requirements
The minimum system requirements for Adobe Illustrator for NeXT software are a NeXT com puter, 1.44 MB floppy disk drive,
(continued on next page)

NeXTPress

(continued from previous page)

NeXTstep system software 2.0 or greater and a PostScript printer of other output device. For best performance and results, a NeXT system with 68040 processor, color monitor and color scanner are recommended.

Adobe systems Incorporated (NASDAQ symbol:ADBE), founded in 1982, is headquartered in Mountain View, California. Adobe is the developer of PostScript, the industry standard page-description language for the electronic printing and publishing fields. In addition, the company develops and markets a complete line of application software programs, including the award-winning Adobe Illustrator, Adobe Type Manager and Adobe Type Library. Revenue for fiscal 1990 exceeded $168 million.

Adobe PostScript, Display PostScript and Adobe Illustrotor are registered trademarks of Adobe Systems, Incorporated. Macintosh is a registered trademark of Apple Computer, Inc. NeXT and NeXTstep are trademarks of NeXT Computer, Inc.

More Illustrator News...

The following article is excerpted from PC Week 7/8/91:

Illustrator Draws on NeXT System's Strengths

By Christopher Stetson

Adobe Systems Inc.'s $694 Illustrator for NeXT exploits the powerful NeXT environment, proving itself a much quicker, richer drawing package than its Mac sibling.

The superior speed of Illustrator for NeXT's drawing tools alone could entice graphic artists to make the switch to the NeXT platform. Illustrator for NeXT, which was release last week, also capitalizes on the multitasking capabilities of Unix and the graphics power of Adobe's Display PostScript, but it has a few sluggish tools.

Graphic artists familiar with Adobe Illustrator 3.0 for the Macintosh will find Illustrator for NeXT a snap to work with: Menus, keyboard shortcuts and dialog boxes remain consistent between the two applications.

However, the NeXT interface provides productivity-enhancing features that are not available on the Macintosh. Choosing colors using the NeXT color palette is a boon to users who prefer selecting colors visually, rather than specifying them in cyan, yellow, magenta, black (CMYK) percentages.

When users press the alternate button on the NeXT mouse, pop-up menus appear under the cursor, providing quicker access to all menu items.

One major feature of Illustrator for NeXT that will make Mac users green with envy is the capability to edit in preview mode. The Motorola Inc. 68040 CPU and Display PostScript on the NeXTstation allow users to edit rendered images, eliminating the need to jump back and forth between view modes.

On the Macintosh, users must draw in an edit mode that uses outlines to represent graphics elements within the drawing. To see color and layer relationships, users must switch to the preview mode, then wait while Illustrator renders the PostScript image. To make any changes, the artist must move back into the edit mode.

Another benefit of Display PostScript is that it stores Illustrator windows in a memory cache, eliminating the many redraws artists must endure on the Mac. Additionally, Display PostScript shows imported graphics in full color. This provides WYSIWYG display rather than the Mac's display of a nondescript box.

In addition, Illustrator for NeXT takes advantage of the multitasking capabilities of the NeXTstation's Unix-based operating system. Users can switch to another application at any time without system hesitation. Background processes also have very little effect on the performance of Illustrator for NeXT. Within the application, some processes are multithreaded a type of multitasking within an application so that graphics can redraw in one window while editing occurs in another.

However, although Illustrator brings a powerful drawing program to the NeXT station and the system's hardware and software components flesh out some of Illustrator's best features, other elements seem unnecessarily slow. For example, using the grabber-hand to scroll images is too slow to be effective, and delays occur during editing for no apparent reason.

(continued on next page)

NeXTPress

(continued from previous page)

NeXT Technical Alert: Networking Bug

NeXT Technical Alert 91-3

Summary

A loop in a NetInfod database can be erroneously created, and deleting this loop will destroy the information in the database. You may create a loop by destroying erroneous duplicate directories.

Description

Duplicate directories can be created in a NetInfo domain during normal operations under various circumstances. For example, a subdirectory of /printers may appear in /machines. This can occur while using high-level manager applications (such as PrintManager) as well as using other tools.

This duplicate entry is benign unless it's deleted. If this duplicate entry is deleted, a loop forms in the database. If you're using NetInfoManager, this loop appears as dir:0 in a directory. This loop leads to NetInfo failures and long delays in retrieving information from NetInfo. If the loop is deleted, the NetInfo database is destroyed and you'll have to retrieve the information from a backup copy.

Workaround

The workaround to this problem is to remove the checksum file created by netinfod when it shuts down in a controlled fashion. The checksum file (there's one per NetInfo database) is written in the .nidb directory; its presence tells NetInfo to assume that the database is consistent.

Warning: This workaround is only a preventative measure. It won't correct existing loops. If you have a NetInfo database with a loop, the database can only be recovered from backups or rebuilt from scratch.

This workaround should be applied on each machine that is the master NetInfo server for a domain. However, you don't have to use the workaround on machines that serve only a local domainPeven though a loop can appear in any NetInfo domain (network-wide or otherwise), it's rare to see one in a machine's local domain. Even though it's not a problem to implement this workaround on any machine, the effort might not be worthwhile on machines that don't serve a network-wide domain.

Example

Let's assume you have a network with three machines: master (the master NetInfo server of the root domain), clone (a clone server for the root domain), and client. The following table describes the domain each machine serves, and in what capacity. Also assume that this is a two-level hierarchy.

Machine	**Domain Served**	**Type of Server**
master	/	Master
master	/master	Master
clone	/	Clone
clone	/clone	Master
client	/client	Master

In this configuration, we recommend applying the workaround to master, assuming that the local domains for each machine won't be modified very much. It's not necessary to apply the workaround to clonePeven though it serves the root domain, it's a clone server of the domain and gets its authoritative information from master.

To implement the workaround, edit the file /etc/rc. Look for the following lines in the file (the first line shown is number 108 in the standard /etc/rc file):

```
# Start up the netinfo daemon
if [ -f /usr/etc/nibindd ]; then
   /usr/etc/nibindd && (echo -n ' netinfo')     >/dev/console 2>&1
fi
```

Add the line shown in bold:

```
# Start up the netinfo daemon
if [ -f /usr/etc/nibindd ]; then
   rm -f /private/etc/netinfo/*.nidb/checksum
   /usr/etc/nibindd && (echo -n ' netinfo')     >/dev/console 2>&1
fi
```

(continued on next page)

NeXTPress

(continued from previous page)

Note: To perform this procedure, you must be logged in as root, or you must su to root.

After you edit the /etc/rc file, it's not necessary to reboot the machine. The next time the machine is shut down or rebooted, the workaround will take effect.

For More Information
Contact your NeXT Authorized Support Provider for more information about this problem.

June NextAnswers Available

The June release of NeXTanswers is now available on the Internet archive servers. NeXTanswers is a collection of answers to commonly asked technical questions about NeXT computers. The topics covered include system administration, programming, and NeXT applications.

When incorporated into Digital Librarian, NeXTanswers supplements the existing on-line and printed documentation, further helping you to understand and apply the power of NeXT computers.

We recommend that you obtain the latest release of NeXTanswers and have it installed on the NeXT computers at your site. You can obtain NeXTanswers via anonymous ftp (file transfer protocol) from one of the following Internet archive servers:

Hostname: sonata.cc.purdue.edu
IP Address: 128.210.15.30
Directory: pub/next/docs

Hostname: cs.orst.edu
IP Address: 128.193.32.1
Directory: pub/next/documents/NeXTanswers

There is an ftp site in Japan for our Asian customers:

Hostname: etlport.etl.go.jp
IP Address: 192.31.197.99
Directory: pub/NeXT/documents/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
path name@site.edu
or
path wellknownsite!yoursite!yourname@uunet.uu.net

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

Lastly, NeXTanswers is now available by annual subscription on floppy disk (part number E1036). For ordering information, contact your local NeXT sales representative or call 800-848-NeXT.

There are two versions of NeXTanswers: a complete set, and a monthly supplement for those who already have the previous month's complete set. The former is called NeXTanswers_Jun_91.tar.Z and occupies 568K of disk space (1649K when installed). The latter is called NA_Supplement_Jun_91.tar.Z, occupies 194K (526K when installed), and contains files that are new or changed since the March 91 release.

To obtain NeXTanswers from an archive server, su to root in a shell and type the commands list below. Replace "filename" with "NeXTanswers_Jun_91.tar.Z" or "NA_Supplement_Jun_91.tar.Z", depending on whether you want the complete set or the supplement.

The documentation (INSTALL.wn, README.wn, and CHANGES) is included in a separate file called NeXTanswers_docs_June_91.tar.Z. This file can go into the same directory where you place NeXTanswers_June_91.tar.Z.

Comp.sys.next Postings for Digital Librarian

Many times I wished that I could read the old csn stuff without wading through all the crud. Many times I wished that someone would re-arrange this information so that it was useful, or at least more accessible to Librarian.

Well, I did it.

I have uploaded three files to sonata (sonata.cc.purdue.edu)

1989.squfold
1990.squfold
1991.squfold.

As well, I've uploaded UnSquashOnly, which you will need to expand these archives.

(continued on next page)

NeXTPress

(continued from previous page)

These are squashed folders. Within each folder is the entire year's comp sys next, one thread per file. A thread was delimited by subject line. E.g. if someone changed the Subject line, a new thread started. (Re:'s didn't count)

I also stripped out the signature files, or at least most of them. Any line that was duplicated exactly was removed. (The first copy was left, subsequent copies deleted.) This may break code fragments. Most headers other than Date, From, Subject, and Keywords were also stripped out.

Each file has for a name csn_<-subject line with spaces replaced with _>

If you don't have it, you will also need to download UnSquashOnly to uncompress these archives.

Note: 1991 in particular is big. Uncompressed is over 10 megs in something like 2500 files. Unpack this where you want it. Next gets it's shorts in a knot over moving thousands of files at a time.

1991 is current through April. I will post the next update for it in early July. Comments & suggestions appreciated.

New NeXT MiniExamples

The July release of MiniExamples is now available on the Internet archive servers. It contains 12 small programming examples from NeXT Developer Support.

• *AtYourService* demonstrates how applications can export *Services: to other applications (they will appear in other applications' Services menu). This version supersedes the one released in March 91.

• *ImageText* demonstrates how to place graphic images in a Text object. It presents a window into which you can drag files from the Workspace. This version supersedes the one released in March 91.

• *AskMe* shows how to internationalize an application. It also contains some selected NextAnswers on various topics such as Technical Support, Archive Servers, etc.

• *BananaSplit* is a simple NXSplitView sizing example.

• *PaginationLab* demonstrates how you can influence the Appkit's pagination methods, and customize your pagination needs.

• *Rotato* illustrates the steps necessary in rotating an NXImage.

• *ScrollingText* shows how to create a document window which contains additional controls in the scroller bars.

• *ScrollTiff* demonstrates how to load a TIFF file into a ScrollView.

• *TIFFandEPS* show how to open EPS or TIFF images and save them back as EPS or TIFF - converting the image in the process.

• *WhatsUpDoc* provides a skeleton example on how you might manage multiple documents in an application.

• *Winfo* is a tool to provide information about window memory usage. It's useful for finding windows which unnecessarily allocate alpha or color planes and waste large amounts of memory.

• *ZooView* is another more specialized example on NXSplitView. It shows how to implement a SideSplitView. The SideSplitView allows a window to be divided into two sections sitting side by side.

The MiniExamples are available via anonymous ftp (file transfer protocol) from one of the following Internet archive servers:

Hostname: cs.orst.edu
Directory: pub/next/documents/NeXTanswers/MiniExamples

Hostname: sonata.cc.purdue.edu
Directory: pub/next/docs/MiniExamples (may still be in pub/next/submissions)

Hostname: etlport.etl.go.jp
Directory: pub/NeXT/documents/MiniExamples/July-91

Purdue also maintains an email-based 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 in your message like "path name@site.edu" or "path well_known_site!your_site!your_name@uunet.uu.net". Note that

(continued on next page)

NeXTPress

(continued from previous page)

neither name@site.bitnet nor name@site.UUCP will work.

The untarred MiniExamples and the README file occupy a total of 0.403 Megs.When uncompressed, they will occupy about 1.7 Megs. To retrieve the files from an ftp-based archive server, make sure you have enough free space on your disk, then su to root in a Terminal window and type the following commands (what you type is in italics).

mkdirs /LocalLibrary/MiniExamples
cd /LocalLibrary/MiniExamples
ftp hostname-or-IP-address [see table above]
[Login:] *anonymous*
[enter your own login name as the password]

ftp> *cd MiniExamples-directory*
[see table above for the name]
ftp> *binary*
ftp> *mget *.tar.Z*
[then type "*y*" for each file]
ftp> *ascii*
ftp> *get MiniExamples_README*
ftp> *quit*

For each MiniExample_month_year.tar.Z, do the following:

zcat MiniExample_month_year.tar.Z | tar -xvf -

This will create a MiniExample directory for each unarchived MiniExample. Finally, remove the archive file(s) after installation:

rm MiniExample_month_year.tar.Z
[if it asks for confirmation, type "*y*"]

Updated Interface Builder Tutorial from NeXT

The file NeXT_IB_Tutorial.tar.Z is available on the archive server nova.cc.purdue.edu (in the submissions directory) and contains a draft version of a tutorial for Interface Builder. This draft is based on the tutorial printed in chapter 8 of the "NeXTstep Concepts" manual. However, it has been updated to agree with release 2 of NeXTstep, and many new features have been added. Of special note are two new projects that describe how to build custom palettes and custom palette inspectors.

This draft contains only the Interface Builder tutorial; an updated version of the Interface Builder reference material will be released at a later time.

Contents

NeXT_IB_Tutorial.tar.Z is .369 megabytes in size. After you unarchive the contents, the tutorial occupies 1.98 megabytes and consists of these documents:

- *IntroToTutorial.rtfd/*
  General introduction and conceptual background
- *Project1.rtfd/*
  Build a trivial application named Simple
- *Project2.rtfd/*
  Build Calculator, converts degrees C to F
- *Project3.rtfd/*
  Modify Calculator to enhance its abilities
- *Project4.rtfd/*
  Build TextEditor and explore multiple nib files
- *Project5.rtfd/*
  Build ProgressView, a custom palette
- *Project6.rtfd/*
  Build ProgressViewInspector, an inspector

Your Help, Please

We in NeXT Publications have two goals in releasing drafts of documentation through archive sites. First, this method allows us to distribute information to you in a more timely manner, shortcutting some of the editorial and production time that would otherwise be required before release. (This implies, however, that these documents will be less polished than they would be in a normal release: we ask your indulgence.) Second, we hope to receive feedback that will help us improve the documentation before it's released in its final form.

If you have suggestions or bug reports relating to this tutorial, please send me email at matt@next.com. I can't guarantee that I'll respond to each message, but I'm interested in all comments

(continued on next page)

NeXTPress

(continued from previous page)

relating to this tutorial. If you have comments, suggestions, or bug reports about other documentation, Interface Builder, the Application Kit, or other aspects of NeXTstep, please direct them to bug_next@next.com.

NeXT Publications

NeXTWORLD Magazine

The third issue of NeXTWORLD magazine will be available on newsstands nationwide July 30th. You'll find it at chains such as B. Dalton's, Walden Books, and Crown Books. Articles in this issue include some of the following:

1. *Special report on NeXT Color*, including: color machines, color standards, alpha channels, color theory/color picker; Adobe Illustrator reviewed/Adobe Separator; The Ins and Outs of Color-Color peripherals for the NeXT; Tips/ Service bureaus, etc.

2. *Product Reviews:* Wordperfect, Frame, OCR Servant, Co-Xist, PLI

3. *Features:* NeXT on Wall Street, Improv Development Profile, and Photo Tour -- The NeXT Labs

4. *Vanishing Point:* Deja-Vu -- Guy Kawasaki takes a look at NeXT's advocate program and style of evangelism.

5. *How-to:* This section focuses on user needs, hints and tips, and system administrator needs.

As announced last month, the magazine will communicate more frequently with its subscribers by launching a monthly newsletter in the late-summer time frame. The magazine is now a quarterly publication and will continue to be mailed to subscribers and distributed on newsstands. The newsletter will be mailed to subscribers during the months in which the magazine is not published; in other months it will be slipped into subscribers' copies of the magazine. To subscribe call 1-800-745-7002 or write:

Subscriber Services,
P.O. Box 56429,
Boulder, CO 80322-6429.

If you are having trouble with a subscription call 1-800-755-6398.

The App Wrapper

The first issue of the "App Wrapper" will be mailed out by July 15. This month's issue features a short article by Daniel Miles Kehoe encouraging the NeXT community to get more visible. Look for the 36 pages of submissions by NeXT developers regarding their new fonts, clip art, software, hardware, furniture, T-shirts, newsletters, books and services. This mailing will be going out to the NeXTWORLD mailing list. The closing dates for submissions/ads for the next issue will be Friday, August 23rd.

If you are interested in more about the "App Wrapper" or would like to advertise your group or project, contact Peggy Thompson via phone at (206)448-0845, FAX - (206)448-2350 or email at 0004454438@mcimail.com.

NeXT on Campus

The Spring 1991 issue of NeXT on Campus is now available. If you're not a subscriber to NeXT on Campus, you may receive it free by sending your request to NeXT_on_campus@NeXT.com Please include the following information: Name, Title, Department, Institution/Organization, Street Address, City, State, Zip, E-mail Address.

NeXTstep Advantage Available on Archive Servers

A PostScript-format version of the "NeXTstep Advantage" brochure is now available on the following archive servers:

Hostname: cs.orst.edu
Directory: pub/next/documents/ NeXT/

Hostname: sonata.cc.purdue.edu
Directory pub/next/docs/ (still in submissions directory?)

Hostname: etlport.etl.go.jp
Directory: pub/NeXT/documents/ NeXTstepAdvantage/

The file name is NeXTstepAdvantage.tar.Z; (its compressed size is 1.3 megabytes; uncompressed, it's 9.5 megabytes). In addition to the PostScript files listed below, the tar file contains the complete source code for the brochure's Plotter application (including makefiles, .h, .m, nib, tiff, and other files).

On the servers there's also an ASCII file called "NeXTstepAd-

(continued on next page)

NeXTPress

(continued from previous pge)

vantage_README" which contains the following Table of Contents:

00_FrontMatter
Cover.ps
TableOfContents.ps
Intro.ps
1 THE NEXTSTEP ADVANTAGE
1 What is NeXTstep?
1 What's the NeXTstep Advantage?
3 A Recognized Advantage
3 Putting the NeXTstep Advantage to Work
3 A Look Ahead

01_StepByStep.ps
5 STEP-BY-STEP THROUGH A NEXTSTEP APPLICATION
6 Introduction to NeXTstep Programming
11 Step 1. Designing the Application
13 Step 2. Assembling the User Interface
24 Step 3. Defining Custom Classes with Interface Builder
27 Step 4. Connecting Objects in the Application
32 Step 5. Implementing the Custom Classes
39 Step 6. Making, Running, and Debugging the Application
43 Reusing Application Features
52 Summary

02_ObjectiveC.ps
53 OBJECT-ORIENTED PROGRAMMING
53 Objects
55 Classes
59 Using an Object-Oriented Kit

03_NeXTstep.ps
61 FUNDAMENTALS OF NEXTSTEP
61 The Window Server
62 The Application Kit
66 Event Handling
67 Drawing
69 Messages between Applications
71 Summary

04_Classes.ps
73 CLASS SUMMARIES
74 Windows
75 Text Handling
76 Controls
78 Panels, Menus, and Pop-up Lists
80 Organizational Views
81 Fonts
82 Support for Printing
83 File Management
84 Images
86 Support for Choosing Colors
87 Comprehensive List of Classes

05_Tools.ps
93 NEXT DEVELOPMENT TOOLS
93 Interface Builder
93 Editors
93 Compilers
94 Debugging and Performance Tools
95 Digital Librarian
96 Tools for Development on the Digital Signal Processor (DSP)
96 Tools for PostScript Development
96 The Mach Operating System

06_Standards.ps
97 STANDARD DATA FORMATS ON NEXT COMPUTERS
97 Tag Image File Format (TIFF)
97 The PostScript Language
97 Rich Text Format (RTF)

07_PlotterSources
99 PLOTTER SOURCE LISTINGS
CodeIntro.ps
Code.ps
SourceFiles/

NeXTware

Diagram 1.1 Demo

Diagram! 1.1 is now available. The demo is on:
nova.cc.purdue.edu
cs.orst.edu
If you already own Diagram!, use your current registration number to fully enable the demo, and update to 1.1 for free. Many thanks for your enthusiastic support!

Diagram! 1.1 includes a brand new collection of pretty and informative sample docs, a bunch of bug fixes, and compatibility with Digital Librarian. So now you can use DL to sift through Diagram! work.

(continued on next page)

NeXTPress

(continued from previous page)

For those of you who haven't heard, Diagram! is a fundamentally different way of looking at graphics. We at Lighthouse Design figured most of you didn't want another freehand illustration tool--the world has enough software for artists, and for those of us who don't spend our time creating magazine covers, it's not particularly useful.

Instead, we find that most of you create "information graphics"--representing concepts, ideas and relationships--basically, homework instead of artwork. So that's how we tuned Diagram!, for people who use graphics, and don't just create them. When cornered, we like to say "for people who think and draw at the same time."

Academics love Diagram! because they can use it for everything from database schema to quick and dirty page layout. Not to mention data-flows, flow charts, other CASE, visual programming languages, legal arguments, mathematic illustrations, decision trees, and on and on... In short, homework.

Diagram! costs $399 list. If you're a student, it's $25. For faculty or staff, it's $75. Why so cheap? Because there was a time not too long ago when most of us at Lighthouse not only had homework, we had academic budgets, too. And we don't want one to get in the way of the other.

And with the feedback you provide through our on-line suggestion box, you make it worth every penny to us.

How do you buy a copy? Just call 1-800-800-NeXT, say "Hi, I'm a student and I'd like to start thinking when I draw. Please send me Diagram!" and it'll be shipped to you overnight. Don't postpone joy.

Roger Rosner
Lighthouse Design, Ltd.
rock@lighthouse.com

System Works: Graphical Shell Scripts

For those that use and write shell scripts, have any of you tried SystemWorks-version 1.1?

It is available on sonata.cc.purdue.edu. You should recompile it when you get it.

What is SystemWorks?

SystemWorks is the first NextStep-based visual oriented shell (as far as I know). You can process files with `scripts' simply by dragging an Icon from Workspace Manager to one of `script windows'.

SystemWorks can perform conventional UNIX operation (such as tar-compress or uudecode) within Icon-oriented framework, and it doesn't require wide knowledge of UNIX, since such complications are hidden behind the Icon metaphor.

SystemWorks is intended to be a visual counterpart for Terminal. SystemWorks can display its current status by realtime animation. For example, when you drag TeX files into a ``TeXprocessor" script, it shows flow of data (tex -> dvi, dvi -> ps, ps -> preview for example) with at-a-glance animation (of course, you can suppress the animation if you asked to do so). Input data and output data are also represented as Input/Output Icon, to make it easier to grasp what's being done.

SystemWorks can encapsulate complications of UNIX commands. With well-designed scripts, you can use UNIX commands without examining hairy details on manual pages. Each script represents `what the script does' in a few words, as opposed to ordinary UNIX command-line interface, where you must describe `how to do it'.

Of course, you could do that if some friendly gurus provide friendly shell scripts; but it's far easier to drag an Icon than typing VERY LONG file name CORRECTLY. What if you must type "/NextLibrary/Documentation/NeXT/SysRefMan/22_ClassSpecs/AppKit/Application.wn" at your Terminal? Now you have an alternative way.

How can I use it?
(Novice Users' section)

If you are new to both UNIX and SystemWorks, check `Novice user mode' in Preference panel. It will disable some menu items, to avoid bothering you with obscure features.

Obtain scripts from somewhere; e.g. friendly gurus, FTP sites...

(continued on next page)

NeXTPress

(continued from previous page)

If you are completely new to UNIX, you must ask friendly UNIX users for assistance. Tell them what you want to do, and ask whether it's possible to do the job with this software.

- Select `Script', then `Open'.
- Choose the script file in Open Panel.
- Once you have preprogrammed scripts, all you need to do is to open the script. It can be performed in standard manner. Double click in the Workspace Manager also works as it should. Keeping a miniaturized browser for the script directory might be a good idea.
- Drag the file you want to process into the newly created window.
- The title of each script window will tell you what the script does.

If you want to perform certain task on a certain file, bring up the script window (double-click when miniaturized, select `Activate` menu when hidden under other windows), then drag the Icon into that window.

For further information...

Some scripts are included in the official distribution of this software. Detailed documentation is also included in the distribution package. Ask your system administrator if you can't find it yourself.

How can I program it?
(UNIX Users' section)

This section documents differences with conventional UNIX shell. For further details on each menu/panel/window, consult the document "SystemWorks.wn", which should be enclosed in the distribution package.

Basically, sequential execution and piped execution are identical. Every UNIX command takes some sort of data as input; output goes to somewhere visible. In that sense, piped commands and sequential commands are identical. Therefore, in SystemWorks, no distinction is made between these two, to simplify the visual representation and to encapsulate details of UNIX.

SystemWorks automatically controls the sequence of execution; namely, if two processes are connected via pipes, they will be executed concurrently. If two processes are connected via intermediate files, they will be executed consecutively.

You don't have to worry about temporary files and redirections. If a filter takes input from a file, and if its previous filter writes output to standard output, automatic redirection takes place; i.e. output of the previous filter goes to temporary file, then the temporary file will be fed to the filter.

This feature improves modularity of each filter. Such modularity enables you to perform complex tasks just by putting elemental functions each other.

However, misuse of such automaticity will lead to unwanted results. Careless scripts may fill the /tmp filesystem with junk. If you specified that "cat -" would read input from file, it will block immediately.

Variables are similar to those for-shell scripts. Currently, the following shell variables are available:

- *Source directory*: where incoming file resides.
- *Input file*: basename of input file.
- *Root name*: name of input file, with last extention stripped.
- Selected menu items in SystemWorks' PopUp List.

You can't define your own variables, however. This restriction is necessary to maintain modularity on each filter, because the ability to define your own shell variable in each filter implies ability to maintain context over filters. What if some novice users incorporated your context-sensitive filter into other scripts?

Designing scripts...

There are some places to take care when you design your own version of a script.

- The Output Icon should represent an output file / device.
- Don't expect versatility from PopUp List.

Acknowledgments

First of all, I would like to thank Ms. Erica J. Liebman. This project was triggered by her article, "What's Up Dock?", *NeXT Users' Journal* Issue 6.

(continued on next page)

NeXTPress

(continued from previous page)

I would also like to thank Mr. Kris Jensen. MoveMatrix code was derived from his column "Stone Age" of *NeXT Users' Journal* Issue 7.

I learned much about Speaker/Listener from the NextAnswers. Icon dragging and Icon image cloning code was derived from this.

NextAnswers, /NextDeveloper/Examples and NeXT Users' Journal have been my invaluable source of information. I would like to say thanks to all those who concerned with these projects. This program couldn't exist without these great work.

Thanks also goes to Dr. Suguru Yamaguti and Dr. Shinji Shimojo at this Laboratory, for comments, criticisms and suggestions on the design of this software.

Youki Kadobayashi
<youki@ics.osaka-u.ac.jp>
Information Network Architecture Laboratory
Department of Information and Computer Sciences
Faculty of Engineering Science
Osaka University, Japan

NeXT Internationalization

What is Localization?

Localization is the process of making your application language-independent, so that a user can choose to use your application in any of a number of languages that you make available. The process of localizing your application consists of two distinct phases:

• First, you make your application localizable. That is, you build localization support into your application by centralizing the language-specific text, images, and sounds and placing them in files in a language-specific subdirectory of your project directory.

• Second, you localize the application. That is, for each additional language that you wish to support, you create language-specific versions of the text, image, and sound files. This translation process doesn't involve modifications to your source code (although any nib files will need to be edited with Interface Builder), so you can have this step performed for you either by an in-house translator or by an outside translation service.

Even if you don't have immediate plans to support multiple languages in your application, there are advantages to designing your application so that it provides support for localization: First, with proper design your application's source code won't have to be touched in order for the application to be localized; therefore, you won't run the risk of introducing additional bugs by putting the necessary hooks in later. Second, testing of the localization code can be done along with the initial monolingual product, so the amount of testing needed for any future localized version will be minimized.

Currently, localization support is provided for the following areas:

• Embedded text strings
• Text strings in nib files
• TIFF and EPS files (which may contain text or culture-specific images)
• Date, time, and currency formats
• Auxiliary files (such as help files and sound files)

In addition to this technical support note, you'll need the following files in order to make your application localizable:

• The source code files *Localization.h* and *Localization.m*
• *Makefile.preamble* and *Makefile.postamble*
• The program *genstrings*

These files are available from the Internet archive server *cs.orst.edu*.

In the current release, language support is provided for English, French, German, and Spanish. If you wish to provide other localized versions of your application, you'll need to contact NeXT Developer Support for additional information and support. In general, you can localize your application for any language that can be expressed in the standard NeXTstep character encoding.

Kyoto Common Lisp

AKCL(Austin Kyoto Common Lisp)-1.530 for NeXT computers is now available on sonata.cc.purdue.edu[128.210.15.30] in

pub/next/submissions/akcl-1.530.NeXT.bin.tar.Z

(continued on next page)

NeXTPress

(continued from previous page)

via anonymous ftp. It runs on NeXT OS 2.[01]. The distribution contains executable files of AKCL-1.530, GCC-1.40, and modified GAS-1.38 which generates Mach-O object files.

The uncompressed `saved_kcl' is large - about 9.9 MB. Since NeXT's sbrk() doesn't assure contiguous memory allocation after restarting from a saved image, I replaced it with my own routine which allocates large virtual memory once. This approach made the saved_kcl large.

This port uses GCC-1.40 and modified GAS-1.38 instead of NeXT's (g)cc and (g)as to compile C-files generated by the AKCL compiler. The current version doesn't work well with NeXT's (g)as.

The restrictions of this port are as follows:

- The process size is fixed. (maximum pages = 4096)
- `system::faslink' is not supported.

Noritake Yonezawa
(yonezawa@cs.uiuc.edu)
Department of Computer Science
University of Illinois at Urbana-Champaign

Demo Software: Gdb, SpinDoctor, PhaseScope

The demo versions of Gdb, SpinDoctor, and PhaseScope are available for ftp from cs.orst.edu in the directory /pub/next/demos. A short description of each program provided below:

SpinDoctor by Fred Richards at Univ. of Illinois
• A program for visualizing three-dimensional experimental data.

Gdb by Bing-Qing Chen at Caltech
• A GNU Debugger Front End.

PhaseScope by Dr. Michael J. Mezzino at Univ. of Houston - Clear Lake
• A comprehensive user-interface and general graphical display program for qualitatively analyzing the stability characteristics of dynamical systems.

If you have any problems getting or using the demo versions from the ftp-site, please let me know. Also, if you would like more information about these programs or their prices, please send the request to impact@impact.shaman.com.

Daniel D. Hung
Impact Software Publishing, Inc.
210 Lake Street,
Ithaca, NY 14850
E-Mail (NeXT):
dhung@impact.shaman.com

NeXT Midi Sequencer

I have uploaded the file Z-quencer.tar.Z to the following archives:

sonata.cc.purdue.edu
cs.orst.edu

both in the directory: /pub/next/submissions It contains 2 files:

- *README.rtf*: readme file
- *Z-quencer*: executable

- *From README.rtf:*

This program is a very basic MIDI sequencer. It allows you to :

- read/write MIDI files
- read/write Score files
- graphically edit the notes contained in MIDI and Score files
- create new parts and add notes to them
- play the edited Score[1] via the MIDI interface (on port B)
- change the MIDI channel of parts in a MIDI (Score) file

Have fun and let me know what you think!

Ralph Zazula
University of Arizona
Department of Physics
UAZHEP::ZAZULA
(DecNet/HEPNet)
zazula@uazhe0.physics.arizona.edu (Internet)
zazula@bonehead.tucson.az.us
(NeXT mail)

NeXT Signal Processing

Anyone trying to use the DSP as a "black box" for signal processing should be interested in the latest version of the dsp_dma_stream programming example. It illustrates efficient DMA streaming in and out of the DSP. The compressed tarfile may be picked up as

dsp_dma_stream.tar.Z.

(continued on next page)

1. The internal representation of a MIDI or Score file is a MusicKit Score object

NeXTPress

(continued from previous page)

For people in the outside world, this file has been placed in the CCRMA ftp directory. (CCRMA is Stanford's Center for Computer Research in Music and Acoustics.) To ftp to CCRMA, say

ftp ccrma-ftp.stanford.edu

or, if that fails, try

ftp 36.49.0.84

When you get the login prompt, log in as "anonymous" with any password. Then say

cd pub
get CCRMA.README
bin
get dsp_dma_stream.tar.Z

Here is the current contents of the file CCRMA.README:

cm.tarfile.Z:
Common Music
clm.tarfile.Z:
Common Lisp Music
pcl.tarfile.Z:
(PCL Mayday release)
SPASM.tar:
SPASM program version 2.106
ResoLab.Z:*
ResoLab program, run it and ask for help
resample.tar.Z:
Utilities for sampling-rate conversion by arbitrary factors
dsp_dma_stream.tar.Z:
Programming example illustrating bidirectional DSP DMA
DMIX/:
Dan's stuff
omr/:
GRD <-> McGill stuff

(All software is NeXT based.)

The State of the NeXT Economy

NeXT Q2 REVENUES INCREASE 86% OVER Q1

NeXT Achieves First $46 Million Quarter

REDWOOD CITY, Calif., July 31, 1991 NeXT Computer, Inc. today announced that it achieved $46 million in revenues in the second quarter of 1991, which ended June 30, on sales of its line of professional workstations. Revenues increased 86 percent compared to the first quarter of 1991.

Seventy percent of NeXT's shipments went to customers in business and government and 30 percent went to colleges and universities. More than 49 percent of NeXT's total shipments went to customers in Europe and Asia. NeXT coordinates its European activities through its three subsidiaries in the United Kingdom, France and Germany. In Asia, Canon Inc. is the company's exclusive representative.

"The investments we have made are beginning to pay off," said Steven P. Jobs, president and CEO of NeXT. "Companies are buying NeXT computers because our object-oriented system software enables them to develop their mission-critical custom applications many times faster than on any other platform and because they want to use networks of NeXT computers for Interpersonal Computing."

NeXT Pioneered Object-Oriented System Software

NeXTstep, NeXT's object-oriented system software (OOSS), enables NeXT's customers to develop up to five times faster the custom applications they need to compete in today's global economy. The recent formation of industry alliances, such as the partnership between Apple and IBM, reflects the computer industry's recognition that OOSS is needed to facilitate programming and end-user interaction with computers. Today NeXT offers the only platform that has been based on an object-oriented environment from its inception.

"We view the IBM /Apple announcement as a 100-percent endorsement of the strategy we embarked upon over five years ago and are delivering today," said Jobs. "Our customers are choosing to get object-oriented system software now, rather than waiting for the rest of the industry to catch up in three to five years."

NeXTstep is the first mainstream software development and user-interface environment that makes complex, powerful operating systems accessible to users, while making it easy for programmers to create easy-to-use applications.

According to NeXT customer Hadar Pedhazur, vice president, Union Bank of Switzerland Securities, Inc. in New York, "The strongest point for us is the NeXTstep object-oriented development environment. The benefits are monumental. It is

(continued on next page)

NeXTPress

(continued from previous page)

inconceivable to me that we would ever go back to the days of functional programming."

NeXT Customers

NeXT customers purchase NeXT workstations for four reasons: to develop mission-critical custom applications; to run large networks, sometimes on a global scale; to run more powerful versions of productivity applications than are available for personal computers; and to use interpersonal computing capabilities to increase group productivity and collaboration.

These users require the power, networking and multitasking capabilities of UNIX workstations, but also need a computer that runs personal computer-like productivity applications and is easy to use. They work in areas such as financial services, medical and legal services, government, publishing and higher education.

NeXT computers have the technology that commercial users are seeking as they migrate to workstation technology. NeXTstep, NeXT's graphical user interface and application development environment, offers the most powerful and easiest to use environment for custom application development. NeXT's systems also provide extremely powerful built-in networking capabilities including TCP/IP and both twisted-pair and thin Ethernet yet setting up and navigating around a network on NeXT is extremely simple.

Unlike other workstation vendors, NeXT has a wide assortment of breakthrough productivity applications for its computers. In addition to Improv from Lotus and WordPerfect, Adobe Illustrator began shipping in the second quarter. These applications are superior in functionality to, but compatible with, their versions on other platforms. Lastly, NeXT is the only computer company focusing on interpersonal computing, with built-in, easy-to-use electronic mail that allows everyone in an organization to use voice, text and graphics to keep each other informed.

Many customers have chosen NeXT systems for their ease of use. "User friendliness was the key factor in choosing NeXT; we wanted to make sure the system chosen would be usefully and fully employed," said Gary Johnson, vice president, law of Northern States Power in Minneapolis.

Other customers find that NeXT allows them to provide a single, unified computing environment to take the place of a hodge-podge of workstations and personal computers, providing increased functionality and communication. "NeXT computers replace the collection of terminals and machines our traders used to use and provides them with extraordinary communication capabilities," said Eric Spahr, vice president at Lehman Brothers in New York.

Second-Quarter Update

NeXT's second-quarter revenues came from continued sales of NeXTstation and NeXTcube, which began shipping in November 1990; from the first full quarter of shipments of NeXTstation Color, which began shipping March 12; and from initial shipments of NeXTdimension, the company's sophisticated, 32-bit color PostScript workstation, which began shipping May 28. All four members of NeXT's professional workstation family are based on Motorola's 68040 microprocessor.

During the second quarter, NeXT also began shipping a Kanji version of its computers in Japan, where NeXT's method of handling Kanji was hailed as the best implementation ever offered to handle the complex characters by computer. NeXT also began shipping 1.4 gigabyte hard drives May 28, making it possible to configure workstations with 2.8 gigabytes of storage.

NeXT expects significant growth to continue during the latter half of 1991; however, third-quarter revenues are expected to be relatively flat because sales in Europe and to higher education generally drop due to summer vacations.

New Mathematica Journal

I am pleased to announce the launching of a newsletter,

Mathematica in Education

Mathematica in Education will contain articles and notes by people using Mathematica in all areas of academia who wish to share their experiences and projects with others. The articles and notes

(continued on next page)

NeXTPress

(continued from previous page)

will be organized into the following broad categories:

- Experiences in implementing Mathematica laboratories and classroom usage.
- Academic projects that use Mathematica in a significant manner.
- Curricular changes related to the availability and usage of Mathematica.
- Philosophical and pedagogical issues.
- Software and hardware compatibility issues.
- Resources (both human and electronic), including user groups and archives.
- Calendar of events (workshops, conferences, etc.).

Mathematica in Education will be produced quarterly, beginning with the Fall 1991 inaugural issue. Subscriptions will cost $15 per year.

I invite you to submit an article or note for the initial issue of *Mathematica in Education*. Notes should be no more than 500 words and articles should contain approximately 3500 words. Although this is not the proper forum for the distribution of Mathematica packages and Notebooks (see *The Mathematica Journal*), it is appropriate to indicate how the reader could obtain any materials that you are making available.

If you are interested in submitting a contribution, please send two copies of the material, together with a cover letter indicating that it is being submitted for publication in *Mathematica in Education*, to the following address:

Paul Wellin, editor
Mathematica in Education
Department of Mathematics
Sonoma State University
1801 E. Cotati Avenue
Rohnert Park, CA 94928
e-mail: wellin @sonoma.edu

Documents may be submitted electronically on Macintosh diskettes to the same address, and should also be accompanied by the article or note in hard copy. Materials may be sent via email, but hard copy should also be submitted through the postal system or by fax. The use of TeX (or any of its supersets LaTeX or AmS-TeX) is encouraged, although documents produced in other formats will not be discriminated against. In particular, documents may be submitted as Mathematica Notebooks if desired.

NeXT at SIGGRAPH '91

(continued from page 1, column 2)

the glitzy Electronic theatre showing videos such as Pixar's "Heavy and Light" starring Luxo Jr., a virtual reality exhibit, and a huge exhibition where Apple, DEC, HP, IBM, Silicon Graphics (SGI), Sun, and scores of others showed their latest graphics hardware and software. The following comments represent my view of NeXT's presence at SIGGRAPH. It comes mainly from memory and thus some of the details may be inaccurate. NeXT is in no way responsible for anything said here.

Was NeXT at SIGGRAPH '91? Well, no and yes. They didn't have an exhibit booth on the gigantic convention floor. Why not? Since a small booth can be rented for $10,000 the cost certainly wasn't the reason (a company that shipped 8000 CPUs in Q1 and did $46,000,000 worth of business in Q2 can certainly afford $10,000). Then why didn't NeXT have a booth? First, NeXT doesn't really have much 3D software. Sure Pixar's Renderman has been ported but there is no modeling software. Having an exhibit on the same floor as SGI and IBM with GL, DEC and others with AVS, PEX (PHIGS+ extensions to X) everywhere, virtual reality headsets and data gloves etc. without even a 3D modeler wouldn't be very impressive. Even though NeXT is clearly the best machine right now for 2D graphics (with Display PostScript and just a few killer apps like Adobe Illustrator 3.0) it's just not as flashy as all that 3D stuff and virtual reality demos. Perhaps another reason is that when Steve decides to exhibit at SIGGRAPH he'll do it in real style with a big booth and several killer 3D apps. The rollouts at Davies Hall have convinced me of that. On the other hand, there were a couple of NeXTs in others' exhibits, including a Wolfram exhibit demonstrating Mathematica 2.0. (By the way, SGI was all over the place - as the hardware used in most high-end software company booths and for most of the best videos in the Electronic Theatre. SGI is clearly the leader for high-end hardware at this moment.)

Although they had no booth on the exhibition floor NeXT did put on a wine and cheese reception in a good sized room at Caesar's

(continued on next page)

NeXT at SIGGRAPH '91

(continued from previous page)

Palace on 8/1/91. There were 2 NeXTdimension and 2 NeXTstation Color systems, several NeXT employees and as many as 50 people in the room at its most crowded. I noticed at least one graphics superstar in the room, namely Don Greenberg of Cornell, Mr. Radiosity and Coon's award winner at SIGGRAPH '87. At the reception I sat down for a really fun hour watching Keith Ohlfs of NeXT in action with a NeXTdimension system. Now Keith is a truly talented fellow. In addition to writing Icon, Scene, Grab and having his own bundled font, Keith had a lot to do with the look of the NeXTstep interface, including its 3D appearance. As it turns out Keith is also a really great guy willing to chat with a mere mortal like myself and quite humble (he knows he blew Icon in 2.0 but read on). Keith has completely redesigned Icon (now known as Pixelist - excuse the possible spelling error) to be much easier to use and apparently bug-free. The tools and inspector panels are now combined, the interface is more natural, and one can set a preference to keep the new panel attached to the image window - something I know I'll use. I sure hope (and got an inkling that) Pixelist becomes a NeXTapp in NeXTstep Release 3.0. We really need a standard, fully-supported, bundled app to manipulate images.

With the help of a Sony HandyCam and cute little 2 inch monitor (both which match the NeXT hardware quite nicely, thank you), Keith also demonstrated NeXTtv, his new app called Oscar and a powerful new object called NeXTvideo (again please excuse the possible spelling errors and my ignorance of the object's creator). The NeXTvideo object was available in Interface Builder (IB) through a loaded palette (with a great "eye" icon atop the Palettes window). In under two minutes Keith built an app as follows. He dragged a NeXTvideo object from the Palettes window into the main window and made it larger, dragged in a couple of buttons and renamed them "Start" and "Stop," connected the buttons to the NeXTvideo object so that start: and stop: messages would be sent when the respective buttons were clicked, and put IB in test interface mode. Clicking the Start button made the live video camera input show up in the NeXTvideo object, clicking the Stop button froze a particular image in the object, which could then be manipulated with cut, copy and paste, etc. All of this was done in under two minutes without any programming or compilation at all - and the app worked in test mode! I dare anyone to build a comparable app on a loaded Mac, Sun or whatever in under two days. Of course the reasons Keith could do this so easily are the powerful object-oriented environment, the brilliantly designed Interface Builder and the muscular NeXTdimension board.

Speaking of OOP, there was a panel on Object-Oriented Graphics, chaired by Nancy Knolle Craighill of Sony, which attracted several hundred people on 8/2/91. The panel members reviewed applications that demonstrate the need for object-oriented graphics classes (beyond user interface classes) and OOP development environments. Several times Nancy mentioned that NeXTstep is a great environment for object-oriented graphics development (and thus the best kept secret in the graphics industry becomes a little less kept). On the exhibit floor I also noticed an preliminary version of a book called *Computer Graphics Using Object-Oriented Programming* (edited by Nancy and 3 others) which contains a paper describing a hierarchy of graphics classes under NeXTstep. Look for the book to be available early in 1992 from Wiley publishers.

So NeXT was at SIGGRAPH '91 even though it wasn't there in a big way. At SIGGRAPH '92 in Chicago I hope for and expect a big splash (including a booth) from NeXT. With the great development environment and NeXTdimension hardware it can't be long before a killer 3D modeler and lots of other 3D software becomes available. When it does, watch out SGI, Sun and others. IBM might even end up being a bit red (read "pink") in the face for opting for the Apple OOP environment which is still years away.

Thanks to the following NeXT employees or enthusiasts (NeXToids?) I met and enjoyed interacting with at SIGGRAPH: Dan, Wayne, Todd, Keith, Joseph (and other SCaNners), Bruce Blumberg (my NeXT Developer Camp instructor), and Android Stone (developer of Create, TextArt, and DataPhile).

COMPRESSION LAB SOURCE CODE

CLabControler.h

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

#import <objc/Object.h>

@interface CLabControler:Object
{
// IB outlets
    id timingForm;
    id compressionForm;
    id uncompressedScrollView;
    id compressedView;
    id jpegSlider;
    id modeRadioMatrix;
    id compressedScrollView;
    id jpegText;
    id uncompressedView;
    id compressItButton;
    id saveMenuItem;
    id saveAsMenuItem;
// Other Variables
    char fileSaved[1024];
}

// IB methods.
- open:sender;
- saveAs:sender;
- setJPEGFactor:sender;
- setCompressionType:sender;
- compressIt:sender;
- quit:sender;
- save:sender;

// Window Delegate Methods - So we can quit on window close.
- windowWillClose:sender;

// Application Delegate Methods - So we can do some more initialization after
// startup.
- appDidInit:sender;

//  Internal methods.
- enableSaves:(BOOL)canDo;
- enableCompress:(BOOL)canDo;
- enableCompressionControls:(BOOL)canDo;
@end
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

CLabControler.m

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

#import "CLabControler.h"
#import "NXImageView.h"
#import "Timer.h"
#import "streamSize.h"
#import <appkit/OpenPanel.h>
#import <appkit/SavePanel.h>
#import <appkit/Slider.h>
#import <appkit/TextField.h>
#import <appkit/tiff.h>
#import <appkit/NXImage.h>
#import <appkit/NXBitmapImageRep.h>
#import <objc/List.h>
#import <appkit/Application.h>
#import <appkit/Matrix.h>
#import <appkit/Form.h>
#import <appkit/ScrollView.h>
#import <appkit/Button.h>
#import <appkit/MenuCell.h>
#import <strings.h>


@implementation CLabControler


- open:sender
// Presents the user with a panel to get the tiff file to open. Directs the
// uncompressedView to display the image.
{
    static const char *const tiffType[] = {"tiff","TIFF","tif","TIF", NULL};
    id openPanel = [OpenPanel new];
    [openPanel allowMultipleFiles:NO];
    [openPanel runModalForTypes:tiffType];
    if ([openPanel filename])
        [compressionForm setIntValue:
            [uncompressedView openfile:[openPanel filename]] at:0];
    fileSaved[0] = 0;
    [self enableCompress:YES];
    [self enableSaves:NO];
    [self enableCompressionControls:YES];
    return self;
}
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

```
- saveAs:sender
// Presents the user with the standard save panel to get the filename to save
// the COMPRESSED image under.
{
    id savePanel = [SavePanel new];
    [savePanel setRequiredFileType:"tiff"];
    [savePanel runModal];
    strcpy(fileSaved,[savePanel filename]);
     return self;
}

- setJPEGFactor:sender
// This updates the text field and slider to match, checks for outofbounds,
// and disables the menus, since the new tiff hasn't been done.
{
    float tempFactor;

    tempFactor = [sender floatValue];
    if (tempFactor > 250.0)
       tempFactor = 250.0;
    if (tempFactor < 1.0)
       tempFactor = 1.0;
    [jpegText setFloatValue:tempFactor];
    [jpegSlider setFloatValue:tempFactor];
    [modeRadioMatrix selectCellAt:2 :0];
    [self enableSaves:NO];
    [self enableCompress:YES];
     return self;
}

- setCompressionType:sender
// This disables the save menus, but doesn't retreive the type, it's retrieved
// at need.
{
    [self enableSaves:NO];
    [self enableCompress:YES];
     return self;
}


- compressIt:sender
// This routine assumes that there is an image in uncompressedView.  Therefore
// the Compress It button should be disabled when there isn't. Simple huh?
// This simply convinces the NXImage that it should save the image at the
// current compression parameters, times the operation, then reloads the tiff,
// timing the decompression and displays it in the other scrollview.  It uses
// a memory stream for the save, so more memory will make it faster. Also
// it displays the size of the image before and after compression and savings.
{
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

```
    int compTable[] = {NX_TIFF_COMPRESSION_LZW, NX_TIFF_COMPRESSION_PACKBITS,
        NX_TIFF_COMPRESSION_JPEG};
    NXStream *tiffStream;
    id tiffTimer = [[Timer alloc] init];

//  Open the tiffStream, then after, write to disk.
    tiffStream = NXOpenMemory(NULL, 0, NX_READWRITE);
// Start timer here then write the file
    [tiffTimer start:self];
    [[uncompressedView imageRep]
        writeTIFF:tiffStream
        usingCompression:compTable[[[modeRadioMatrix selectedCell] tag]]
        andFactor:[jpegSlider floatValue]];
// Stop the timer here and update the stats.
    [tiffTimer stop:self];
    [timingForm setFloatValue:[tiffTimer floatValue] at:0];
    [compressionForm setIntValue:streamSize(tiffStream) at:1];
    [compressionForm setIntValue:([compressionForm intValueAt:0] -
        [compressionForm intValueAt:1]) * 100 /
        [compressionForm intValueAt:0] at:2];

// Read the compressed stream in again, into the compressed view
// start and stop the timer before and after the readTIFF: call.
    [tiffTimer reset];
    [tiffTimer start:self];
    [compressedView readFromStream:tiffStream];
    [tiffTimer stop:self];
    [compressedView display];
    [timingForm setFloatValue:[tiffTimer floatValue] at:1];

    NXCloseMemory(tiffStream, NX_FREEBUFFER);
    [self enableCompress:NO];
    [self enableSaves:YES];
    [tiffTimer free];
     return self;
}

- quit:sender
//  This is just a little glue so that the window close box causes CL to quit.
//  Also, I may add a confirmation panel or let the user save first.
{
    [NXApp terminate:self];
     return self;
}

- save:sender
//  This checks fileSaved to see if we have saved the file before, if we have,
//  it just saves under that same name, otherwise it calls saveAs.
{
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

```
    int compTable[] = {NX_TIFF_COMPRESSION_LZW, NX_TIFF_COMPRESSION_PACKBITS,
        NX_TIFF_COMPRESSION_JPEG};
    NXStream *tiffStream;

    if(!fileSaved[0] || [[sender selectedCell] tag] == 2)//Need Name saveAs
        [self saveAs:self];
// First byte is non-zero, we have a valid filename.
//  Open the tiffStream with NXOpenMemory, then after, write to disk.
    tiffStream = NXOpenMemory(NULL, 0, NX_READWRITE);
    [[compressedView imageRep]
        writeTIFF:tiffStream
        usingCompression:compTable[[[modeRadioMatrix selectedCell] tag]]
        andFactor:[jpegSlider floatValue]];
    NXSaveToFile(tiffStream, (const char *)fileSaved);
    NXCloseMemory(tiffStream, NX_FREEBUFFER);
    return self;
}

// Window Delegate Methods - So we can quit on window close.
- windowWillClose:sender
{
    [self quit:self];
    return self;
}

// Application Delegate Methods - So we can do some more initialization after
// startup.
- appDidInit:sender
{
//  We create the NXImageViews here.  They take care of creating the actual
//  NXImage instances.  Then we put them in the scrollviews. Also we setup the
//  booleans that tell us what should be enabled in the doc menu, and how we
//  should do saves.

    fileSaved[0] = 0;

    compressedView = [[NXImageView alloc] init];
    uncompressedView = [[NXImageView alloc] init];
    [compressedScrollView setDocView:compressedView];
    [uncompressedScrollView setDocView:uncompressedView];
//  This lets the view know that it should check it's size and it's superviews
//  size and let the scrollview know it should add scrollers if need be.
    [compressedView isScrollDocViewOf:compressedScrollView];
    [uncompressedView isScrollDocViewOf:uncompressedScrollView];

//  Setup the jpegText so that it displays the right number of digits.
    [jpegText setFloatingPointFormat:NO left:3 right:1];
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

```
// Disable the save menus and the compress button.
    [self enableSaves:NO];
    [self enableCompress:NO];
    [self enableCompressionControls:NO];
    return self;
}

// Internal methods.
- enableSaves:(BOOL)canDo
// This method just enables/disables the save and saveas menucells
{
    [saveMenuItem setEnabled:canDo];
    [saveAsMenuItem setEnabled:canDo];
    return self;
}

- enableCompress:(BOOL)canDo
// This method just enables/disables the compressit button
{
    [compressItButton setEnabled:canDo];
    return self;
}

- enableCompressionControls:(BOOL)canDo
// This method just enables/disables the compressit button
{
    [jpegSlider setEnabled:canDo];
    [jpegText setEnabled:canDo];
    [modeRadioMatrix setEnabled:canDo];
    return self;
}

@end
```

NXImageView.h

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

#import <appkit/View.h>

@interface NXImageView:View
{
id imageRep;
id scroller;
}

- initFrame:(const NXRect *)frameRect;
- (int)openfile:(const char *)filename;
- readFromStream:(NXStream *)tiffStream;
- isScrollDocViewOf:(id)scrollView;
- drawSelf:(const NXRect *)rects :(int)rectCount;
- imageRep;
@end
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

NXImageView.m

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

#import "NXImageView.h"
#import "streamSize.h"
#import <appkit/NXBitmapImageRep.h>
#import <appkit/ScrollView.h>

@implementation NXImageView

- initFrame:(const NXRect *)frameRect
{
    imageRep = scroller = nil;
    return [super initFrame:frameRect];
}

- (int)openfile:(const char *)filename
{
    NXStream *tempStream;
    int len;

    tempStream = NXMapFile(filename, NX_READONLY);
    [self readFromStream:tempStream];
    len = streamSize(tempStream);
    [self display];
    NXCloseMemory(tempStream, NX_FREEBUFFER);
    return len;
}

- readFromStream:(NXStream *)tiffStream
{
    NXRect tempRect;
    NXSize tempSize;

    [imageRep free];
    imageRep = [[NXBitmapImageRep alloc] initFromStream:tiffStream];
// set up our frame from the size of the bitmap (in pixels)
    tempRect.origin.x = 0; tempRect.origin.y = 0;
    tempRect.size.height = [imageRep pixelsHigh];
    tempRect.size.width  = [imageRep pixelsWide];
    [self setFrame:(const NXRect *)&tempRect];
// if we are larger than the size of the scrollview, give it scrollbars
    if(scroller) {
        [scroller getContentSize:&tempSize];
        [scroller setHorizScrollerRequired:
            (tempRect.size.width > tempSize.width)];
        [scroller getContentSize:&tempSize];
        [scroller setVertScrollerRequired:
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

```
            (tempRect.size.height > tempSize.height)];
        [scroller getContentSize:&tempSize];
        [scroller setHorizScrollerRequired:
            (tempRect.size.width > tempSize.width)];
    }
    return self;
}

- isScrollDocViewOf:(id)scrollView
{
    scroller = scrollView;
    return self;
}

- drawSelf:(const NXRect *)rects :(int)rectCount
{
    [imageRep drawIn:&frame];
    return self;
}

- imageRep
{
    return imageRep;
}
@end
```

Timer.h

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

#import <objc/Object.h>
#import <sys/time.h>

@interface Timer:Object
{
    struct timeval elapsedTime, startTime;
    struct timezone ignored;
    BOOL running;
}

- init;
- reset;

- start:sender;
- toggle:sender;
- stop:sender;
- (BOOL)running;

- (float)floatValue;
- setFloatValue:(float)elapsed;

@end
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

Timer.m

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

#import "Timer.h"
#import <math.h>

#define FRACTMAX 1000000

@implementation Timer

- init
{
    timerclear(&elapsedTime);
    running = NO;
    return [super init];
}

- reset
{
    timerclear(&elapsedTime);
    running = NO;
    return self;
}

- start:sender
{
    if (!running) {
        running = YES;
        gettimeofday(&startTime, &ignored);
    }
    return self;
}

- stop:sender
{
    struct timeval temptime;

    if (running){
        gettimeofday(&temptime, &ignored);
        if(temptime.tv_usec < startTime.tv_usec){
            temptime.tv_sec--;
            temptime.tv_usec += FRACTMAX;
        }
        elapsedTime.tv_usec += temptime.tv_usec - startTime.tv_usec;
        elapsedTime.tv_sec  += temptime.tv_sec  - startTime.tv_sec;
        if(elapsedTime.tv_usec > FRACTMAX) {
            elapsedTime.tv_usec -= FRACTMAX;
            elapsedTime.tv_sec++;
        }
    }
    return self;
}
```

(continued on next page)

Compression Lab Source Code

(continued from previous page)

```
- toggle:sender
{
    if (running)
        [self stop:self];
    else
        [self start:self];
    return self;
}

- (BOOL)running
{
    return running;
}

- (float)floatValue
{
    float temp = elapsedTime.tv_usec;

    temp /= FRACTMAX;
    return (elapsedTime.tv_sec + temp);
}

- setFloatValue:(float)elapsed
{
    elapsedTime.tv_sec = floor(elapsed);
    elapsedTime.tv_usec = (elapsed - floor(elapsed)) * FRACTMAX;
    if (running){
        gettimeofday(&startTime, &ignored);
    }
    return self;
}
@end
```

StreamSize.h

```
// streamSize.h  Simply a function to determine the size of an NXStream.
#import <streams/streams.h>

int streamSize(NXStream *theStream);
```

StreamSize.m

```
// streamSize.c  Simply a function to determine the size of an NXStream.

#import "streamSize.h"
#import <streams/streams.h>


int streamSize(NXStream *theStream)
{
    char *addr;
    int len;
    int maxlen;
    NXGetMemoryBuffer(theStream, &addr, &len,&maxlen);
    return len;
}
```

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. 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, 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 adjacent box "Downloading .."). We read it and love it and occasionally make copies available at our meetings.

Downloading SCaNeWS (from an Archive)

If you miss(ed) any issues of *SCaNeWS* and have access to Internet, you can obtain them via anonymous file transfer protocol (ftp) as follows (**bold** indicates onscreen prompts, *italics* what you type):
At your local system prompt, type

ftp -n nova.cc.purdue.edu
ftp> *user anonymous <your full address>*
(e.g. mahoney@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 the NextApp *Preview* and printed.

SCaNeWS CONTRIBUTORS

Contributors:
Mike Mahoney
Robert Thille

Editors:
Mike Mahoney
Lorraine Rapp

If you would like to submit an article to *SCaNeWS*, contact Mike or Lorraine.

Articles are copyrighted by their authors.

SCaNeWS is composed using FrameMaker™

If you have any comments or questions about SCaN or *SCaNeWS*, please contact one of the above officers

SCaN OFFICERS

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

VICE-PRESIDENT: Bob Desharnais
(213) 343-2056
bob@biol1next.calstatela.edu

SECRETARY: Lorraine Rapp
rapp@csulb.edu

TREASURER: Henry Chiu
chiu@csulb.edu