Newsletter of the NeXT Programmer's Community

NeXT Users' journal

May 1990 • ISSUE 6

WELCOME

Erica J. Liebman

This is rather a special month of firsts- two new arrivals and one special announcement. First, Joshua David "Quimby" Stieglitz, my very first nephew at a staggering six pounds, ten ounces, dark hair, dark eyes and a perfect nose. Second, the first issue of the NeXT Users' Journal. Finally, with all mention of a certain April Fool's article aside, NeXT really did announce their 68040 machine this month. I've included the official press release.

The 68040 machine is due out late this year, but the new format of this newsletter and Joshua are here right now.

I am hoping that with this issue, as with my new nephew, we will find this Newsletter growing, learning and exploring the world around us. In this NeXT Users' Journal we again are showing our commitment to discussions of new issues, to exploring programming problems and solutions and to always including code where possible. We are continuing our reviews, market and user group information.

Additionally, we are hoping to start some new special interest groups. Conrad Geiger (conrad_geiger @next.com) has offered to help coordinate getting leaders for these new groups. Space will be put aside in each issue of the Journal for reports and articles from these SIG leaders. We are looking for people interested in using the NeXT for Medicine, for Music, for Desktop-Publishing and for Courseware. If you think you might be able to run such a group or are interested merely in being put on the mailing lists, please contact Conrad at the electronic address above.

As I mentioned last month, we will be starting the *Government Watch*, column. I have already received information from ESL and Compus, but am waiting for write-ups from Hughes and Jim Greenwood of ADS.

I should note that I just visited Maryland and stopped by to see the Boys (and one Girl) at Lighthouse Design. We had a lovely evening and they fed me white pizza! I'd like to thank Roger & Jonathan for such a nice time. You can read about some of their new product development in the *Myopic Eye* section at the back of this issue.

Anyway, things have been a bit hectic around here and this issue is a few days late. Hope it was worth the wait. -- Erica

HEADLINE NEWS

• **NeXT** Announces 68040 Cube.

• **Wingz** introduced at BCS meeting. ***(Informix)***

• ***ImagineInc*** is now shipping **Media Station**

• ***MetaResearch*** is shipping **Digital Eyes.**

• Keep your eyes open for the New **On-Campus** Newsletter from *Patty Kammerer* at NeXT

Copyright 1990 by BuzzNUG Individual Articles Copyright 1990 by their Authors. This issue of NUJ may be freely copied and distributed but not altered. Authors are free to resubmit articles for publication in other media. This issue of NUJ may not be sold for profit.

Next Issue Highlights

• Government Watch - profiles of Government Contractors using the NeXT.

• Minimum Spanning Trees- another adventure in Graphics Education

• A discussion of MUMPS for the NeXT done by Plus Five Computers

• A Strip Chart recorder by Pat McGee

• Something nice by Jiro

• C-threads, we hope, by Dave Jeorg

• Dave Stutz may do an article on communicating with servers from within Interface Builder (for example the SYBASE server)

INDEX

Welcome • 1
Headline News/Next Issue Highlights• 1
Editorial Stuff/Feedback from the Trenche • 2
BaNG's First Meeting • 5
Speaker/Listener • 6
HP Hard Disk Installation • 12
MultiMedia Project Announcement • 16
Programming in Lisp on the NeXT • 17
Who's Calling Review/Paper Sight • 22
The **STONE AGE** • 27
NeXT Announces the 68040 • 30
The <c-S,Q,@> Wars • 30
What's Up Dock? • 32
Reviews, Rumors & Stuff (The Myopic Eye) •35
Market View • 36
User Groups • 38
Buzz's Hint Corner • 40

Contributors:

Erica J. Liebman,Patrick Haener, Joe Barello, David "the man" S. Joerg, Carl Erickson, Conrad Geiger, John S. Karabaic, Terrence J. Talbot, Andrew "Monolith" Stone, Dick Silbar, Joel McClung, Skip Montanaro, Doug Kieslar

•Editorial Matters•
1150 Collier Road NW/L-12
Atlanta, GA
30318

•Editorial Stuff

Articles for Buzzings are accepted in various forms, NeXT mail enclosures and Internet .wn.tar.Z forms are preferred but ascii text via the net, IBM and Mac Disks via USMail and (yikes) written text via the same USMail are happily accepted. We can guarantee no return of materials without SASEs, sorry. Our focus is how-to articles, especially with sample code. All articles are subject to editorial review.

We also welcome copies of new (and old) software for review from third party vendors. Again, we can not guarantee material return without SASE or guarantee publication dates, if at all, although we try to be prompt.

"Feedback from the Trenches" is open for comments/ letters of limited length from all readers. Please write and tell us what you liked and what you disliked.

Mailed subscriptions should start real soon now, with any luck. Please write for information.

Feature Submission Guidelines

If you have NeXT mail, simply drag your icon into next mail and post it off to me. Don't worry about tar'ing and compressing.

•Avoid passive voice. Please.

•Spellcheck.

•Wit is welcome, overwhelming cuteness and obscenity are not.

•Preformatting with the following guides will aid us ENORMOUSLY:

•Paragraphs: No tabs, single return at end of paragraph, no hard-returns at ends of lines

•Title on one line, author information on one line.

•Avoid blank lines between paragraphs, but use blank lines to denote sections.

•Start sections on first line after section header.

Editorial matters to:

BuzzNUG c/o EJ Liebman

1150 Collier Rd./NW L-12

Atlanta, GA 30318.

1-404-352-5551.

There is always an answering machine, but please respect relatively normal hours. Long distance phone calls may not be returned by the impoverished student at the other end -- if you leave a message, indicate if it is acceptable to call you collect. Please send any deliveries of items that will not fit within a tiny mailbox care of the Leasing Office. To contact me directly for subscription information, corrections, requests, or just to say hi, write via internet: erica@kong.gatech.edu

This issue of NeXT Users' Journal was prepared in FrameMaker, generously donated by Frame..

FEEDBACK FROM THE TRENCHES

••At our University we had the opportunity to evaluate one cube duringtwo weeks. The NeXT was provided by the Workstation AG and the people there told me, that by now they sell the cube to anybody.As far as I know the cube is sold in Europe by the following companies:

SwitzerlandWorkstation AG/(contact Mr. Gartmann)/ CH-8302 Kloten (cube with 330MB HD/29,000 SFR, laserprinter/ 7,000 SFR, MO disk/ 150 SFR, educational discount:/30%)

France Qualigraf/Nanterre, (cube with 40MB HD/ 77,950 FF, 330MB/107,800 FF, 660MB/122,450 FF, 4MB memory/22,844 FF, laserprinter/29,950 FF, MO disk/985 FF)

UK/Businessland (I read, that NeXT has opened an office in the UK. Do you know anything about it?) (It is however not clear to me, if these are official resellers) Have you heard about other peolple in Europe interested in the cube? I'd be glad to contact them. I hope this information helps you, and I'll inform you as soon as I have any news.

Greetings from chocolate country, *Patrick Haener*

• •I just thought you might like to know that I FINALLY figured out what I was doing wrong when trying your "Guesser" example from the first issue of Buzzings. I was originally unable to get it to work, and after checking my typing and going through the thing several times without success, I gave up. Lately, I decided to have another look at it. Sure enough, it was something "obvious."

I wasn't using the full pathname for the sound files, so theprogram couldn't find them. I went in and added the full pathnames and PRESTO! it works!

Thanks for the programming example. This is the ssort of thing that novices like myself find helpful.-*David Carpenter*

••Hello! I just ftp'd Buzzings #5 from the server at j.cc.purdue.edu. A couple of comments:

1) This issue was as terrific in content as all the others. You are doing a splendid job with the newsletter. I am temporarily cut off from access to netnews and so Buzzings is my most important source of essential NeXT-related information. (Buzzings articles always go more in depth than notes in comp.sys.next, too.) My only fear is that the enormous amount of work you must be putting in will burn you out. Keep up the good work!

2) About the format: I hate to have to say it, but it was awful. Let me temper that somewhat by explaining the context: I felt that the single best thing about the newsletter was that I could read it on-line. I have several times had an article from Buzzings open on the screen next to a Shell or Interface Builder, trying out things as I read about them, and learning at a great rate. But Frame is slower even than WriteNow, and the two-column format with tiny type makes it necessary to scroll up and down, up and down in an enlarged view to read each page. Now I know that you want to improve the appearance so that you can start publishing Buzzings in a paper format, but PLEASE stick to one column so that the issues can still be browsed on-line, and PLEASE if you're going to compose in FrameMaker, then distribute the finished issues not as Frame documents but as PostScript that can be browsed page-by-page in Preview (which is much faster -- after all, we who read the issues don't really need to be able to edit them). Check out the Adobe documentation in /NextLibrary/Documentation/Adobe -- it is all organized this way and works fine.

I hope that didn't sound too crabby. I really liked issue #5, it just wasn't as convenient to read as the previous issues were. I am (as always) eagerly looking forward to the next issue. *Cameron Smith* CAMERON@MID-D.BITNET.

••Harumph!!! How much of that nonsense in the April issue about color displays, color printer, one gig disks, etc. was true? At first you had me reading it aloud to everyone in the lab, but eventually it just got too bogus for words. We decided that it was an April fool's article in it's entirety. At first I was mad, now I am just sad. I wish you hadn't done it. Oh well, I'll live :-) -- B*rian Bartholomew* ,bb@pisces.ee.ufl.edu

••BTW: Is the "TECHNOLOGY WATCH" in Buzzings5 an April fool's joke? It all sounds a bit too much like rumors or fantasies. On the other hand, I didn't see anything hinting that it was, in fact, a joke.

I can't believe Dave Joerg wrote his "Technology Watch" without some sort of annotation at the end saying it was bogus -- for us gullible types of course.*Daniel L. Green*, danielg@cs.umr.edu

••Is the TECHNOLOGY WATCH by Dave Joerg (April issue #5) the sickest April Fools joke i've ever heard of or the best news i've ever heard of? I couldn't find any hint of joking except for the completely unbelievable giving spirit of Steve. What goes?? *Dave Hieb*, davehieb@boul der.Colorado.EDU

••Before I post something to comp.sys.next, (like Do I live under a rock?) I wanted to make sure that the "Technology Watch" column was on the up-and-up. I find it strange that the news everone has been waiting for is buried on page 16! Most of it SEEMS plausible, but the bit about "Malaysia, Libya, Finland and USSR" sales is a little far-fetched. Also the bit about the blue-green algae drive is about as wacky as it can get.

If this was a bit of humourous fiction on Dave's part, it should have been clearly marked as such at the end of the article. If it's true, why haven't we heard such information from other sources (NeXT for one....)

I like the new format of the journal and anxiously wait each and every edition. I might get the guts to write a small article on the results of my Master's project - a NeXT-based Answering Machine - when it's done in mid-July. Keep up the great work. *Mike Brown*, Gradual Student in Computer Science, University of Western Ontario, brown@csd.uwo.ca

Most people started catching on by the passive-aggressive updating and were giggling when they hit the floating algae-- EJL.

••I apologize if you've already received a letter to this effect. I used to REALLY find Buzzings useful as a reference. Just pop it in /LocalLibrary, update the index, and whenever I was confused about a NeXT feature, often Buzzings would have some bits of sage advice about it. Neither the *.ps or *.frame formats are particularly Digital-Librarian friendly, and I feel bad.

Obviously, after spending substantial time and effort learning how to use FrameMaker, you would be unwilling to give up this graciously-given power-tool. I like Buzzings' format better now, too. So, "What is to be done?" Maybe FrameMaker offers you the ability to save formatted Frame stuff as plain ASCII. You could then distribute the ASCII alongside the *.ps and *.frame stuff for those who desire referencability [sic]. Or the articles (each a separate file) could be distributed in plain ASCII alongside, etc. Or perhaps this is not a big issue for the majority of NeXTians, in which case you shouldn't bother. *Dave Joerg*

••I tried to print Buzz 5 (both the .ps from Preview and the .frame from the old frame demo) NEITHER WORKED! If the thing won't print on the NeXT without the new version of frame which we can't afford,

then I am disappointed.

Why did you abandon the WriteNow Format??? We can put that into the Librarian as far as I know this won't work with the Frame document. And since I can't print the thing, I am stuck reading it off the screen at a lower resolution than it really requires with your innumerable fonts. I'm sure that using framemaker is fun and everything, but have some compassion on us that can't afford to get it.

I do really want BuzzNug (or whatever you are going to call it) to succeed and I will write an article as soon as I learn something that would be useful to others.-*Kelly Anderson*

Decisions about changing from WriteNow to FrameMaker were not done lightly. However, options for access to Digital Librarian and Write-Now compatable versions will be investigated. -- EJL

••Pehaps I missed something, but when I went to get Buzzings #5, I found only PostScript and Frame versions. I have been using the WN format Buzzings and cutting and pasting article, especially the programming tutorials, into a single file so I can work through those things that are of interest to me without all the other unrelated stuff getting in the way. Since we don't have Frame(maker?), I have to rely on Preview to read and it's just not as nice! Just my two cents worth! Keep up the good work down there! Eric J. Isaacson (the other) isaacso@silver.ucs.indiana.edu

You can use your freebie Frame sample program to read and cut/copy/paste from your copy of the newsletter -- EJL. Eric Replies : "Thankfully, when I don't realize the obvious, it means there IS an obvious, and the solution to my problem is cheap. Thanks! Eric"

••Has there been any interest in NeXT sigs (special interest groups) in Buzznug? I have two groups that I could help with. I already have a pretty extensive list of interested parties that would like a NeXT MEDICAL user group. These people are from all over the country. I have about 10 just from the University of Washington here in Seattle (pathology, radiology, medical library, anatomy, etc). The other SIG group that I though of was a COURSEWARE SIG. Do you have any more that you can think of? Do you know of anyone that could help out with the SIGs if we get them started? Conrad Geiger - se seattle NeXT

At the time of "going-to-print", the BuzzNUG Medical SIG has been started - please contact Conrad_Geiger@next.com for information. Additional SIGs we hope to start are CourseWareSIG, DesktopPublishingSIG and MusicSIG. We need volunteers to help coordinate these special interest groups who will be responsible for a monthly column in the NeXT Users' Journal, coordinating special mailing lists and working with the archive sites - EJL

••Dear Erica:

Thanks for responding to my email about the format of Buzzings #5. talking about.
I'm afraid I didn't explain myself very well in my previous note. You asked whether I had tried reading the issue at 150% magnification. Yes, I did; that's what led to the problem with the two-column format.

You see, when you magnify the page enough to make the letters legible, then it is big enough that the top and bottom of a column won't fit on the screen at the same time. This means that if you are reading down a column, you have to scroll the window down. When you get to the bottom of the left column, you have to scroll back up to the top to continue reading in the right column. This means that you have to scroll up-and-down over every page at least twice.

It's worse if you are reading a tutorial article and trying it out as you read. If you are trying to learn something new you may have to read the same passage again and again to figure it out. That's a real pain if the key passage spans the break from bottom-left to top-right.

I am writing this note in two columns so that you can see what I am I hope that by now
I have written enough text to make each column longer than the window you are reading this in. If so, you will see what I mean about scrolling. Now multiply that by a 40-page document.

My point is, two narrow columns may be easier to read on paper than one wide column, and they may make the design more elegant, but for text that you want to be able to read on the screen, one column is really better. That way you minimize the amount of backwards scrolling that has to be done.

Again, as I said, I have no complaints about the content of the newsletter -- it's great. I would just like to request that it be laid out in a format that encourages, rather than penalizes, reading on the screen. There aren't many computers with a display as sharp and attractive as the one on the NeXT; it would be better to take advantage of that than to force people to use up paper printing a document that is more useful in electronic form anyway.

Thanks again for your time and attention. --*Cameron Smith*

••Extremely professional job, no wonder I didn't hear from you. I was a bit startled when I saw the new .frame.Z suffix when ftp'ing Numero Cinco, being a solid supporter of free software, I had never used frame, fearing that clicking the license button would enable some thought police software. Nothing bad happened, but BuzzingS looks 'marvelous'. And ^Z's article fascinating, I have always been a fan and supporter of his excellent work [A gigabyte, Erica, and

20ns drives, !] Special thanks for 'splash appeal'! Everybit helps when you practice guerilla marketing. I ran the numbers and they do not look too encouraging: I think our nation should just support programmers such as us, so we can continue to provide freeware, instead of having to sell commercial product, which is a serious hassle and impediment to the fun things I would rather be designing. I was inspired by Jiro's article on *ware, sleazeware, snakeware... Free association - *Andrew "The Polylith" Stone*

••When I saw the name "Buzzings5.frame.tar.Z" (why tar?), my first reaction was "Oh, no..." But the more I read it, the more I liked it. Hint for people who read it in Frame: use 200% magnification. Hint for people who read it in the demo version of Frame that was shipped with the 1.0 release: launch it before you go to bed, it will finish launching by the time you return in the morning. (Is the official version any quicker?)

Dave Joerg (if that's your real name:-) -- good job! I wasn't convinced that the "Technology Watch" thing was April Fool Fodder until I hit the bit on foreign sales. J*acob Gore* <jacob@gore.com>

••Buzzings 5 was quite impressive! Didn't have a printer handy so I couldn't read most of the fine print, though. Pulling interesting quotes out of the text and putting them big-fonted smack-dab in a column gives it a very "professional" touch. Touches make the image. I shuddered with fear upon seeing my article in there -- just imagine the amazing flames I will receive! -- *Dave Joerg*

••I have looked at part of the new BUZZNUG. I have seen pages 22-the end so far and I like the Frame format. You can get lots of info on less than 35 pages!

I want to correct a rumor that was in the rumor column...that Conrad Geiger was promoted to Director of Western Sales. I have changed positions, but my new title is "Western Sales Development Engineer". Thanks for the BIG promotion though. *Conrad Geiger*
P.S. People really love Bernie the Duck!

Mea culpa! I dropped two words! Congrats anyway.-- EJL

BaNG's FIRST MEETING

Joe Barello

Stanford campus, the grounds still green, Braun Music Center, a perfect place to hold a NeXT meeting featuring "Sound and Music using the DSP".

In the 150 seat room approximately 180 people milled about. Not too bad for a first meeting, don't you think. Robert Poor and Julius Smith prepared to talk about DSP related topics. 4 NeXT machines were along the sides of the room with third party developers and their products. 25 to 30 NeXT employees were hanging out. There was marketing, sales, support and programmers all from NeXT! Here was an opportunity for users and programmers to connect with NeXT employees face to face in an informal manner .

The evening went like this. The first hour was spent munching cheese and fruit while people played with software, looked at brochures, hob nobbed with each other, and generally looked on in anticipation to what was to come

The meeting started. Robert Nielsen, the man to organize such an event, started off with a hello and an introduction. Robert Poor talked about and demonstrated music produced on the NeXT. He showed the Ariel Digital Microphone and the MetaResearch Digital Ears hardware products that were used to input sampled music and sound. The sound demos were played back with MetaResearch's SoundWorks software. He played some experimental music from Carnegie Mellon School of Music and some sound tracks generated for MTV. He took small 2 second "chunks" of sound and built up longer series of sound by editing the sound "object" with copy and paste. This was a great example of intuitive direct manipulation. Your are on a computer, here is this sinusoidal wave on the screen that represents the sound you are hearing. Click and drag to select the area of interest. Copy - Paste, Paste, Paste, Paste, - Play. Weeeeeeeee! That was fun! To end his talk, Mr. Poor described his wish list of applications that should get built for sound and DSP. I have conveniently forgot what they were as one or two might be on my future development list.

Julius Smith, an outgoing DSP guru, talked about neat developments on NeXT DSP software projects across the country. H mentioned the CMU Sphinx speech recognition software, some text to speech software being developed in Calgary, musical instrument modeling, hardware projects like multiple DSPs on a board, and the work done locally at Stanford on voice and sound analysis. As I remember Robert Poor took classes from Julius while at Stanford and learned most of what he knows about Digital Signal Processing from him.

After the "formal" presentations, someone from Portal Communications in Cupertino got up and offered mail only UUCP connections for $15 a month. You need a 2400 baud modem and whammo, you have NeXT mail. Attachments, sound, the whole smeer.

Then, two third-party vendors demonstrated products.

Had I mentioned the large screen projector that was used for demonstrating NeXT programs? It's bright and big. There were other third-party vendors and products on regular NeXT machines, but "TopDraw" and "Who's Calling" were the featured demonstrations that day.

"TopDraw": (by MediaLogic) Take the best features of well known McDrawing packages like McDraw II, CricketDraw, McDraft; add about 80 percent of the Postscript operations that were not simulated in CricketDraw, add a few more things and you have a great object oriented (object scrunch, shade, rotate..) drawing package.

"Who's Calling": Adamation is a little company who is doing custom NeXT software. The William Morris Agency decided to buy in big-time with NeXT machines because "Who's Calling", was the application they needed for their personnel. To date, I believe that the William Morris Agency has 250 NeXT machines. GO ADAMATION! Well, "Who's Calling" is an integrated calendar, resource tracker, memo, and reminder system done in Interface Builder and NeXTStep with an underlying INGRES database. I liked the demo, especially when the calendar reminder system announced in a nice female voice of an appointment

Of the other third-party products: HSD demonstrated their two scanner hardware products. Stain Glass Software demonstrated their Calendar product. There was a physical security NeXT locking system (bar and lock). There was a beta sheet music generator, editor, player system.

Robert Nielsen closed the day with an open forum for comments and a group talk about future meetings.

In all it was a great first meeting. There will be more to come.

Comments, questions and concerns should be addressed to:

BaNG c/o

Robert D. Nielsen

PO Box 90460 San Jose, CA 95109

nielsen@everest.portal.com 415 780 2884

Joe Barello Lawrence Livermore National Laboratory

joeba@lll-lcc.llnl.gov

SPEAKER/LISTENER:

Why it's cool, How to use it

David S. Joerg

(joerg@alliant.mcs.anl.gov)

MindShock, Inc.

Introduction

The Appkit provides programmers with a pair of wonderful classes that let applications effortlessly communicate with each other: Speaker and Listener. Because NextStep applications can talk to each other, each application can concentrate on doing one class of things very well, and depend on other applications to provide other necessary services. This article provides general principles for using Speaker/Listener, and some simple examples. A few of the finer points and some troubleshooting suggestions are hidden away at the end. Note that a great wealth of information on this complex topic is in the class descriptions for Speaker and Listener, as well as in the man page on msgwrap.

Disclaimer

I am not a guru of any sort. I didn't write the Speaker/Listener system, and I know next to nothing about Mach. Everything I know about this stuff is from repeatedly reading everything I could find on the topic, and from using the Speaker/Listener system in my own applications. If I tell you something, and you try it, and the black box says "Dave's wrong", then I'm wrong! Blame my mother. When I was very young, she used to warn me, "Davy, you'll never understand low-level system software, never, never, never...." Prophecies have a way of self-fulfilling, you know.

Why would I want to use Speaker/Listener?

First of all, your application may need to communicate with the bundled apps: Librarian, Webster, Quotations, etc. This is not hard to do; see the examples.

Secondly, your application need not consist of a single executable; many applications profit from separation of the "front-end" and "kernel" variety. Speaker/Listener helps here by handling the messy Mach stuff for you -- all that's left for you is a simple method call. Fun!

More food for thought: NextStep, and Interface Builder in particular, inspires people to design their own applications. By providing Speaker/Listener connectivity, you allow users to write their own front ends, with their own customized code, and hook it in to your application, all without headaches. No recompilation is required by this scheme; you are not forced to give source code away, and your application can enforce rigorous validity checking on the data coming from the user's front end.

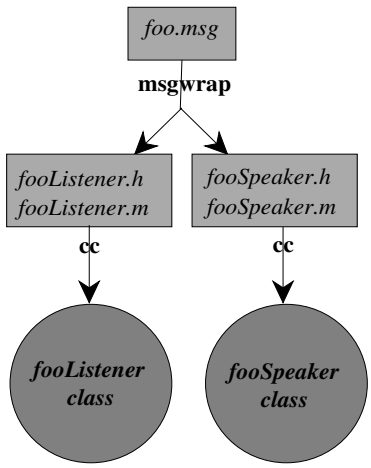
Lastly, consider this: if humans have a need for your application's capabilities, maybe other applications do too! By providing a public Listener, and documenting the messages it expects, you help extend the usefulness, and ultimately the indispensability, of your application.

Simple summary

The Appkit's standard Speaker and Listener already know about a wide range of messages pertaining primarily to Workspace communication and pasteboard manipulation. You can create your own subclasses of Speaker and Listener automatically by feeding a foo.msg file to the msgwrap utility (see Figure 1), which will create the source code for fooSpeaker.[hm] and fooListener.[hm]. Speakers and Listeners are created in pairs and must agree, at msgwrap-time, upon which messages they will use.

The application that will be able to receive messages defined in foo.msg should instantiate a fooListener, and register it under some name (name registration is discussed later). Any application that wants to send messages to fooApp must instantiate a fooSpeaker, "aim" the fooSpeaker at fooApp's fooListener, and "fire" the message (see Figure 2). The fooListener will call the appropriate method of any object in fooApp, whomever fooListener is set to deliver messages to.

If that sounds too easy to be true... you're right! There are a few tricky things to remember when using Speaker/Listener, but once you know the pitfalls, it's generally smooth sailing.



Example: Messaging Bundled Apps

This example is ruthlessly stolen from NeXT Developer Camp's Speaker/Listener materials, and then reworked a bit for more generality. To make this method come to vibrant life:

• Make a new project in IB.

• Create a subclass of Object (we'll call it blah). Give blah one action (sendMessage:) and two outlets (theWordField and appMatrix). Unparse blah. Instantiate blah (to get blahInstance).

• Create a button titled "Send Message". Set its target to blahInstance, with an action of sendMessage: .

• Create a textfield, set blahInstance's theWordField outlet to it.

• Create a matrix of radio buttons, marked "Webster", "Librarian", "Quotations", "Preview", "Edit", "WriteNow", etc. Set blahInstance's appMatrix outlet to it. Be sure that the titles on the radio buttons are EXACTLY (case-sensitively) the same as the names of the apps you want messages sent to.

• Go into blah.m and put in the code for sendMessage:

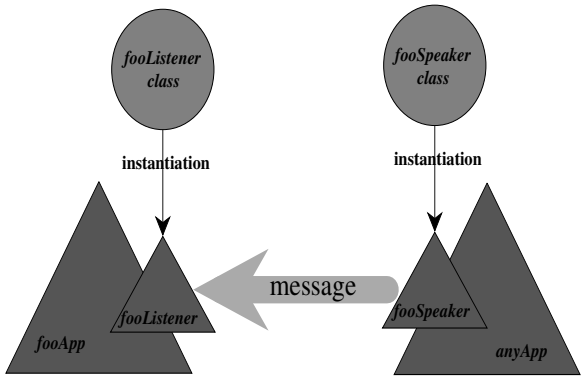
• Make and execute!

```
- sendMessage: sender
{
  int speakerResult, ignored; port_t appPort;


  /* look up the public port for app's Listener on local host */
  appPort =
     NXPortFromName([[appMatrix selectedCell] title], NULL);
  if (appPort == PORT_NULL) {
    fprintf(stderr, "Port for %s was not found.\n",
      [[appMatrix selectedCell] title]);
    return self;
  }


  /* connect the port to the Application's Speaker */
  [[NXApp appSpeaker] setSendPort: appPort];


  /*
   * use the openFile:ok: method (Webster, Quotations,
   * Librarian use this for lookup requests. The others
   * use it for file-opening requests)
   */
  speakerResult = [[NXApp appSpeaker]
    openFile: [theWordField stringValue]
    ok: &ignored];


  if (speakerResult != 0)
    fprintf(stderr, "Message failed to get through.\n");
  return self;
}
```

Customizing the code generated by msgwrap

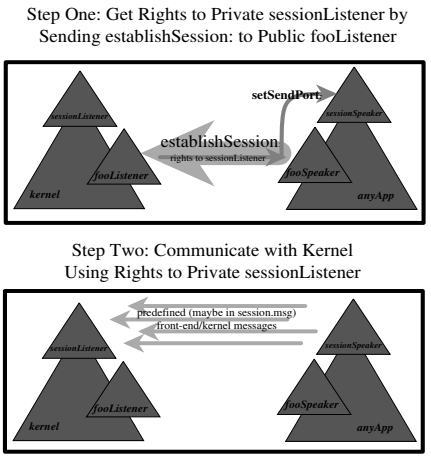
DON'T!!!! Just don't do it. You should never touch the code generated by msgwrap; just put it in your project, and make sure to #import the header file in the right places.

If you need to attach extra state data to a Listener, don't change the Listener code. Instead, create a custom object that contains the state data, and set this custom object as your Listener's delegate. Then, your message-responding methods can be in your custom object, or your custom object's delegate (your subclass of Application, or wherever your "nerve center" code is).



Names, names, who's got the names?

The net nameserver has the names. When a Listener is checked in (see next section), it can, optionally, have its port registered by name with the nameserver. When an application like Webster is launched, one of the first things it does is register the name "Webster" with the nameserver. Then, when other applications look for "Webster", pow! there it is. That's what NXPortFromName() does. The careful reader notices that there are two arguments to NXPortFromName(), and that the second is set to NULL in the above example. Well, the second argument is a hostname! That's right, communicating with applications on other hosts is as easy as knowing the name of the host they're running on! Setting hostname to NULL indicates a local name search. Another very nice feature of NXPortFromName() is that it will automatically launch the named app for you if it isn't running already, and hostname is NULL, and it's an application recognized by the Workspace (found by "Find Applications...").



It is highly recommended that you use setSendPort: before every attempt to send a message, because the application you are messaging to might have quit or been killed or otherwise been rendered unable to respond.

Listener's Delegate's Delegate's Dele.....

Setting up your own Listener is a bit more involved (and fraught with choices) than simply messaging to someone else's Listener. The easiest way to create your own Listener is to put this little bit of code in your main program, right before the call to [NXApp run]:

```
[NXApp setAppListener:[someListener new]];
```

Before your application's main event loop begins, if there is no appListener already set, a standard Listener will be created and designated the appListener. If a name has not yet been chosen with setAppName:, your application's name will used as the check-in name automatically.

By not registering with the nameserver you can pick and choose who gets to send messages to your listenPort

If you'd rather not put your own code into the main program because Interface Builder maintains (and thus overwrites) the main program file, the same effect can be achieved by putting the setAppListener: and setAppName: calls at the very beginning of your application's run method.

Of course, applications can have more than one Listen-

er. To create your own Listener, you must:

• Create the Listener with [someListener new].

• Check the Listener in under some name. If someone already has the name you're trying to check in under, the check-in fails (see "Example: avoiding name collision"). Name check-in is optional, as sometimes you need to create ports that are not public-access. By not registering with the nameserver you can pick and choose who gets to send messages to your listenPort (see "Port rights: what and why"). If you choose not to check in, don't forget to call privatePort, which creates a port but does not tell the nameserver about it.

• Set the Listener's delegate. When the Listener receives messages, it will try to call the corresponding method of its delegate's delegate or its delegate. What this means is that if you want the messages to be received by some "custom object", you can designate the Listener's delegate as NXApp and NXApp's delegate as your "custom object" and your "custom object" will still receive the message. If your message-receiving code is in a subclass of Application, this is just fine as well.

• Call the Listener's addPort method. When this is done, the application will check the Listener's message queue whenever it looks for events, and call the proper Listener methods whenever there is something in the queue, depending on the relative priorities of the Listener and the current threshold (see "Timeout mania").

Example: avoiding name collision

Most applications never need to worry about this. It's not likely that two independently developed applications will have the same name. However, if your application is the kind that gets launched multiple times on the same host (like Terminal) and you want them all to have active Listeners, here's what you might do:

This example assumes that registeredNameField is an outlet to a textField, and that another textField's target is connected to the changeRegisteredName: method.

These methods need to be in a subclass of Application, or in NXApp's delegate.

```
- appDidInit:sender
{
  [registeredNameField setStringValue:[[NXApp appListener]
     portName];
  [[NXApp mainMenu] setTitle:[[NXApp appListener] portName]];
  return self;
}


- changeRegisteredName:sender
{
  [[NXApp appListener] checkOut];
  if ([[NXApp appListener] checkInAs:[sender stringValue]])
  {
    NXRunAlertPanel("Error", "Sorry, there's something wrong
with that name.  Probably someone else already has it.", "OK",
NULL, NULL);
    if ([[NXApp appListener]
          checkInAs:[registeredNameField stringValue]])
    /* we can't get our own name back! */
    {
     NXRunAlertPanel("Error", "Wow! Someone grabbed the name I
was registered under while I was trying to get a new one!",
"Sure", NULL, NULL);
      [registeredNameField setStringValue:"Help!!!"];
    }
  }
  else /* we got the name just fine */
  {
    [registeredNameField setStringValue:[sender stringValue]];
    [[NXApp appListener] addPort];
  }
  [[NXApp mainMenu] setTitle:[pubListener portName]];
  return self;
}
```

Those pesky return values

Let's say you want your app, SimpleApp, to be able to receive (and send) simple character strings. You might write Simple.msg to look something like this:

```
- sendString:(char *)theLetters;
```

And that would work just fine. Then you would make your method declaration in SimpleApp.m:

```
- sendString:(char *)theLetters;
```

And nothing would work. Why? Even though your Simple.msg file declares no return type for the sendString: message, the actual method must be declared to return (int). What's more, the integer that sendString: returns will always be ignored and will not be passed back to the caller; instead, an error code (0 for success) is always returned. The correct declaration (in SimpleApp.m) is:

```
- (int)sendString:(char *)theLetters;
```

The careful reader may ask: "Well, then, how do I return values back to the caller, e.g. my own error codes, or responses to queries?" You must establish parameters that are pointers to the data you want returned. (valid types for sending and returning discussed later)

For instance, to extend the sendString: example, perhaps SimpleApp would be expected to return the Italian

equivalent of whatever it was sent. The new Simple.msg file would look like this:

```
- sendString:(char *)englishText translation:
            (char **)italianText;
```

And the new SimpleApp.m method declaration would look like this:

```
- (int)sendString:(char *)englishText transla-
        tion:(char **)italianText;
```

A final warning about return values: although the integer that your method returns is ignored, it must return something. Failure to do so will cause the Speaker/Listener system to flag an error. This is hard to track down if you don't know you're looking for it.

Why can't I send (or receive) float's?

The only types you can send are (int), (double), (char *), (port_t_send), (port_t_rcv), and (byte *). What? You've never heard of (byte *)? You thought there was only (port_t)? Welcome to the wonderful world of msgwrap. If you're sending a parameter the same size as an (int) or (double), you can typecast it to (int) or (double) and send it. If you're sending a zero-terminated chunk of data, you can send it as (char *). If you're sending a chunk of data that is not zero-terminated, or has zeros in the middle, you need to use (byte *). For every (byte *) that you declare, you must declare an (int) to follow it, that says how long the byte array is. Note that your actual code should still deal in (char *)'s or (void *)'s; the (byte *) is a fictional construct that only msgwrap understands or needs. Sending and receiving port rights are discussed in "Port rights: what and why".

Receiving data is also a lot of fun. The valid return types are one reference level up from valid send types: (int *), (double *), (char **), (port_t_send *), (port_t_rcv *), and (byte **). Yes, every (byte **) must be followed by a (int *). Remember to only manipulate what the return parameter points to, and not the return parameter itself (this was a conceptual stumbling block for me, coming from Pascal).

The Speaker/Listener system automatically handles the copying of zero-terminated strings and byte-arrays from the caller's address space to the callee's, BUT that data is not guaranteed to stick around for very long, so malloc() some space and copy it as soon as you get it. If you're receiving large character strings or byte arrays (more than 1K) you may want to use NXCopyInputData(). It will vm_allocate() the memory needed to copy the input parameters for you. Be sure to use vm_deallocate() to free space allocated by NXCopyInputData(), not free().

Example: Sending and Receiving Byte Arrays

Since no widely available apps use byte arrays in their message protocol, examples of byte-array transfer are not as dramatic as messaging the bundled apps. In this example, the only message known to BitmapSpeaker and BitmapListener is demandAppBitmap:length:, which gets the image and alpha data (concatenated) for the called application's icon. This example assumes that the application's icon is a 48-by-48 bitmap with 2 bits per sample, and 2 samples per pixel, one for gray, and one for alpha.

The Bitmap.msg file might look like this:

```
- demandAppBitmap:(byte **)bitmapData length:(int *)len;
```

The calling application might have a method like this:

```
- getAppBitmapFromApp:(char *)app onHost:(char *)hostname
/*
 * this method gets the icon for application app on host hostname
 * and registers it as the bitmap app. Also returns the bitmap's
 * id. Returns NULL on failure, with error message on stderr.
 */
{
  id appBitmap; /* what we return */
  int speakerResult, ignored;
  port_t appPort;
  void *bitmapData;

  /* look up the public port for app's Listener on local host */
  appPort = NXPortFromName(app, hostname);
  if (appPort == PORT_NULL) {
    fprintf(stderr, "Port for %s on host %s was not found.\n",
app, hostname);
    return NULL;
  }

  /* connect the port to the Application's Speaker */
  [[NXApp appSpeaker] setSendPort: appPort];

  speakerResult = [[NXApp appSpeaker] demandAppBitmap:
     &bitmapData length:&ignored];

  if (speakerResult != 0)
    fprintf(stderr, "Message failed to get through.\n");
  else if (bitmapData == NULL)
    fprintf(stderr, "Invalid data returned.\n");
  else
    /*
     * note that we don't bother to allocate space for the
     * returned byte array because addName:data:width:
     * height:bps:spp: copies the data for us.
```

```
     */
    appBitmap = [Bitmap addName:app data:bitmapData
     width:48 height:48 bps:2 spp:2];

  return appBitmap;
}
```

The called application might have a methods like this:

```
 These methods need to go in a subclass of Application to work
properly.
- run
{
  [self setAppListener:[BitmapListener new]];
  return [super run];
}

- demandAppBitmap:(char **)bitmapData length:(int *)len
{
  id appBitmap;

  *bitmapData = malloc(2*2*48*48);
  /*
   * if we use IB's make, the machO __ICON segment will have
   * our application's icon under the name "app".
   * findBitmapFor: will find it there
   */
   appBitmap = [Bitmap findBitmapFor:"app"];
  [appBitmap readImage:*bitmapData withAlpha:*(bitmapData
     + 2*48*48)];
  [appBitmap free];
  return 0;
}
```

Port rights: what and why

If you want to let anyone send messages to your Listener, you need not worry about port rights. Just remember to check in under some name, and everyone who knows that name will be able to send messages to your Listener. If you don't want to let all applications send messages to your Listener, just don't check in. Easy! Don't forget to call privatePort though.

Unless you set them otherwise, Speakers are created with a sendTimeout and replyTimeout of 30000 milliseconds

Some applications, particularly those that work in a "front end" and "kernel" pair, definitely should NOT leave their important Listeners available to anyone. The sticky part, naturally, is granting port send rights to only the applications you want to have port send rights. The best way (IMHO) is to have the kernel's appListener recognize a message like establishSession:, which will make the kernel give a private Listener's port send rights to the caller, if a session is not already in progress. Or, if your kernel will support multiple simultaneous sessions (that maintate state data that persists from call to call), you could instantiate a new private Listener every time an establishSession: message came in, and have that Listener's delegate keep the state data.

When giving or taking port send rights, make sure that your foo.msg file declares the parameter as (port_t_send) or (port_t_send *). If you want to exchange port receive rights (although I can't figure out why you'd want to) you should use (port_t_rcv) and (port_t_rcv *).

Apparently, it is important to port_deallocate() received port rights once you are done using the received port. I would appreciate a Mach guru's comments on why this is necessary, what it does, etc.

Timeout mania

Unless you set them otherwise, Speakers are created with a sendTimeout and replyTimeout of 30000 milliseconds. This means that if the message sent from a Speaker cannot arrive at the Listener's doorstep within 30 seconds (sendTimeout), the message fails. Then, if the message arrives successfully, and the Listener cannot return whatever data she is obliged to return within another 30 seconds (replyTimeout), the message fails.

Listeners have a timeout too, which also defaults to 30000 milliseconds. Apparently this timeout does the same thing as the Speaker's replyTimeout, but from the Listener's side of the connection. That is, if the method called by the Listener in response to a message does not return in timeout milliseconds, the Listener will return anyway, whether your application has the correct return values ready or not.

Usually this is nothing to worry about; 30 seconds is a long time, even for humans. However, this can become a problem when an application is doing something besides regular event processing (like sitting in a modal event loop). Sometimes the right solution is to not worry; sometimes the solution might be to place your Listener's priority on or above the level of NX_MODALRESPTHRESHOLD or something like that. Or just jack it up to 30, why take chances.

HP Hard Disk Installation & More

Carl Erickson

Michigan State University

erickson@frith.egr.msu.edu

uunet!frith!erickson

erickson@msuegr.bitnet

This article outlines my experiences with purchasing, and installing a Hewlett-Packard 97544S 324M SCSI disk. It may prove useful to others in three regards:

•installing an HP disk

•repartitioning an existing disk

•moving the swapfile

Some of the steps towards the end of this procedure are complicated and assume some basic proficiency in Unix. The consequences of making a mistake when following these procedures can include a cube that will not boot and lost hard disk data. I believe that resorting to the standard system files and BuildDisk app would always rescue someone in this unfortunate situation, but I can't be completely sure. In the NextAnswers database, NeXT explicitly warns against trying this sort of disk repartitioning procedure (April_90/sysadmin.290). If you are going to do this to an existing SCSI disk, then make very sure you have complete, up-to-date backups. I've recorded my experiences and tried to provide warnings about problems at crucial stages, but I certainly don't promise anything.

I found the following info in the NextAnswers database to be quite helpful (search for 'scsi'):

April_90/os.355

April_90/hardware.412

April_90/hardware.411

April_90/hardware.241

April_90/sysadmin.290

For those who are interested, I purchased this HP disk from an HP OEM retailer. The name of this entity for the 7 states in the Midwest is Cranel (800) 288-3475. If you call whatever is listed for HP in the phone book you will eventually be given the name of an HP OEM dealer. I paid $1661 + state tax + shipping for the drive in April of 1990. The performance is rated (conservatively, supposedly) at:

16.5 ms random average seek

3.5 ms track-to-track

7.5 ms rotational latency

1.5M asynchronous transfer

4.0M asynchronous transfer

The best part is that it comes with a 5 year warranty. Also included is a nice 80 page OEM support manual.

I started from an OD-with-40 machine, so if you're only interested in repartitioning an existing disk, you'll have to skip some stuff below. Major steps of the process:

```
1. Craft new /etc/disktab entry
2. Remove swapdisk entry from /etc/fstab of OD
3. Power down
4. Physically remove swapdisk
5. Reboot, verify cube runs from OD only
6. Power down
7. Install new HD
8. Reboot from OD
9. Gather info on new disk
10. Run /usr/etc/builddisk manually
11. Customize the new HD
```

1. Craft new /etc/disktab entry

I wanted 2 partitions on my HD so I could use dump/restore the way they were intended to be used; this means different dump schedules for static things like most of the utilities and dynamic things like my home directory and mail spool. Another problem this solves is that of not being able to dump an entire 330M disk to a single OD. Each of my two partitions is smaller than a single OD.

Since I used an HP disk that was already mentioned in /etc/disktab, making an entry for a 2-partition disk was simple - I simply copied what had been done for the 660M Maxtor (with a small exception) Here is the new 2-partition disktab entry:

```
 #
97544S-512-2p|HP 97544SU|HP 97544S 2p|Hewlett Packard 324MB w/
512 byte sectors, 2 part's:\
        :ty=fixed_rw_scsi:nc#1447:nt#8:ns#28:ss#1024:rm#3600:\
        :fp#160:bp#0:ng#0:gs#0:ga#0:ao#0:\
        :os=sdmach:z0#32:z1#96:ro=a:\
        :pa#0:sa#224000:ba#8192:fa#1024:-
ca#32:da#4096:ra#10:oa=time:\
                :ia:ta=4.3BSD:\
        :pb#224000:sb#99968:bb#8192:fb#1024:cb#32:db#4096:r-
b#10:ob=time:\
                :ib:tb=4.3BSD:
```

I picked the size of the 'a' partition based on my own

OD disk usage, minus dynamic things like swapfile, tmp, spool (uucp, uucppublic, mail) home directories. I had trimmed the OD down considerably following the suggestions in one of the NeXT docs. The 224000 blocks gives 216,559 kbytes of space which reduces to 194,903 kbytes when the standard 10% is taken into consideration. The number 224000 wastes 160 sectors (80k) on the end of the 'a' partition.

The 'b' partition represents the balance of the 323968 blocks of the disk. I have 96,703 kbytes in that partition, of which I can access 87,032 kbytes. Note that I left the number of bytes/inode (db) at 4096 for this partition, unlike the 'b' partition of the 660 Maxtor.

2. Remove swapdisk entry from /etc/fstab of OD

Comment out the entry for the swapdisk. This will cause the machine to use the swapfile found in /private/vm on the OD.

3. Power down

Logout first.

4. Physically remove the 40M swapdisk

Follow the procedure in the manual for opening the cube. Remove the central disk frame with two screws at the bottom. Slide it and the cpu board out a few inches, remove the ribbon cables from the cpu board and fully remove the disk frame from the cube. Be careful to pull the ribbon connector straight off, as you can bend pins otherwise. Remove the power supply lines and the SCSI ribbon cable from the disk. Remove the two screws on each side of the disk frame which hold the 40M accelerator drive in place. Remove the drive.

5. Reboot, verify cube runs from OD only

Close the cube and reboot it. If it runs you're fine, otherwise figure out why you can't run off the OD only.

6. Power down

7. Install new HD

Set the SCSI target address to 1. On the HPs, the manual shows which jumpers to use to do this:

```
jumper  in/out
--------------
5       out
6       out
7       in
```

For the other jumpers I made some educated guesses. The Quantum 40M drive had a jumper labeled "EP" which was in. I assume this means "Enable Parity" and thus put a jumper on number 3 of the HP. The other positions I left as they came (no jumpers in).

The HP has holes in the sides which line up with the holes in the disk frame. The screws for the 40M are even the right size. Slide the disk in from the front right side up (connectors on the bottom). Connect the power lines and the SCSI ribbon cable. Screw the disk in place. Put the cube back together (don't forget the fan power lead!).

8. Reboot from OD

Reboot the cube from the OD.

9. Gather info on new disk

Use the /etc/scsimodes program to find out about the new disk. As root (in all that follows I will assume that '# ' is your prompt) while you are su'd to root.

```
# /etc/scsimodes /dev/rsd0a
<lots of info on the disk>
```

The HP I bought came labeled "HP 97544SU". What the 'U' means I don't know. However it is not in the disktab file that way. I added that 'U' to my new disktab entry and everything went well. You can probably use /usr/etc/disk to write a new label to the disk and make the name consistent. If you are the paranoid sort, write the information down somewhere and compare it to the disktab entry for this disk.

10. Run /usr/etc/builddisk manually

I believe that the BuildDisk app simply gathers some info on the disk then runs the script /usr/etc/builddisk. The builddisk script has the following arguments:

```
/usr/etc/builddisk -l "label" -t "disk type" -s "script" type-of-disk
```

For the HP with 2 partions you invoke it as

```
builddisk -l "HardDisk" -t "97544S-512-2p" -s "sd330" sd0
```

(in this case the quotes aren't really necessary, but they don't hurt) You can call the disk something other than the arbitrarily chosen name of HardDisk.

If you planned for a very slim 'a' partition, as I did, then you must make some changes to the /etc/rc.swap script in order for the cube to boot properly

---

The builddisk script uses the /usr/etc/disk utility to initialize the new disk and build file systems on it. It uses the disktab entry you've written to lay the partitions out. After that it copies the operating system and utilities to the new disk by following commands out of the /etc/

BLD.sd330 file. It takes at least an hour to build a 330M hard disk from an OD.

When builddisk is done it unmounts the new HD partitions. Since you must make some changes on the new HD, manually mount these partitions in this way:

```
# cd /
# mount /dev/sd0a /mnta
# mount /dev/sd0b /mntb
```

In the rest of this section I refer to files like /etc/fstab, /etc/rc.swap, etc. These references are all to files on the hard disk, unless otherwise noted. If you perform the changes below while the hard disk partitions are mounted on /mnta and /mntb, then you must remember that the fstab I am referring to is in fact /mnta/etc/fstab, and not /etc/fstab.

If you planned for a very slim 'a' partition, as I did, then you must make some changes to the /etc/rc.swap script in order for the cube to boot properly. If you raced ahead without reading this and your cube is now in an infinite loop of "Checking disks..." "Loading from disk..." this is probably your problem. The swapfile in /private/vm takes 16M (at least). Since this can grow larger, depending on what you are working on, I wanted to have this file in the 'b' partition. I did the following to create a swap file on the second partition.

```
# cd /mntb
# mkdir vm
# cd vm
# touch swapfile
# chmod 1600 swapfile
```

I then made some changes to rc.swap on the HD (/mnta/etc/rc.swap). These changes destroy the generality of the rc.swap script. They are meant only for this case of putting your swap file on a second partition. As my machine's name is "moose", I named the 'b' partition "/moose". The complete modified rc.swap script is given in Appendix B of this article.

Since you now have two partitions, you must modify fstab and swaptab so that both partitions can be automatically mounted at boot time. Here is my new fstab:

```
#
# NOTE: This file is never consulted if NetInfo is running. It is only
# used during bootstrap.
#
/dev/sd0a /             4.3 rw,noquota,noauto 0 1
/dev/sd0b /moose        4.3 rw,noquota,noauto 0 2
```

And here is the new swaptab:

```
#       /etc/swaptab
#
/moose/vm/swapfile      lowat=20971520  # 20 Meg low water mark
```

The comment in the top of fstab is a good clue as to where it is easy to run into problems. If you are using NetInfo, then it is not enough to simply change fstab - you must also change the NetInfo data base. Since you are still running on the OD, you must shut the cube down and reboot single user to affect these changes. I suggest copying and editing all the files listed in Appendix A (and others you may want) while running on the OD with the HD partitions mounted in /mnta and /mntb. For instance,

```
# (cd /home; tar cpf - erickson) | (cd /mntb; tar xpf -)
```

copied my home directory to the new partition without changing any access times or file permissions. Once you've copied and edited all the system files (but definitely at least fstab, swaptab, and rc.swap) on the new HD partition, you are ready to boot single user. First you must unmount the HD partitions:

```
# cd /
# umount /mnta
# umount /mntb
```

If you get a "Device busy" error on umounting, make sure you don't have any shells (or other apps) open which are "in" either the /mnta or /mntb directories or their subdirectories.

Now logout and power down the machine. When you power up again, let the machine run until it is past the checking sytems state, then hit the right command and tilde (~) keys at the same time. This will put you in the NMI monitor. To boot single user from the new HD, type

```
NMI> bsd -s
```

When the machine gives you a "# " prompt, you are running a Bourne shell in one window as root. Use the niload utility to load the various ascii system files that have changed into the NetInfo database:

```
# niload -v -d fstab . < /etc/fstab
# niload -v -d hosts . < /etc/hosts
# niload -v -d passwd . < /etc/passwd
# niload -v -d group . < /etc/group
< and whatever else you may want >
```

For more info see the niload(1)/nidump(1) manual pages. When you are done fixing the NetInfo database, halt the machine and reboot multiuser:

```
# sync; halt
NMI> bsd
```

If everything has gone as planned you should now be running off the HD while swapping on the second 'b' partition. Note that the /tmp and /usr/tmp directories will

also be on the second partition.

A. Files which may have been changed on the OD

Some of the files I had to restore from the OD are listed below. Simply copy them (use tar to preserve ownership/permissions) to the HD.

```
<home directories>
/LocalApps
/LocalLibrary
/etc
          remote
          gettytab
          disktab
          rc
          rc.local
          uucp
          hostconfig
          hosts
          group
          passwd
          sendmail/aliases
          sendmail/sendmail.cf
/usr/lib
          uucp
          crontab
/usr/spool
          mail
          uucp
          uucppublic
```

B. Modified /etc/rc.swap script

```
#
# This script mounts and initializes swap space on a second partition.
#
# This is a modified version of the standard NeXT /etc/rc.swap script.
#
NEWSWAPPART=/moose
NEWSWAPDIR=$NEWSWAPPART/vm
NEWSWAPFILE=$NEWSWAPDIR/swapfile


# Find out which device to check.
SWAPDEV=`grep $NEWSWAPPART /etc/fstab | awk '{print $1}' \
        | sed -e 's/\/dev\///'`


# Check for the swapdisk
if [ -z "$SWAPDEV" ]
then
        exit 0
fi


useswap=0
if [ $1x = autobootx ]
then
        /usr/etc/fsck -p /dev/r${SWAPDEV} && useswap=1 >/dev/
console 2>&1
else
        useswap=1
fi


#
if [ $useswap -eq 1 ]
then
        # Mount the swapdisk
       /usr/etc/mount $NEWSWAPPART           >/dev/console 2>&1


        # Set up swapping on it
        if [ ! -d $NEWSWAPDIR ]
        then
              (echo "Creating vm directory on swapdisk")     >/
dev/console
                mkdir $NEWSWAPDIR
                chmod 755 $NEWSWAPDIR
        fi
        if [ ! -f $NEWSWAPFILE ]
        then
              (echo "Creating swapfile on swapdisk")         >/
dev/console
                touch $NEWSWAPFILE
                chmod 1600 $NEWSWAPFILE
        fi
       /usr/etc/mach_swapon -v -o prefer,lowat=16777216 $NEWS-
WAPFILE   >/dev/console 2>&1


        # Make sure there is a tmp directory
        if [ ! -d $NEWSWAPPART/tmp ]
        then
              (echo "Creating tmp directory on swapdisk")    >/
dev/console
                mkdir $NEWSWAPPART/tmp
                chmod 1777 $NEWSWAPPART/tmp
        fi


        # Point /private/tmp to it
        if [ ! -h /private/tmp ]
        then
                (echo "Linking /private/tmp to $NEWSWAPPART/
tmp")>/dev/console
                rm -rf /private/tmp
```

```
                (cd /private; ln -s $NEWSWAPPART/tmp)
        fi
else
        # Make sure /private/tmp is a directory
        if [ ! -d /private/tmp ]
        then
                (echo "Creating /private/tmp directory")>/dev/
console
                rm -f /private/tmp
                mkdir /private/tmp
                chmod 1777 /private/tmp
        fi
fi
```

MULTIMEDIA PROJECT

ANNOUNCEMENT

forwarded to us by Conrad Geiger,

SE Seattle NeXT

You may want to tell your User Group about this :

LIBERAL ARTS MULTIMEDIA EXPLORATION PROJECT

Mills College, Reed College, and the Advanced Technology Planning Group at the University of California at Berkeley are pleased to sponsor the Liberal Arts Multimedia Exploration Project. The target audience for this program is liberal arts faculty in the fine arts and the humanities with access to NeXT workstations who have an interest in instructional computing.

Participants will be provided with an opportunity to explore project ideas pertinent to their disciplines with other faculty, to gain hands-on experience with a powerful multimedia authoring environment on the NeXT workstation, and to reflect upon the value of this experience in their approach to teaching and student learning. The expected outcome for each participant is the development of one or more modules for use in some part of a course rather than a large scale effort requiring total course revision and formal classroom testing and evaluation.

The project provides a one week summer workshop experience with other faculty, followed by a one year period of electronic dialog, sharing ideas, techniques, and examples. In return, some commitment on the part of each participant's college or university is expected, particularly in providing travel funds and in providing access to NeXT computing resources at each participant's home institution similar to those used in the summer workshop.

The project seeks first-time faculty-developers as well as faculty with prior experience in developing computer-based materials.

The project seeks first-time faculty-developers as well as faculty with prior experience in developing computer-based materials The project will familiarize participants with multimedia design issues and tools, primarily through the use of a new NeXT multimedia authoring environment called MediaStation. MediaStation provides tools for creating and searching through large multimedia archives without having to master traditional programming skills. It provides tools for creating digital image libraries, digital audio libraries, and text libraries that can then be browsed or structured into multi-media learning experiences. For those interested in the field of electronic music, a variety of other tools centered on the NeXT DSP and MIDI interfaces will be provided. Project activities will consist of:

•a 5 day faculty workshop from June 17-23, 1990

•a year of development at each participant's home institution

•electronic dialogues and updates throughout the year among participants

•a 2 day conference in June,'91 to compare experiences & share insights

•dissemination of examples and a collection of essays on experiences

•Faculty members from schools interested in supporting this project will use the June 1990 summer workshop as a starting point to:

•discuss teaching and learning issues with other faculty members

•become familiar with the NeXT workstation environment

•become familiar with authoring tools of MediaStation

•become familiar with image scanning and sound digitizing

•begin dialogues with other participants that can be continued throughout the year

After a year of developing one or more educational modules using MediaStation or electronic music author-

ing tools on the NeXT at each participant's home institution, the June 1991 meeting will provide an opportunity to display the materials that each participant has created and to discuss student reaction to their materials.

More importantly, the June 1991 session provi be held in the Olin Library in the Scholar's Multimedia Workroom, using 8 NeXT workstations in a collaborative environment, on the Mills College Campus in Oakland, California. Rooms on the Mills campus, all meals and refreshments, the workshop itself, and materials for the workshop will be provided through funding from the Carnegie Corporation. Additional support for the project will be provided by NeXT, Inc. in the form of workshop speakers, publication of descriptions of participant projects in the planned NeXT Higher Education Newsletter, and the dissemination and publication of project results and insights. Participating schools are expected to contribute to the project by

•providing access to a NeXT machine on their campus for participants

•providing transportation to both workshops for the faculty developer

•helping the faculty-developer in other ways such as granting release time or providing TA assistance for materials gathering and copy editing.

•The workshop will be limited to 14 participants. Participants will receive a pre-conference packet consisting of print materials and possibly a video on previous faculty development projects with workstation technology.

•Applicants should submit a one or two page description of project ideas. The project description should include information on how the proposed MediaStation or electronic music authoring environments might be used to develop multimedia modules that could be incorporated into a course.

Carol Lennox of Mills College, Gary Schlickeiser of Reed College, and Barbara Morgan of the University of California at Berkeley will be coordinating the project and its selection process. Proposals should be sent to Carol Lennox at the addresses below by Friday, May 4, 1990. Candidates will be informed of the decision by Friday, May 11, 1990. Questions and comments can be addressed to :

Carol Lennox, Mills College, 5000 MacArthur Blvd, Oakland, CA 94613

E-mail addresses:

Internet E-mail address: lcl@mills.berkeley.edu

Bitnet E-mail address: lcl@mills.ucbjade.bitnet

AppleLink E-mail address: U0101

Phone: 415-430-2167

FAX: 415-430-3314

Programming in Franz Allegro Common Lisp on the NeXT Computer

John S. Karabaic, Systems Engineer

NeXT, Inc.

4434 Carver Woods Drive

Cincinnati, OH 45242-5455

513 792 5904

John_Karabaic@NeXT.com

In this paper, I will attempt to describe how to install Lisp, the programming tools available, how to use the on-line documentation, and how to access NeXT bundled software (e.g. the Application Kit) and hardware (e.g. the DSP).

March 31, 1990

1.0 Introduction

This is a paper intended for the somewhat experienced NeXT programmer who wants to know how to "do Lisp" on this system. I will assume some knowledge of Objective C, Lisp, Unix and the NeXT programming environment. I will not teach the Lisp language; there are many good books out there which do that much better than I would be able to in this space. It would be helpful if you've either been to Developer's Camp or have read the System Documentation on Objective C and Interface Builder thoroughly *while coding*. If you don't fit this profile, then get some experience in these areas before going on (unless you are particularly brave).

Programming in Lisp on the NeXT can be approached in a number of ways.

1. Self-contained, command-line-oriented Lisp programs can be written and used with great success. Large Lisp programs, such as Soar, have been successfully loaded and run in this way. We will not discuss this type of programming here, since it is covered sufficiently by Lisp texts available in many bookstores.
2. Lisp programs can be given a NextStep interface. This is very analogous to the way existing C code can be ported to the computer: Wrapping a user interface

around your existing code. Lisp code that has been given a NextStep interface still has to be started off from the command line, however. A typical session would begin by loading Lisp from the command line, loading the Lisp code which defines your application, and then running it from the Lisp interpreter prompt, causing the NextStep interface to appear. In Release 1.0, you cannot make a stand-alone Lisp-coded app that is startable from the Workspace using this technique.

3. Stand-alone applications with the core code written in Objective C which communicate to a Lisp image via a sockets (or Mach port) interface, much the same way Emacs talks to Lisp. We will not cover this here (since I haven't tried it yet), but it may be the topic of a future white paper. This opens up the possibility of an Objective C "Lisp Object", which would manage the communication between the Objective C and Lisp portions of an application.

We will concentrate on the second case.

2.0 Building a Lisp System

Most everything associated with Lisp on the NeXT machine is located in /usr/cl. To quickly build a Lisp system, change directories to /usr/cl/build and run, as root, the config script you will find there by typing ./config. Type a control-D at the first question, and yes at the next question. This will build a Lisp image in /usr/bin/cl in a few minutes.

You will find all of the detailed documentation you need to build Lisp located in /NextLibrary/Documentation/RelNotes/LispNotes.wn. This document gives you the lowdown on how to install Lisp and setup the Lisp/Emacs interface. This documentation will be called the *Release Notes* from now on.

Details on the Lisp/Emacs interface are found in /usr/lib/emacs/Lisp/fi in a file called spec.out. This gives you command summaries and other goodies. This documentation will be called the *Emacs Spec* from now on.

There are several fixes to the Lisp/Emacs interface, which are included in the same directory with this document. If you don't have them with this document, they can be found on any of the NeXT archive servers or on the Lighthouse Designs First Compilation Optical Disk, hereafter referred to as the *Lighthouse First Compilation*.

Franz Inc.'s Common Lisp Documentation, which consists of manual pages on their extensions to the language, is located in /NextLibrary/Documentation/CommonLisp. Inside that directory is a directory called CLUserMan, which contains various subdirectories, like *abouts* and *foreign*. These contain manual sections on various topics. This is not the complete *Common Lisp: The Manual*, which can be purchased at most bookstores. This is the same documentation which emacs uses for its Lisp manual commands. You can also just drag this directory into Digital Librarian for use there; it's already been indexed.[1] You may want to do that right now, so you can use the Request menu in FrameMaker to message Digital Librarian about topics in this document which interest you. This documentation will be called the *Manual* from now on.

3.0 Programming Tools

3.1 Editors

You have your choice of editors when working with Lisp on the NeXT machine.

Edit is the NextStep text editor which is fully documented in Chapter 11 of the System Documentation. Utilities like .userdict and .editdict files make Edit a somewhat extensible and customizable editor. Edit is located in /NextApps and can be launched from the dock or the Browser. See Chapter 11 of the System Documentation for details.

Emacs is a text editor from the Free Software Foundation which comes bundled with the NeXT Computer. It is a customizable, extensible editor which can also serve as your Lisp programming environment. You can specify that every Terminal you start be an Emacs session by typing

```
dwrite Terminal Shell emacs
```

at the command line. See Chapter 11 of the System Documentation for details on this topic. See the on-line emacs help for details on emacs functionality. The Emacs tutorial is not bundled with the NeXT System Software because of space limitations; it is available at many internet archive sites, on the Lighthouse First

1.To give the document a special icon in its well in Digital Librarian, you should either create a TIFF 48x48 pixel icon using Icon or some such or use the one bundled with this document in the file icon.tiff. You can copy it into the /NextLibrary/Documentation/CommonLisp/.index directory as icon.tiff. It will then appear in the icon well next time you launch Librarian.

Compilation, or is bundled with this paper.

A NextStep front-end to Emacs has been written by John G. Meyers and is available on the archive servers or is bundled with this paper.

The Emacs Spec shows how to hook various emacs commands to the on-line manual. Fixes from Franz, Inc., to make these functions work are either bundled with this paper or are available as described in Section 1. Emacs may give you many advantages when used as the Lisp editor; the Emacs Spec is a good starting point to learn about the possibilities of this tool.

3.2 Debugging

Debugging Lisp on the NeXT has to be divided into two domains: debugging your Lisp code, and debugging foreign code (*eg*, Application Kit code and Objective C objects) that you have defined in Lisp.

3.2.1 Debugging Lisp Code

The debugger included with Franz Allegro Common Lisp consist of a tracer, stepper, and inspector, a set of top-level[2] commands that allow dynamic examination and manipulation of Lisp data and the run-time evaluation stack, and mechanisms to single-step through expressions or function calls. The debugger is not much help when trying to debug foreign code. which includes the Application Kit. Some of the other facilities are helpful there.

The Franz debugger is completely documented in the Manual.

3.2.2 Debugging Objective C Objects in Lisp

When you define Objective C Objects in Lisp, as you will do any time you wish to use Interface Builder or the Application Kit, there are some bundled functions which allow you to inspect Objective C objects with great detail. These include the Lisp functions `print-class`, `print-class-methods`, `print-class-ivars`, `simple-print-class`, `get-entry-points`, `find-class`, `method-number-to-name`, and `print-instance`. The global variable `*all-classes*` keeps track of all the classes Objective C knows about.

2.Top-level commands are special commands in Allegro CL. See the “about top level” section in the Manual.

4.0 Programming on the NeXT

4.1 Using the AppKit with Interface Builder

Code examples come bundled in /usr/cl/examples/objc. These are a good introduction to programming with Interface Builder and Lisp. I am going to assume experience in Interface Builder, the AppKit and Objective C in this section; I strongly urge you to read the System Documentation on these subjects.

The first thing you would probably do is design your application’s interface with Interface Builder. You would define most of your new classes there, parse out the Objective C code that IB creates and save the project and nib files.

When you parse out code for your class specifications, Interface Builder will generate Objective C code. You must translate this code into Lisp manually, since Interface Builder does not generate Lisp code. The file that Interface Builder generates to store the interface you designed (the `.nib` file) is loadable into Lisp, however.

Let’s take a look at the example Calculator created by Charles Cox of Franz, Inc. This file is in /usr/cl/examples/objc/Calculator.cl. It is a direct translation into Lisp of the application in /NextDeveloper/Examples/Calculator--it even uses the same Calculator.nib file!

The first thing to be noticed is the packages we use:

```
(require :objc)

(require :foreign)

(use-package :excl)
(use-package :objc)
(use-package :appkit)
(use-package :foreign-functions)
```

Each of these packages is described in the Manual; check the “about packages” and “about objective c” sections.

Let’s contrast the Calculator object’s definition in Objective C and Lisp. In Objective C, you need separate header and module (.h and .m) files for the object. Lisp has no such requirement, so while the Calculator.h file looks like this:

```
@interface Calculator:Object
{
  id aWindow;
  id viewer;
  char viewerStr[24];
```

```
  doublestack[STACKSIZE];
  int  topOfStack;
  BOOL frozen;
}
```

the corresponding code in Calculator.cl looks like this:

```
(def-objc-class Calculator (Object)
  ((aWindow :type :id)
    (viewer :type :id)
    (viewerStr :type :int)
    (stack :type :int)
    (frozen :type :int)))
```

Note that keywords are used to make explicit type definitions of the instance variables. The Manual section "about C types" contains valuable information on the foreign function interface to C and Objective C. Correct type definitions are vital to making your Lisp code work correctly with the Application Kit. In this case, we are going to store the stack in a Lisp list, which means we eliminate one instance variable (`topOfStack`) and change the type of another to accomodate an index to a registration table where a Lisp-value is stored. We also change the `frozen` variable to an `int`, because we use two integer constants to store the state of the Calculator. We do this to keep things simple because the foreign-functions interface only knows about a few vanilla C types. We could define a C type (see `def-c-type` and "about C types" in the Manual) for BOOL, but that's best left as an exercise for the student. We'll discuss indexes a little later when we talk about the functions `lisp-value` and `register-value`.

We don't need the rest of the Calculator.h file, since it just contains information on the Calculator object's interface to the outside world which Lisp doesn't need. This means that in Lisp you can't create "private" instance methods, as you can do in Objective C.

Let's compare an instance method definition in Objective C

```
-displayValue:(double)aNum
{
  if (frozen)
    return self;
  [viewer setFloatValue:aNum];
  strcpy(viewerStr,"");
  return self;
}
```

and in Lisp

```
(def-objc-method
  (displayValue= Calculator :id)
    ((aNum :double))
    (if* (= (iv "frozen") YES)
      then (return self))
    (send (iv "viewer")
      "setFloatValue=" aNum)
      (setf (lisp-value
      (iv "viewerStr")) "")
  self)
```

Note that we specify the returned type of the instance method in the first argument to `def-objc-method`, which is a list specifying the instance method name, the Objective C class to which it belongs and the type returned by the method. Since `:` is a special character in Lisp, we specify that method takes an argument with an = sign. We could also have written that list as

```
("displayValue:" Calculator :id)
```

We follow this list with a list of lists specifying arguments to the instance method. In this case, there's only one argument, a `:double` called `aNum`.

To access the instance variables of our object, we call the function[3] `iv`, which can only be used from within a method definition, since it references the variable `self`. `Self` is a global variable in the `objc` package bound to the current object (or receiver) during the execution of a method.

We send messages to Objective C objects using the `send` function

```
(send receiver
  method-name-or-number
  arg1 arg2 arg3…)
```

So you can see that we are doing the equivalent functions in the Lisp code that would be done in the Objective C code.

The one function I did not touch upon is `lisp-value`. This is a lisp function which allows your code to access a value stored with `register-value`, which is a way for Lisp to keep track of values that C (or other languages) care about. So, for example, while we used vanilla Lisp lists to store the stack in, we declared it to be an `:int` so C could keep track of the value in a registration table where the list was stored. We then used the

3.From the Manual: "Several 'functions' in the Objective C interface, such as *iv* [and *send*--JSK], are actually both functions and macros. They act like functions when interpreted and may be used with *funcall* or *apply*. But when compiled they may expand in line like a macro. This is done to extract maximum efficiency while still enabling the programmer to use *funcall* or *apply*."

lisp-value function to simulate accessing a registered value. If we wanted to access a Lisp list from C, we would include the file /usr/cl/lib/misc/lisp.h and use the macros in there to play with registered values, once we had loaded the C executable into Lisp. register-value makes sure that the value is never garbage collected. Allegro CL includes a similar function called register-function, also.

See the Manual sections on lisp-value and register-value and the Calculator.cl file for further details.[4] See the Manual section "about foreign functions" for even more details.

Once we have translated the Objective C code in our header and module files, we need to have some way of loading our nib file and running the application. In Calculator. cl, this code is at the tail of the file, and it's a pretty faithful translation of what's in Calc_main.m.

```
main(argc, argv)
     int argc;
     char *argv[];
{
  NXApp = [Application new];
  [NXApp setAppName:"Calc"];
  [NXApp loadNibSection:"Calc.nib"
  owner:NXApp];
  [NXApp run];
   [NXApp free];
   exit(0);
}
```

This code is pretty straightforward:

```
(make-application "Calc")

(send NXApp "loadNibFile=owner="
   "/NextDeveloper/Examples/Calcu-
lator/Calc.nib"
     NXApp)

(send NXApp "run")
```

The make-application function is a special Lisp function that sets a global variable, NXApp, equal to the one and only one Application object that should be present. We then send two messages to the NXApp object, telling it to load the nib file for this application (the same one as for the Objective C version) and then run.

Presently Lisp will not return after telling NXApp to run. If you quit the application via its Quit menu selection, you will exit Lisp.

That is a brief overview of using the Application Kit with Interface Builder. Next, we'll tackle using the Application Kit without Interface Builder.

4.2 Using the AppKit without Interface Builder

Interface Builder is a great way to use the Application Kit without worrying about details too much. You can change your interface and not change your code much, if at all.

If that doesn't appeal to you, you can access the Objective C Classes and code that makes up the Application Kit directly from Lisp via the foreign function interface. In this section, we will recreate, in Lisp, one of the most popular labs from NeXT Developer's Camp: ConverterFromScratch.

That lab is the Celsius-to-Fahrenheit converter in Chapter 8 of the System Reference Manual ("Interface Builder"; found in /NextLibrary/Documentation/NeXT/SysRefMan/08_IntfBuilder.wn) redone without the benefit of Interface Builder. It's still a surprisingly small amount of code, but easily an order of magnitude greater than the Interface Builder version.

The first thing you need to do is grab some #defines out of header files in /usr/include and convert them to defconstants. I have done most if this in the file ConverterFromScratch.cl that is in Appendix A, but here's a sample converted from /usr/include/dpsclient/dpsNeXT.h:

```
(defconstant NX_CLEAR 0)
(defconstant NX_COPY  1)
(defconstant NX_SOVER 2)
(defconstant NX_SIN   3)
(defconstant NX_SOUT  4)
(defconstant NX_SATOP 5)
(defconstant NX_DOVER 6)
(defconstant NX_DIN   7)
(defconstant NX_DOUT  8)
(defconstant NX_DATOP 9)
(defconstant NX_XOR   10)
(defconstant NX_PLUSD 11)
```

4.From the Manual: "The issue here is as follows: C code does not expect that values will be moved. Therefore C code expects pointers, once established, to be good for ever (or at least until the C code itself changes things). The garbage collector in Lisp, however, is constantly moving things about. The garbage collector will update all relevant pointers in Lisp, but it knows nothing of C code. The function register-value allows you to explicitly direct Lisp to keep track of these pointers."

```
(defconstant NX_HIGHLIGHT 12)
```

Some C macros will have to be replaced with Lisp macros or functions. For example, from View.h:

```
#define NX_XSIZING(mask) (mask) & 7
```

This will probably have to be redefined with a function that maybe calls some C code or does bit-twiddling on a C type.

Review of Who's Calling?

Terrence J. Talbot

Capsule Review:

Who's Calling? is, according to the brochure, *a fully automated client data management system*. For the most part, this is an accurate statement. Who's Calling?, as its name implies, works exceptionally well for managing incoming and outgoing phonecalls, including client lists, as well as managing in-house notes and reminders. Who's Calling? does, however, lack some natural extensions to working with client data that would truly automate routine office tasks. Depending on the scope of your needs and the nature of your business, Who's Calling? may be just what you've been looking for.

Pros: Provides a *tickle mechanism* to notify you of upcoming events, including optional voice messages. Uses SQL technology to manage a rich, information intensive environment. Fosters the creation of multiple ™resources for comprehensive event management and agenda creation. Easily handles creation and distribution of *While You Were Out* messages. Tracks phonecalls, both incoming and outgoing, and assists in placing outgoing phonecalls. Creates and prints mailing labels from the Client Database.

Cons: No way to easily set up recurring appointments. No way to overlay reports from several *resource*s. No way to easily associate clients with multiple categories and other clients. Does not have a letter writer for external correspondence. Lacks a variety of printed report formats needed for keeping agendas while away from the NeXT machine. Arbitrarily sized fields that do not accept returns or tabs to allow for structuring of event descriptions and general comments.

Price: $495 (Adamation tells me that part of this cost comes from the licensing fee they must pay to Ingres for a copy of their database server. Adamation has informally told me that they are working to incorporate other SQL servers, such as Sybase or Informix products, should you use them to handle SQL transactions. Whether this will lower the per user cost of the product is doubtful, but would depend on individual situations.)

System Requirements:

For single machine use, Adamation recommends a NeXT cube with at least a 330 Mb hard drive. This is, I assume, because the Ingres database that Who's Calling? utilizes requires at least 24Mb just to install. Beyond that, the application itself is over 3 Mb including sound files. (Let me state, however, that with a little creative installation I was able to install the Ingres server on a 40Mb swapdisk and the application in /LocalApps and use it that way on an OD only machine. I doubt that this would stand up however once the size of the database files became significant and I don't think such a procedure is recommended by either Adamation or NeXT.)

For Further Information Contact:

Adamation

1435 Center Street

Oakland, California 94607

(415) 452-5252

Preface

Let me begin this review by thanking Adamation and their president, Stephen Adams, for sending me a demonstration copy of their software to examine. I originally contacted Adamation by telephone requesting that they send me information about Who's Calling?. They sent me a small but attractively designed brochure. Unfortunately, the brochure, while artistically appealing, was not particularly informative. I mentioned this fact almost offhandedly and somewhat harshly on CompuServe and Young-Kyu Yoo at Adamation, who participates in our UNIXFORUM NeXT group on CI$, was quite willing to listen to my criticisms and, indirectly, put me in touch with the president of Adamation, who was very kind to send me a demonstration copy of Who's Calling? for review. Adamation was very helpful which I think speaks well of their company. I should also mention that the demonstration copy I reviewed was stated to be fully functional except for the fact that the database will only handle 25 contacts. I think this is a great way to produce a demo, rather than the often seen *feature-disabled*, a. k. a. *brain-damaged*, variety, and is a further praise for Adamation in general.

Let me further preface this review by stating that I, personally, have a way of focusing in on details that I es-

pecially enjoyed or that I think need improvement when I review a hardware or software product. Who's Calling? is a fine software product that can become even better. As with many software packages, Who's Calling? somewhat forces you to adapt to it, rather than the other way around. I would prefer software products that adapt to me. Therefore, when I review a product, I occasionally suggest improvements that would make the software work as I would expect it to work and need it to work. Let me say, however, that Adamation has been very open to criticism and suggestions for improvement. In a telephone conversation with Stephen Adams last week, he detailed improvements and additions to Who's Calling? that answered nearly all my concerns. The improved 2.0 product is under construction now and should be ready for a ™late fall release. This review deals primarily with the 1.0 release which is shipping now.

Main Text of Review

The Who's Calling? system centers around five *modules* that handle information relating to your clients, or contacts, and your daily schedule. Of these, three are key to working with the system. These are the Calendar module, the Client Database, and the Conversation Manager. Beyond these are the Message Board, the Telephone module, and the Tickler module.

The learning curve is initially fairly steep as the mind tries to grasp the intent of the several components of the system. This is the one place where I think the Who's Calling? system conceptually gets in the way of itself

I was, at first, dismayed by the scope of the application. The learning curve is initially fairly steep as the mind tries to grasp the intent of the several components of the system. This is the one place where I think the Who's Calling? system conceptually gets in the way of itself. Rather than simply say there are several windows that you will want to know about immediately and use all the time, the documentation presents the system as a number of different modules, each of which is separate and must be learned in turn. As such, I initially found the software somewhat forboding, rather than feature-rich. Luckily, I persisted and ultimately found the software quite easy to use. Do not be daunted by Who's Calling?'s complexity: like all good NeXT applications, the interface works hard to simplify things.

The Calendar begins by presenting you with two windows: a monthly calendar and a daily agenda. The monthly calendar is straightforward and moves you easily from day to day, month to month, and year to year. The daily agenda window presents you with a list of the selected day's events and allows you to manipulate events in several ways. An event can be considered anything that you need to tend to at a certain time, such as placing a phonecall, keeping an appointment, or simply a general reminder to do something. You may obtain just a short list of the day's events, or you can select an ™extended view which places events in the context of one hour blocks on the page.

New events are created in a separate window which then feeds information into the Daily Agenda list. When creating or updating an event there are a number of adjustable options. Chief among these are the ability to assign events to particular *resources*. A resource is essentially an alphanumeric code, eg. *room* or *Janine*, that specifies a person or place to associate an event with. This allows one to very easily set up schedules for a number of different places and entities. For example, through careful use of the resource field, one can set up a schedule for a particular conference room or a workgroup.

Other fields allow for the setting up of the date and time to notify you of the event, a field to set up the duration of the event, eg. a two-hour meeting, and a field to set up a *snooze* interval for the event. A snooze interval is simply a period of time set up to reactivate the event if you press the ignore button after the first notification of the event is provided: this works much like the snooze button on a bedside alarm clock.

Once events have been created and placed in the daily agenda, printed reports can be generated, organized by either date or resource. Additionally, Who's Calling? can output your calendar in the form of a WriteNow data document for later merging with whatever WriteNow templates you devise.

The most glaring omission from the Calendar is the lack of ability to set up recurring appointments. These are useful for setting up reminders about events that happen periodically, eg. quarterly taxes, staff meetings, etc.

Additionally, a potentially important feature that is not available in the current release is the ability to overlay agendas from several different resources onto the same schedule. This would make it easy to notice conflicts between resources and arrange schedules for multiple resources, as well as provide a method for filtering out various resources from the master schedule. For the cur-

rent release, I found it was generally better to keep all events under one resource so that I could view them on the same page.

There is a way to combine events, identified by resource, on a printed report. However there is no easy way to select particular and specific resources or combine them on the report in a free form manner. Rather, Who's Calling? forces you to specify a *from* and *to* resource list which becomes difficult to work with after three or four resources are added. Schedules with multiple resources are an excellent innovation for analyzing and tracking events. Hopefully a future update to the software will bring the power of this scheme to its full resolution.

One fascinating feature of the application, although I'm sure with the cube's extraordinary capabilities it's not difficult to add, is Who's Calling?'s voice prompts. Rather than, or in addition to, standard informational panels that inform you of events and application activities, Who's Calling? notifies you by speaking to you. A female voice, similar to the one that notifies you if the printer is in trouble, announces standard panel messages such as *Is the information correct*? and *Record added*.

Additionally, Who's Calling? will vocally notify you of upcoming appointments, phone calls, etc., by saying You have an appointment. and then bringing the window with the appropriate information to the front. I find this much more preferable than having a modal panel appear on the screen freezing you out of further work until you click the *ok* button. Some may find the voice a nag, but I thought it was an elegant, nonintrusive solution.

Additionally, Who's Calling? will vocally notify you of upcoming appointments, phone calls, etc., by saying You have an appointment. and then bringing the window with the appropriate information to the front

Who's Calling also features a *Talking Clock* that tells you the time. The time is spoken somewhat robotically since the digits are pieced together but it is quite acceptable. A powerful feature of the talking clock is its ability to set up two types of alarms. One alarm merely notifies you at a certain time of day, say at 6 AM and 6 PM. The other alarm allows you to set up intervals, say every fifteen minutes, when the clock will announce the time aloud. This presents you with a very flexible and non-intrusive alarm clock for reminders while working in other applications. For example, say I have an appointment at 2:30 PM but I am working feverishly to finish a paper in WriteNow or FrameMaker. The answer is simple: pop up Who's Calling?; set up an alarm for 2:30; pop back to the word processor and keep typing until the computer tells me it's time to go. Very nice.

Similarly, it is quite easy to set up *reminders* of things that need to be taken care of at certain times during the day, today, tomorrow, next week, or next year. The tickler then monitors when events are upcoming and notifies you by announcing the fact, e. g. the voice says *This is a reminder.*, and then a panel pops up showing you all the information concerning the event. You can then choose to ignore the reminder which will places it into snooze or you can accept it and no further notification of that event will result.

The Client Database stores and manages information relating to the clients or contacts that you want to keep track of. The variety of fields provided for entering information about your clients is quite rich. For example, a field is provided for a person's birthday. Storing this information, you could then have Who's Calling? reming you to send a gift at the appropriate time. You can index your clients not only by name and address but also through the category field. This allows you to associate many different clients into one overarching group for doing such things as printing out groups of mailing labels (both 2-up and 3-up labels are provided for) and the like. In addition to the standard fields, you can define and label up to five additional fields for search purposes. Unfortunately, like the event description box, the comment box for any additional information about a client does not accept returns for formatting several pieces of data within one field. Moreover, many of the fields are unfairly limited in the numbers of characters which you may input. For example, the category field is limited to 10 characters forcing you to often devise abbreviations and codes for your categories. Still, quite a lot of information about any particular client can be recorded and the options for printed reports are quite varied. Clients may be listed by any named criteria and may be printed according to company name or contact name. An additional feature is the ability to record voice comments to store with the client info should you have a microphone attached to your workstation.

In my opinion, a significant productivity enhancement potentially available with personal computers in an office place is to automate all but the composition of the body of a piece of correspondence you need to send out. Who's Calling? falls somewhat short of my ideal system. There is no automatic link between the Client Database and the NeXT Mail application. To be fair, the Message

Board does allow you to send notes to other user's of your network's Who's Calling? system. However, there is no provision for sending regular electronic correspondence over whatever network(s) your cube has access to. More importantly, for at least this point in time, there is no (easy) way to automatically create a standard letter for U.S. Mail.

I was able to create letters to clients by creating a *letter* template in WriteNow and then using Who's Calling?'s mail merge report function to create a data document containing one entry. An example template document for use in WriteNow is given in the paragraph below. To use it, copy the text into a blank WriteNow document and save it under an appropriate name. Leaving the template open in WriteNow, flip back to Who's Calling? and open the Client Database window. Choose the *Reports...* function button and click on the *Mail Merge Data* button. Save the file, as named by Who's Calling?, to the same directory as your WriteNow template. Next flip back to WriteNow, making the template the key window. Next, click on the *Merge...* menu command and send the output to a new document. Compose your letter. As you can see, this is clunky, but it does work. It is certainly far better than nothing. I would like to be able to easily set up a letter from within the Client Database and to easily paste in boilerplate language, such as a signature block, where needed. Perhaps Adamation will add this feature in a future release.

Sample mail merge template to bridge Who's Calling? and WriteNow

---

´DATA Clients.datª

´mrmsª. ´first nameª ´middle nameª ´last nameª

´titleª, ´companyª

´address1ª

´address2ª

´cityª, ´stateª ´zip codeª ´zip code endingª

´countryª

Dear ´mrmsª. ´last nameª:

The other main part to Who's Calling? is the Conversation Manager and the related Telephone module and Message Board. The Conversation Manger allows you to place calls (should you have a modem attached to your cube) and to track incoming and outgoing calls. The Manger allows you to track the lengths of calls, to maintain notes about the conversation, to set up and track call back times and to record voice comments about the conversation. It seems quite fully featured (the name of the product is Who's Calling?, right?). It interfaces completely with the Client Database., provides report functions for creating printed phone logs and should prove invaluable to people who need to place and receive many calls throughout the day.

Similar to the Conversation Manager is the Telephone module. The Telephone module allows you to set up a *speed dialin*g list of your most frequently called numbers and also provides quick access to, and dialing of, emergency numbers. Additionally, the Telephone module presents you with an on-screen representation of a standard Touch-Tone phone pad for direct dialing of other numbers not entered in your database, foreign numbers, and for negotiating with unusual phone systems and long distance servers.

The Message Board provides a simple way of creating and sending *While You Were Out* messages to other people on your Who's Calling? system. Standard information is provided for, including a large number of check off buttons for conveying what action the caller wants you to take. It's best potential seems to be as a quick and simple e-mail system for receptionists and similar persons to notify people of their messages. I'm sure it would also prove valuable for people in an organization who need to dash off quick notes to each other, although, as mentioned, a direct link with NeXT Mail would be even better.

Installation is relatively uncomplicated but does require setting up the supplied Ingres database and creating accounts for the users of Who's Calling? as well as a system account for Ingres itself

In addition to the five modules, the clock, and the tickle window, there are three additional aides accessible from the Who's Calling? palette. These are the calculator, an area codes list, and a state abbreviations list. None of these three are terribly exciting but do help to make the software complete. The calculator is a standard six function desktop calculator (four operators, square root, and percent) with rudimentary memory features. Having becoming accustomed to more complex on-screen calculators with the Macintosh, I was looking for more than what Who's Calling? provides. At a minimum, I would

like to see a calculator with a paper tape readout (invaluable for double checking calculations), standard trigonometric functions, and exponential functions. Of course, the functionality of an on-screen calculator could be extended almost indefinitely, but Mathematica is already included, right? Still, a more fully featured calculator would be welcome. The Area Codes and States lists are just what you'd imagine them to be. How often they would prove useful depends on individual needs.

Additionally, it would be nice to have the Who's Calling? palette appear a little farther down the left side of the display so that longer menus, such as those used in WriteNow and Frame, do not cover up the palette.

An excellent feature of the NeXT's multitasking nature and large screen is the ability to click on buttons in other applications, have that application brought forward, and the button target execute its function. For example, it's quite easy to be working in WriteNow, hit the Calendar button to bring up the Daily Agenda window, check over your events, and flip back into WriteNow to keep working. To me, that is Who's Calling? (and the cube's) greatest selling point.

The documentation, at least in the version I received, is supplied as WriteNow files on the optical disk. This makes it easy to drag into the Digital Librarian, for quick, on-line reference. In my estimation, this is as it should be.

A word about installation – Installation is relatively uncomplicated but does require setting up the supplied Ingres database and creating accounts for the users of Who's Calling? as well as a system account for Ingres itself. Additionally, Who's Calling? requires the Ingres clock to be set up and synchronized with the system clock. I was not altogether successful in this. Although I thought I had followed instructions precisely, I could not synchronize the Who's Calling? clock with the clock displayed by the Preferences application. The problem can be straightened out though and perhaps Adamation will discover a way to automate that part of the installation process.

As mentioned in the preface, Adamation is hard at work on Who's Calling? 2.0. This new version should address all my criticisms as well as add some outstanding new features. Realizing that product specifications often change radically until officially released, the 2.0 product is said to do a much better job at handling multiple resources and multiple users on the same agenda page (what I like to think of as overlaying resources onto a master agenda). Additionally, the calendar and daily agenda views should be much improved and much more flexible with the addition of Day-At-A-Glance (and other) styled views, as well as Pert charts (wow!) and, perhaps, a Gantt chart interface. Recurring appointments will be included, fields will work more intuitively, key equivalents will be added, and, hopefully, multiple client categories will be introduced for more flexible client handling.

And finally, for those who wonder if anyone else has bought a cube, the William Morris Agency has apparently inked a deal with NeXT and Adamation to supply the company with 250 workstations, all running Who's Calling?. It should be a mighty impressive setup and generate exciting new ideas for client management and other productivity applications.

Overall Who's Calling? is a gigantic step toward automating and keeping track of the details of life that seem to take an endless amount of time. As the product is refined, its innovative features (such as resources) should develop into an extremely powerful environment. If you need to keep track of time, people, or phonecalls, Who's Calling? is certainly recommended. Finis.

The author can be reached electronically at 72307.1727@CompuServe.COM until such time as he gets his cube hooked in to Internet or a NeXT specific network somehow, someday. (Please, someone from NeXT tell me what the present situation is on this.)

Bonus Preview of Paper Sight

by Terrence J. Talbot

On May 11, I had the opportunity to attend a demonstration of an intoxicating new product called Paper Sight from Visual Understanding Systems, located in Pittsburgh. This application, billed as `personal microfiche and filing cabinet' creates and maintains files of scanned documents, manipulable with a number of exciting features.

Using a scanner, one can, from within the program, scan documents (correspondence, forms, journal articles, etc.) into the work area and store them in ™file cabinets for later retrieval. The application is unique in that it not only stores the image at whatever resolution you scanned it, eg. 300 dpi, but also creates a tiff image tailored to the resolution of the MegaPixel Display. This allows for amazingly quick retrieval and display of the document on the screen as well as surprisingly fast printing of the document at full resolution. (We're talking on the order of a magnitude greater than what I've seen the Macintosh capable of doing.)

If VisiCalc sold Apple IIs, 1-2-3 sold IBM PCs, PageMaker sold Macintoshes, and Mathematica should have sold everyone, products like Who's Calling? and Paper Sight could sell a billion NeXTs

Beyond the simple scanning and storing of documents, Paper Sight provides for several innovative technologies all available in one package. Because the image is stored separately, multiple users can overlay comments in the form of directly placeable text or graphics onto the page. Additional features allow you to notify others on your network that a particular document has been updated with comments. I assume this allows you to easily set up workgroups for editing or commenting about documents, no matter what their form. Paper Sight also provides for the attachment of notes to each page or document that can be indexed for quick retrieval of the document from its cabinet.

In addition to manually entering notes to describe or index the document, Paper Sight allows you to draw boxes around parts of the document that you want to bring in as full ascii text. Using OCR technology licensed from Kurtz-Weil, Paper Sight quickly, and as far as I could tell accurately, converts your selection to text and places it in a note for indexing purposes. Your selection can be as small as a word or as big as the whole page. OCR technology is available on the NeXT!

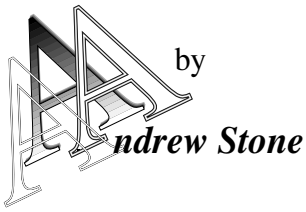
It was a short demonstration and I did not have the opportunity for any hands on experience, however, from what I saw, I think Paper Sight is a product with huge implications for dealing with the dreaded Information Age. Academics should find it quite a tool for archiving journal articles, research notes, and other pieces of paper that clutter their desks and filing cabinets as well as provide an attractive way to handle group editing of documents and layouts. Government agencies could use it to archive applications and official records and general businesses ought to be able to use it for any variety of those purposes. The main drawback, as it is when working with any scanned graphic image, is storage space. An uncompressed scan of a letter sized page at 300 dpi takes about 1 Mb of disk space. Paper Sight provides several compression options which, at best, reduce the size of each page to about 200k. Still, disk space will be eaten rapidly. (Time to get working on those 1 Gb optical disks!)

Paper Sight is reportedly shipping now but still undergoing some interface changes. (Apparently Steve didn't like the interface.) Unfortunately, I was unable to find out the price of the product or a phone number to call for more information. (It's now the weekend and I have to get this mailed off.) Visual Understanding Systems has some connection with Carnegie-Mellon University and I would recommend checking with the University if you are unable to locate a number for the manufacturer's Pittsburgh offices.

My excitement about the NeXT is now increasing exponentially. If VisiCalc sold Apple IIs, 1-2-3 sold IBM PCs, PageMaker sold Macintoshes, and Mathematica should have sold everyone, products like Who's Calling? and Paper Sight could sell a billion NeXTs.

Stone Age n

(1864) :the first known period of prehistoric human culture characterized by the use of stone tools.



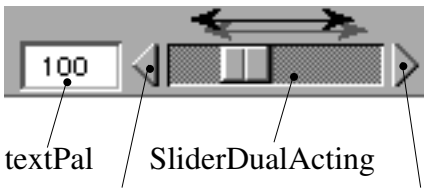
Welcome to the Stone Age! This column shall be dedicated to a recursive descent into appkit programming: stone tools aka Acme Gizmos, new objects, tricks, and tales from the dark night of the programmer's soul. In the spirit of FSF and my conviction that this really is a revolution, here's a tool from my chest. Like all good object libraries, this column should be written collectively. I invite all of you to "Step right up" in this space and show off your neat objects, even if unpolished, because its the concepts that count. Next month: Kris Jensen describes a new Gizmo.

Composite Objects: A Slider/TextField Hybrid

SliderDualActing and its cell class, SliderCellFine, combine a special ™dual-acting slider, a text field, and two arrow buttons into one composite object. This allows the

end-user multiple input techniques while abstracting the client or application user of the object away from the details of coordinating the various Appkit components. The special slider solves some of the problems of providing fine control over a large range of values in a small space. This project illustrates the power gained both by subclassing Appkit objects and combining objects.

SliderDualActing Components



Features

Component objects make programming easier:

The end-user gets multiple input methods: a slider, a textfield, and a pair of buttons which decrement or increment the slider, while the program only deals with one object. All of the validation and synchronization methods are in one place instead of distributed throughout source files. Programs often have lines like these in read: and other resetting methods:

```
[numberField setIntValue:number];
if (number>[numberSlider maxValue]&&okToExceedMaximum)
          [numberSlider  setMaxValue:number];
[numberSlider setIntValue:number];
```

All of the coordinating code can be replaced by the last line. The SliderDualActing handles the rest.

•We want to continually update the textfield as we drag the slider (and SliderDualActing handles this without any programmer intervention), but we also want to notify another target either continually or on mouseUp. The second case is more usual, as continual notification may take too much processing time. Therefore, our composite object has an UpTarget and an UpAction.

•Users should not be stopped from entering values higher than maxValue or lower than minValue, if it makes sense to the application. For example, in TextArt, the fontSizeSlider has a default value of 140 points. However, if users want a higher value, they simply type in a value into the textPal, and if the SliderDualActing BOOL instance variable allowHigher is YES, the slider resets its maximum value. A further enhancement to this class would be methods to write the user's maximum and minimum values to the NX_Defaults mechanism to restore on the next launch.

•Sometimes, the user justs wants to ™crawl along the slider's values. I have added a matrix of two buttons which increment or decrement the slider by altStep. They connect to the sliders' action method incrementDecrement:.

• Component objects can provide a convenient method for implementing Undo. The SliderDualActing knows its last value and can respond to an undo method. Alternatively, you can specify an undo target and a tag telling the target what needs to be undone. The default is for the slider to handle undo itself. One simple strategy for a single-level undo is to have a global undo object, which stores the id of the last control set. Undo from the menu tells this undo object to send the undo method to the control that last set it. My undo object is accessable via NXApp, and looks like this:

```
@interface Undo:Object
{
    BOOL  hasUndo;
    id  lastClass;
}
- setLastClass:anID;
- undo:sender;
@end
```

Special sliders provide several additional user interface features:

• Sliders with a large range can cause problems: If the slider isn't very long, the resolution (how much the value changes when the slider is dragged one pixel) can become unacceptably large. On the other hand, screen space is often in short supply, making it hard to use the long sliders needed to obtain a fine resolution. SliderDualActing and SliderCellFine provide methods to change the slider's value by specifiable small amounts by checking if the Alternate or other meta keys are down while the slider is being dragged. Currently, pressing the Alternate key causes the slider to change by the altStep instance variable and adding the Shift key halves this amount.

• Looking at float values that have insignificant digits is *noisy and very unSteveLik*e.. Sometimes, however, a user wants to specify a value to a high degree of accuracy. Our composite object provides a mechanism to dynamically change the TextField textPal's floating point format. Normally the user will not be bothered with the

floating point values, but dragging the slider while pressing the Alternate and Shift keys puts the SliderDualActing into decimal state (as well as changing the value by a prespecified small amount). This allows precision while preserving aesthetics.

• The user can reset the slider to its various defaults (maximum, minimum, number of decimal places displayed) while using the program by clicking the slider while pressing the Command key.

Using SliderDualActing, a step by step guide in Interface Builder:

The first time the object is used, perform steps 1

This is what a SliderDualActing looks like in Interface Builder

Object kept together in box

thru 13; thereafter only step 13 is necessary:

0] Assume that you have a class with an instance variable named **sliderDA**

1] Copy SliderDualActing.[hm] and SliderCellFine.[-hm] to your project directory.

2] Bring up the Class Window (Command-5) and the Class Attributes Inspector (Command-1)

3] Make a subclass of slider:

a] Traverse the Class hierarchy to Object->Responder->View->Control->Slider

b] Select SubClass from the Operations pull-down menu in the Class Window

c] Rename the subclass to SliderDualActing

d] Select Parse from the Operations pull-down menu and answer **OK** to the **Add**

SliderDualActing to Project dialogue box

4] Create a new custom view by dragging one from the Palette Window

5] Make it a SliderDualActing by bringing up Inspector Attributes Window (Command-1) and clicking on the SliderDualActing class name

6] Size the new slider: Bring up Inspector Size Window (Command-4) and type in a width of 16. if it's to be a vertical slider, or a height of 16 if it's a horizontal slider.

7] Drag a textField from the Palette Window

8] Drag a button from the Palette Window. Make a matrix of two buttons by dragging while pressing the Alt key. In the Inspector Attributes window, be sure to check **Cells Tag** = Position. Add arrow icons to these buttons.

9] Select the three objects and **box** them by typing Command-g. This allows you to copy and

Paste the control while retaining the connections among the three objects.

10] Connect the textPal outlet of sliderDualActing to the textfield.

11] Connect the action of the textfield to the SliderDualActing method: **sendTextAction:**

12] Connect the matrix' target to the SliderDualActing method **incrementDecrement:**

13] Connect your control object's sliderDA outlet to the sliderDualActing view. Now, for other instances of SliderDualActing, you can copy and paste this box, and only need to set your application's control object to the slider itself.

In your code, in the ™appDidInit method sent to the App add some initialization code:

You need to set the slider's UpTarget and UpAction and its various default values. An excellent place for this is after the Application receieves an AppDidInit: message. For example:

```
/* The app has an outlet  named "sliderDA" in this example */
- appDidInit:sender
 {
   [[[[sliderDA setUpTarget:self action:@selector(setAngle:)]
          setMax:360. allowHigher:NO min:-360. allowLower:NO]
          setAltStep:1. whole:YES default:0]
          setFormat:NO left:1 right:3]; // formats text pal

          /* other initialization code here */
          return self;
}
```

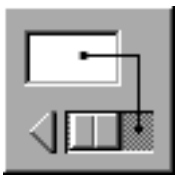
Caveats:

•Problem: IB has no inspector for subclasses of known objects in V1.0. This means you lose the ability to specify various defaults in IB.

Fix: Call the initialization routines illustrated above and documented in SliderDualActing.m.

•Problem: If the value of altStep is large compared to the range of the slider, drawing update anomalies occur. Instead of adding resolution, you subtract it. SliderDualActing should have more error checking for bad parameters.

The archiving methods have not been tested since I read mine in from a nib file.



Documentation and Files:

The documentation includes Class Specifications for SliderDualActing and SliderCellFine. I really enjoyed "cloning" the superclasses in WriteNow and just cutting and pasting my methods into the preformatted pages. It looks *just like the book*. I have a deep appreciation now for the great insight and hard work that went into producing the Appkit and its extraordinary documentation, definitely a winning aspect of the NeXT programming environment.

SliderDualActingDistribution (available in NeXT archives everywhere):

SliderDualActing.doc.wn

SliderDualActing.h

SliderDualActing.m

SliderCellFine.h

SliderCellFine.m

SliderCellFine.doc.wn

SliderDualActing.nib

SliderDualActingDemo.m

SliderDualActingDemo.h

Assorted:README,Makefile,IB.proj,snds,tiffs.

copyright 1990 Andrew C. Stone.

All Rights Reserved.

NeXT ANNOUNCES 68040 COMPUTER

via Conrad Geiger, NeXT inc, Seattle SE

Redwood City, Calif., May 9, 1990

NeXT, Inc. today announced plans to offer a version of its computerbased on the Motorola 68040 microprocessor in Q4 1990. In addition the company announced that all of its current customers will be able topurchase an upgrade to the 68040 for a suggested retail price of $1,495. Universities will receive a discount from that price.

NeXT will also ship its release 2.0 NextStep software at that time.

More than 60 percent of all workstations and 100 percent of allMacintosh computers are powered by Motorola's 68000 family of microprocessors. All current NeXT Computers use the 68030, the most powerfulmember of the 68000 family now in use. The 68040 is Motorola's-fourth generation of the 68000 family and is binary-compatible with itspredecessors, providing a smooth upgrade path for existing software.

"The 68040 offers NeXT's customers both RISC performance and personalcomputer affordability," said Steven P. Jobs, president and chairman of NeXT.

The 68040 incorporates the best of CISC technology (single chip, on-board cache and large software base) and of RISC technology (performanceand reduced clocks per instruction). In addition, its floating-pointperformance is up to ten times faster than the 68030.

Motorola claims that the 68040 outperforms all other CISC microprocessorson the market (including the Intel 80486) as well as many of the RISCofferings currently available (including SPARC) at similar clock speeds.

NeXT, Inc., of Redwood City, Calif., began shipping the NeXTComputer with 1.0 system software in September 1989 to customersin a wide range of business, government and academic environments. Thecomputer is available from Businessland throughout North America and theUnited Kingdom, from Canon Inc. in Asia and directly from NeXT forselected university and federal customers and authorized developers.

The <c-S,Q,@> Wars

Dick Silbar, Los Alamos National Laboratory

The following war story was originally going to be some remarks about how to use the NeXT Edit App more efficiently. Somehow, as you'll soon see, I got sidetracked into telling you about some troubles I have had with control characters. It is clear that much of this story has little to do with the NeXT *per se*, but perhaps it has *some* socially redeeming features. For example, maybe a sociologist could glean out of this that, despite a Jobsian effort on the part of the NeXT Corporation, computers are *still* too hard for most people to use.

Until about three years ago I did all my text editing using EDT (on Digital VMS miniframes) and TEDI, a locally-produced line editor (for all other computers I then used). Then someone (who might be referred to as an "AIgnu aboriginal") introduced me to Emacs. Among other things, this led to the eventual arrival in my office

of a sleek black cube named Whistler. The NeXT was my first real experience with UNIX machines, so at first I thought I would be involved in learning another editor, one at which I would be a no"vi"ce for some time.

Aha!, I soon learned that GNUemacs comes bundled with the NeXT as well as vi. A few experiments showed me it could be run in the Terminal App, though not in Shell. That is, the GNUemacs is fully functional as it comes out of the box, *as long as you don't touch the mouse*. There is even no trouble with the arrow keys acting as they should (i.e., as <c-F>, <c-B>, <c-N> and <c-P>), unlike some of the Emacs-like editors I have had to use. [The notation <c-F> denotes playing the keychord of the control-key and the F (or f) key, i.e., press both at the same time.]

There is a small annoyance about having to use the <esc> key as a <meta> key, despite the conspicuously available <cmd> and <alt> keys on the NeXT keyboard. In fact, because the Sun I (also) use has its <meta> key right where the NeXT has its <cmd> key, I sometimes forget and start to reformat a paragraph with a <cmd-Q> stroke. Which drops me out of the Terminal application altogether! How to fix that is another story for another time.

Unfortunately, I am by now very much addicted to using a mouse and I miss it badly. If something especially mousy needs to be done and it isn't too tricky a text, I'll switch and use the NeXT Edit App instead of Emacs. At first I thought it might not be too hard to write an App using the Interface Builder that would be a mouseable front end to the GNUemacs. That, however, is still a bit beyond me. There *is* some progress in this respect (by others) in this direction, and I'm sure there will be a usable "EmacsTool" right soon now. That is, maybe it's yet another story for another time, probably by a different author.

If something especially mousy needs to be done and it isn't too tricky a text, I'll switch and use the NeXT Edit App instead of Emacs

The present war story has to do with control keys, however. First, I have to tell you that one of the ways I use the NeXT in my office is by signing on from home, using my wife's PC-AT clone and modem and a terminal emulator package called VTERM. The convoluted way I did this, at first, is as follows:

1) Whistler is connected to my division's Ethernet LAN, which in turn is connected through a gateway machine to the LANL "backbone" network, and from there out to the wide world of telnet, news and ftp. All these things work nicely, provided of course that the Net itself is working nicely.

2) Thus, through the PC and modem at home, I can dial-up by phone to the LANL Integrated Computer Network (ICN), then connect to a gateway machine called P, from which (with the limited privileges I have on that machine) I can telnet to Whistler and sign on. "Good Morning, Dick!", says my .login file, even though this sign-on usually takes place after supper.

3) And, Emacs is then the editor of choice, since I certainly cannot use Edit and I haven't yet been really forced to learn vi. To work properly the NeXT needs to know it is talking to a VT100 (or some equivalent), so I also have the following lines in my .login file:

```
if ( $TERM != NeXT ) then
  set noglob
  eval `tset -Q -s -m 'dialup:vt100' -m 'network:vt100'`
  unset noglob
  endif
set term=$TERM
```

The trouble with working with Emacs this way is that, somewhere between the PC and Whistler I lose all control of the <c-S> key (needed for incremental-forward-search and, with <c-X>, for saving), the <c-Q> key (for quoted character), and the <c-@> (or <c-space>) key (for setting the mark). These are very important keys -- at least the <c-S> and <c-@> are. The reason is that <c-S> and <c-Q> are used (somewhere -- perhaps several places) for the Xon/Xoff protocol, while <c-@> is thrown away as a "useless" NULL character. I have been unable to find a clean way to pass these control characters through the ICN system. Or, for that matter, get anyone in our Computing Division interested in whether this is a problem that can be solved. That is, *this* battle has not been won.

There are, of course, ways to work around the lack of <c-S,Q,@>. First of all, you can do all of your work by typing out the command names after a <m-X> keystroke to bring up the extended command prompt in the minibuffer. Less drastic than that, but something which requires paying close attention, is that you can perform all your searches backwards (<c-R>) rather than forwards. You can save your file with <c-X><c-W>, but must be sure to get the spelling of the filename the same as you originally had it.[5] Or, you can take advantage of

the fact that when you quit GNUemacs (with <c-X><c-C>), it kindly asks if you want to save first.[6] Finally, you can set the mark by killing something innocuous.

A more permanent workaround is to alter the .emacs file to load in some substitute key bindings for isearch-forward and set-mark. These are

```
(global-set-key "." 'set-mark-command)
```

```
(global-set-key "\" 'isearch-forward)
```

which use <m-.> in place of <c-@> and <c-> in place of <c-S>. These are otherwise unused bindings and they do pass unscathed through the labyrinth of machines between the PC and the NeXT. However, this kludge has several disadvantages. First, you have to remember these bindings, and the fact that they are different from any other Emacs you might use. Second, the <c-> cannot be used successively, as the <c-S> binding dan, to find the next occurrence of the string typed so far. And, third, this <c-> substitute for the <c-S> won't work in combinations like <c-x><c-S>.

I'd call the outcome of that battle a tie, not a win.

In my next foray to send a <c-S,Q,@> from home to NeXT, I tried going through a *different* set of computers. Phoning in to a VMS-VAX I know and use, I was able to telnet directly over to Whistler. However, at least at first, these control characters were again trapped somewhere. Perhaps even for the same reason, since at least some of the gateway computers involved are the same. So, no win there, either. Funny thing, though; for a while (but not forever) after they upgraded the VMS operating system, the traps did not occur!

Well, this has been a long story but now it becomes short. It should have been something I thought of from the start. The clean solution to the passing of <c-S,Q,@> is to put Whistler on a modem and to call in *directly* from home. This way, there is no problem at all (once you set the VTERM software to turn off the Xon/Xoff protocol).

5.Filename completion, a <space> character in GNUemacs, is a help here, but it is a pain to havbe to remember whether completion uses a <space>, an <esc>, (or even, in Emacs I use on the Sun mentioned above, a <tab>!).

6.Actually, you may not really want to quit very often. For example, I like to do TeX (from home, on the NeXT) by saving the working .tex file (with <c-X><c-S>) and then putting Emacs into background (<c-Z>). After TeXing the file, looking at the mistakes I made, I then bring back Emacs (with "fg" after the UNIX prompt) without having to reload everything, i.e., with the working file already in buffer.

Well, not quite *no* problem. You see, the IBM AT that sits next to the NeXT at the office (one for which I am still developing software and which I therefore cannot get rid of) also must use that modem to get out to the outside world. That was solved with an A/B switchbox. Now the only thing I have to do, when logging off the AT, is to remember to switch the box back to A from B so the NexT knows to answer the phone. And, to power-off/power-on the modem to remove the residues of VTERM that haunt the modem otherwise and cause it *not* to answer. But these are annoyances one can almost live with.

So, the War of the <c-S,Q,@>'s is now more or less over. If I won anything here, I suspect it has only been by default.

What's up Dock?

Erica J. Liebman - with a lot of help, encouragement and support from _Jayson_Adams and Andy Stone

This is a story of failure -- not a bad story mind you, but a failure none-the-less. My goal was pretty simple : emulate the functionality of the black hole - but instead of deleting files, print them. I even created a lovely icon for it :

Anticipated execution was straightforward : drag an an icon from the browser onto DockPrint as it was running and bring up a dialog for printing via "lpr". The approach was simple too :

1. Register the Dock Icon with the Workspace Manager.

2. Catch events on the order of IconEntered (see below) and send out system requests for printing.

That was it. That's all. Let's see why it didn't work.

*There's another heirarchy of windows on the NeXT. These levels of priority are called **Tiers**.*

Each window on the NeXT has a certain priority with respect to other windows. In a straightforward application, you may change one windows precedence to another by ordering it front (making it "key", or of a higher

level) or ordering it back (making it a lower level, or even closing it).

There's another heirarchy of windows on the NeXT. These levels of priority are called **Tiers**. In the NeXT Glossary section of the online manuals these are defined as follows "The five sections of the screen list. Each tier is occupied by a different type of window, with attention panels in the frontmost tier, menus in the next two tiers, and docked icons in the tier below menus. All other windows are in the bottom tier".

There are actually six tiers. The top five tiers are for keeping important windows from being buried behind other windows of less priority when they are on the screen This is the order of the tiers. (I quote from the documentation)

1.Pop-up and pull-down lists

2.Attention panels

3.The main menu and attached submenus

4.Other menus

5.Docked icons

6.All other windows

Here's the problem. Dragged icons from the browser are at level 6 and docked icons are at level 5. It is impossible to drag these icons for printing *on top of* the docked icon.

Bummer.

Mind you, the program worked fine if the DockPrint icon was floating around the Workspace, it would just not run when in the dock

Well, I wrote frantically to Jayson, can I change the tier level of the docked icons? After a bit of conversation, we came to the consensus that No, for right now, it was virtually impossible without touching some software that really should not be touched. Mind you, the program worked fine if the DockPrint icon was floating around the Workspace, it would just not run when in the dock.

It is important to know how to conserve a failure. Here's what I did. I added on functionality which lets you decompress and de-tar as well as print. After all, if you have a spare icon wandering lonely among the Workspace you may as well make it useful. Instruction and Code follows.

INSTRUCTIONS

1. Create a "null-app" in IB. (no main window, fill in the INFO panel as desired.)

2. Save it into a fresh new directory as **Utilities** and create a project

3. Move UtilityObject.[mh] to the new directory. Add these to the Project's .[mh] files.

4. Cmd-5 to pull up the Classes browser. Subclass **Object** and add in **UtilityObject** as the new class. Parse in the new code.

5. Create an object instance. Connect via cmd-alt-drag the *File Own*er **to** *UtilityObjectInstance*. Select "**delegate**" and press **OK** in the inspector to complete the connection.

6. Compile & run. Deep problems to erica@kong.gatech.edu.

Discussion

It is interesting to note that most of the work done in this program occurs in the IconReleasedAt::ok: section. IconEntered:::..etc..:: merely logs the active files into the instance variable *iconPathList* and waits for someone else to handle actions.

For the most part, registering the window to the Workspace Manager (WM) is straight forward. First, the window number of the application icon is converted from its relative number within the application to the true global number that WM will recognize. WM is accessed via the application speaker who asks that messages to this window from other applications will be handled through his port. Note that this is all being processed in the delegate method of appDidInit that we arranged to be activated by the File Owner in step 5 of the instructions above.

NXRunAlertPanel allows a maximum of three buttons. In order to increase functionality, I looked for key letter endings of file names to select file compression, uncompression, de-tape-archival and printing. The user may press "cancel" if a file has been dragged over in error. You may, if you want, customize this to avoid the alert panel for smoother operation. There are times I myself find the alert panel annoying.

CODE : UtilityObject.h

```
#import <objc/Object.h>

#import <string.h>

#import <appkit/nextstd.h>

@interface UtilityObject:Object

{
```

```
            char *iconPathList;
}
-appDidInit:sender;
- (int)iconEntered:(int)windowNum at:(double)x :(double)y
    iconWindow:(int)iconWindowNum iconX:(double)iconX iconY:(-
double)iconY
    iconWidth:(double)iconWidth iconHeight:(double)iconHeight
    pathList:(char *)pathList;
- (int)iconReleasedAt:(double)x :(double)y ok:(int *)flag;
@end
```

CODE : UtilityObject.m

```
#import <appkit/graphics.h>
#import <appkit/publicWraps.h>
#import <appkit/Application.h>
#import <appkit/Listener.h>
#import <appkit/Speaker.h>
#import <appkit/Window.h>
#import <appkit/TextField.h>
#import <appkit/Panel.h>
#import "UtilityObject.h"

@implementation UtilityObject

-appDidInit:sender
{
           NXRect bRect;
           unsigned int wn;
           id speaker = [NXApp appSpeaker];

        NXConvertWinNumToGlobal([[NXApp appIcon] windowNum],
&wn);
           [speaker setSendPort:
            NXPortFromName(NX_WORKSPACEREQUEST, NULL)];
        [speaker registerWindow:wn
           toPort:[[NXApp appListener] listenPort]];

        return self;
}

- (int)iconEntered:(int)windowNum at:(double)x :(double)y
    iconWindow:(int)iconWindowNum
    iconX:(double)iconX iconY:(double)iconY
    iconWidth:(double)iconWidth
    iconHeight:(double)iconHeight
    pathList:(char *)pathList
{
    if (!iconPathList || strcmp(iconPathList, pathList)) {
        NX_FREE(iconPathList);
        NX_MALLOC(iconPathList, char, strlen(pathList)+1);
        strcpy(iconPathList, pathList);
    }

    return 0;
}

- (int)iconReleasedAt:(double)x :(double)y ok:(int *)flag
{
    char *file;
    char command[128];
    volatile int foundOne = NO;
   int which;

    NX_DURING
        file = iconPathList;
        [NXApp activateSelf:YES];
          if (file[strlen(file)-1] == 'Z')
          {
             if (NXRunAlertPanel(file,
             "File is of type .Z : Uncompresss?",
             "Uncompress", "Cancel",NULL) != 0)
          {
              sprintf(command,"uncompress %s",file);
              system(command);
          }
        }
          else if ((strlen(file) > 3) &&
           (file[strlen(file)-1] == 'r') &&
           (file[strlen(file)-2] == 'a') &&
           (file[strlen(file)-3] == 't'))
          {
             if (NXRunAlertPanel(file,
             "File is of type .tar : Untar?",
             "Untar", "Cancel",NULL) != 0)
          {
              sprintf(command,"tar xf %s",file);
              system(command);
          }
        }
          else if ((which = NXRunAlertPanel(file,
          "Please select utility to run",
             "Print", "Compress", "Cancel")) >= 0)
          {
               if (which == 0)
               {
                sprintf(command, "compress %s",file);
           system(command);
               }
               else if (which == 1)
               {
```

```
                sprintf(command, "lpr %s", file);
            system(command);
                }
            }
    NX_HANDLER
    NX_ENDHANDLER
    *flag = YES;
    return 0;
}
@end
```

Reviews, Rumors and Stuff: *a Myopic Eye into the NeXT Community*

Erica J. Liebman

TopDraw

I have received a copy of TopDraw and will be reviewing it in the next issue.

Rumors (planted) of Lighthouse Products

Some astonishingly good sources who took me out to dinner (as mentioned on page 1 -- you can guess who they are), have given me the dirt on some new products in development. First up is **BLT**, a diagramming tool. BLT stands for *Box & Line Tool*, not the more familiar sandwich. However, the Lighthouse Dudes (and one Dudette) have been using an astonishingly ugly icon of a rather destitute sandwich during the development cycle. Aw, c'mon guys.

Next up is **Exploder**, a CASE tool with a code-generating engine. It is supposed to extend Objective C to be persistant and automatically maintain some object relationships. The idea is that you draw pictures and get Objective-c out of it. Pretty good deal, if you don't start drawing some bees. Hopefully their work with Exploder will prompt our Lighthouse Guys (and one Guyette) to get us their Object Persistance article mentioned a few issues back! :)

Anyway, as usual contact rock@lighthouse.com if you're interested in these or just want to Betatest. Oh yeah, don't forget the 1st Public Domain compilation disk -- it's moving well and has had very good feedback from the assembled masses.

FTPs Sites abound

Here's the list of FTP sites I've tracked down

• **cs.orst.edu & j.cc.purdue.edu** - *the most popular archive sites for NeXT related materials currently used. There is a mail server at purdue :* ***archive-server@c-c.purdue.edu -*** *send mail with the subject line of "HELP".*

• **uvaarpa.acc.virginia.edu** - *has a lot of dsp related materials*

• **umd5.umd.edu** - *an official site, but not used much*

• **prep.ai.mit.edu** - *sacred GNU burial ground - none of it NeXT specific*

• **sutru.sfsu.edu** - *some activity lately*

• **linc.cis.upenn.edu** - *nethack sources -- patchlevel 7 compiles right off if you follow the instructions*

• **athena-dist.mit.edu** - *where to get X sources including NeXT-X*

• **cs.uiuc.edu** - *where to get Epoch (an emacs tool)*

• **ps-file-server@adobe.com or adobe!ps-file server** - *not really an FTP site - these are mail servers of files. Send a mail letter to one of these addresses with the subject line of "HELP"*

• **eesun1.arl.utexas.edu/129.107.2.51** - *new arrival, some nice stuff, but nothing especially innovative or new.*

• **sumex.stanford.edu** - *some interesting stuff*

•**winnie.princeton.edu** - *ditto, but supposedly some relating to MUSIC in particular*

• **dagon.acc.stolaf.edu** - *ditto again*

• **cica.cica.indiana.edu** - *some old NeXT update stuff*

• **csus.edu** - *some stuff*

PLANNED FTP ARCHIVES :

People with BCS, Washinton Apple Pi and BaNG have all expressed interest in starting their own sites.

ADOBE DEVELOPER CONFERENCE

Adobe is holding its first Developers Conference 5 - 8 June in San Jose California. They'll be talking about PostScript Level 2 in particular. This is the first major revision in the Postscript langauge.

"Quoting the information sent me :

Benefits of PostScript Level 2 :

• *Improved Performance*. We have increased the speed fo the PostScript interpreter, including techniques that reduce time of transmission of PostScript language programs, and added langauge constructs that speed up processing.

• *Enhanced Functionality* We have included langauge extensions from the Display PostScript system and improved forms handling, color support and pattern manipulation.

• *Convenience for applications developers* Application developers are the main "users" of the PostScript langauge, so we've worked with our ISVs to incorporate features that will make their programming jobs easier.

• *Compatibility with Display PostScript* PostScript Level 2 includes several features from the Display PostScript Sytem that enhance compatibility between display and printer environments.

• *Improved memory management.* We have eliminated arbitrary implementation restrictions on memory usage as well as providing for explicit and automatic memory reclamation"

Speakers will include Steve Jobs. For conference information, call 415-398-5352.

Market View

New This Month

THIS SECTION IS PROVIDED AS A COURTESY TO OUR READERS AND THIRD PARTY FRIENDS. BuzzNUG IS NOT RESPONSIBLE FOR THE CONTENT OF THESE OR ACCURACY OF INFORMATION PRESENTED IN THIS SECTION

• WINGZ has been officially announced and is now shipping. Some Businessland Pals have reported that its slow to get to the shelf, but it is definately official.

• MetaResearch is now shipping **DIGITAL EYE**. This is a hardware and software system that "allows users to enter and record still or moving NTSC video images on the NeXT (tm) Computer.

"'The applications for Digital Eye are essentially unlimited' said Lee Buhler, president of Metaresearch. 'In combination with still or VCR cameras, Digital Eye can turn the NeXT Computer into an electronic darkroom, enhance the full range of desktop publishing applications or aid in any visual storage or archiving activities."

Digitial Eye comes with two software packages : SNAPSHOT for still pictures and MOVIEWORKS for moving images.

We hope to get review equipment from MetaResearch and the update to Digital Ear's software, which we are promised crashes far less than the highly Beta copy we reviewed a few months ago.

• Imagagine Inc is shipping **MediaStation**, a multi-media software package for the NeXT. "As one fo the first applications built from the ground up specifically for the NeXT Computer, MediaStation takes full advantage of the computer's unique features...(It)combines powerful database features; high-resolution image scanning and processing; CD-quality sound recording and editing; frame-by-frame animation; and text processing to create multimedia archives.

"Most multimedia software available today is narrowly aimed at the desktop presentation/animation market. MediaStation's powerful searching, organizing and report generating features, extend its productivity beyond the desktop presentation market into the broad field of information management. It can function as a multimedia file server over a high-speed ethernet network with password protected, multi-user access to the same files".

We, again, hope that Imagine Inc will get us a review copy for an upcoming issue of the Newsletter.

Product Listings

I've received literature from NeXT or Third Party Developers on the following products. No warranty, express or implied, is given. The quotes are mostly the developers' and may not reflect reality. Please send up-to-date info and review copies for new products.

•Digital Instrumentation Technology Inc (505)662-1459 is shipping beta copies of **Cube Floppy 1.4**, a 3.5" floppy disk drive connecting to the SCSI port, reading and writing MS-DOS(720K and 1.44MB disks), UNIX and Macintosh (1.4MB disks) file formats.

• WeDesign Inc (415)-479-1105 sent a flyer on **TheLibrary**., an on-line information system with Objective C references and authoring tools.

• Pacific Micro (415)948-6200 is shipping the **PM1.44** and **PM HDE** external 3.5" floppy drive and hard disk enclosure. Special order line is 1-800-628-DISK.

• BYTE's **BIX** service has a special NeXT interest group. Contact 1-800-227-2983 (BIX Customer Service) if interested. Please mention both BuzzNUG and Dave Andrews (of Byte) as contact names.

•Communicae - Active Systems 1-617-576-2000 "high performance communications package. VT240 emulation" *SHIPPING*

•Wingz Informix Software, Inc. 1-913-599-7100 "graphic spreadsheet featuring advanced charting, desktop presentation capabilities, and HyperScript" *SHIPPING*

•Scan 300/GS Abaton 1-415-683-2226 "300 dpi flatbed scanner with TIFF compatility"

•DM-N Digital Microphone Ariel Corporation 1-201-249-2900 "software-selectable sample rates from 88.2 kHz to 5.5 kHz per channel" *SHIPPING*

• BUG-56 : DSP debugger, Ariel Corporation 1-408-982-0400/1-201-249-2900, *SHIPPING*

•DaynaFILE Dayna Communications, Inc. 1-801-531-0600 "external, SCSI floppy disk drive to write to standard UNIX-formatted diskettes, as well as MS-DOS formats" *SHIPPING*

•Smart Art Emerald City Software, Inc. 1-800-223-0417 "50 text and graphics effects and easily customized in any NeXT word processor, desktop presentation, or page layout program *SHIPPING, -- I GOT A THIRD PARTY LISTING OF THIS FROM NeXT WHICH SAYS THIS IS NOW OFFERED BY ADOBE! (415)962-2045*

•FrameMaker 2.0 Frame Technology Corporation 1-408-433-3311 "powerful, cost-effective workstation publishing software" *SHIPPING, RECOMMENDED FOR DESKTOP PUBLISHING, PAGE LAYOUT FOR IN-HOUSE PUBLICATIONS, BROCHURES.*

•Artisan Media Logic Incorporated 1-213-453-7744 "high-resolution paint and image processing system"

•TopDraw Media Logic Incorporated 1-213-453-7744 "complete and advanced page-based graphics software" *SHIPPING*

• HSD incorporated US Scan-X 1600/600 415-964-1400. Scanners for “line art and grayscale” *SHIPPING*

•TextArt Stone Design Corporation 1-505-345-4800 "array of tools that allow immediate creation of outstanding PostScript images" *SHIPPING, RECOMMENDED FOR “SPLASH” APPEAL, PROGRAMMING WITH ICONS.*

•Encapsulated PostScript ClickArt T/Maker Company 1-415-962-0195 "combines ClickArt EPS portfolios into a collection of high-quality Encapsulated PostScript (EPS) artwork" *SHIPPING*

•Public Domain Disk #1 - Lighthouse Design 1-800-FOOBAR9 "Public Domain Software & More" (I'm in on this one. Buy it. Please!) *SHIPPING*

•Scematic Entry - Lighthouse Design 1-800-FOOBAR9 "CAD Tool for designing electrical circuit schematics"

•Media Station - Imagine Inc 1-313-434-1970 "archival, retrieval and processing of multi-media information" *SHIPPING*

•Fortran 77 -- Absoft 1-313-853-0050 "Objective Fortran-77"

•DisplayTalk - Emerald City Software - 1-800-223-0417 "Complete development environment for Display PostScript programming" *SHIPPING-- I GOT A THIRD PARTY LISTING OF THIS FROM NeXT WHICH SAYS THIS IS NOW OFFERED BY ADOBE! (415)962-2045*

• *SmartArt*, Adobe Systems Inc. 415-962-2045 “Graphics and headline type effects using Display PostScript”.

•Video Monitor and Projector Interfaces Extron Electronics 1-800-633-9876 "offers three video monitor and projector interfaces"

•Digital Ears Metaresearch, Inc. 1-503-238-5728 ""allows entering and recording compact disc-quality sounds" *SHIPPING, REVIEWED IN ISSUE 4*

•Digital Eye Metaresearch, Incorporated 1-503-238-5728 "allows entering and recording NTSC video images"

•NVT High Density Video Drive New Vision Technologies, Inc. 1-415-285-8744 ""video playback device for interactive multi-media applications"

•JETSTREAM Tape Backup System Personal Computer Peripherals Corporation 1-813-884-3092 "high performance tape backup system"

•A/D64x Analog/Digital Interface Singular Solutions 1-818-792-9567 "a low-cost platform for sound recording, experimentation, and analysis"

•Who's Calling Adamation, Inc. 1-415-452-5252 "lets sales & business professionals keep track of phone calls and other client information" *SHIPPING, THEIR BROCHURE IS REALLY COOL, THIS IS ONE I WANT TO REVIEW.*

•GEMS (Generalized Equilibrium Modeling System) Data Transforms, Inc. 1-303-832-1501 "a flexible way to model economic systems"

•InDia (Influence Diagram Processor) Data Transforms, Inc. 1-303-832-1501 "graphical application for representing complex decision-making"

•Knowledge Retrieval System (KRS) KnowledgeSet, Corporation 1-415-968-9888 "rapidly searches and retrieves information from large databases of text and graphics"

•OMEN III Microstat Development Corporation 1-604-228-1612 "stock quotation and financial system"

•TACTICIAN Plus SouthWind Software, Inc. 1-316-636-5100 "multi-user spreadsheet that supports high-level functions and adds built-in presentation graphics" *I’VE SEEN A DEMO VERSION, BASIC SPREADSHEET FUNCTIONALITY -- IT’S IN THERE.*

•Adobe Illustrator Adobe Systems Incorporated 1-415-961-4400 "graphic design and illustration program for generating high-quality artwork"

•Adobe Type Library Adobe Systems Incorporated 1-415-961-4400 "offers more than 500 different typefaces" *I WANT A COPY OF THIS!*

•Flash Graphics Flash Graphics 1-415-331-7700 "extensive charting, illustration, and text functions in a graphics package for screen, slide and paper presentations"

•InterFax 24/96N Abaton 1-415-683-2226 "combines a 9600 bps Group 3 fax modem with a 2400 bps MNP 5, Hayes-compatible data modem"

•GatorBox Cayman Systems, Inc. 1-617-494-1999 "LocalTalk to Ethernet gateway that translates the Network File System (NFS) protocol into Apple Filing Protocol (AFP)" *SHIPPING*

•MacLinkPlus/PC DataViz Inc. 1-203-268-0030 "kit for transferring and translating files between NeXT and Macintosh environments" *SHIPPING*

•Ethernet PhoneNET, Sound and Interpersonal Communications Farallon Computing, Inc. 1-415-849-2331 "used to build LANs over standard telephone cables"

•Etherport NL Kinetics 1-415-947-0998 "allows the NeXT computer to connect directly to standard twisted-pair Ethernet networks"

•INFORMIX-TURBO Informix Software, Inc. 1-415-926-6300 "database engine for on-line transaction processing (OLTP)"

•INGRES Relational Database Management System Relational Technology, Inc. 1-800-4-INGRES "SQL database engine provides on-line transaction processing (OLTP) in single- or multi- CPU and distributed environments"

•DAN - The Data Analyzer Triakis Inc 1-505-672-3180 "data analysis package for reducing data and generating presentation-quality plots"

•Math++ - Triakis Inc 1-505-672-3180 "C-language math library. Approx 100 math functions"

•Dreams - Innovated Data Design 1-415-680-6818 "Frm the makers of MacDraft, drawing and drafting tools"

•Cross Assembler/Simulator Programs - Motorola 1-512-891-2030 "for the 56000 and 96000"

•Fortran, C and Pascal Compilers - OASYS 1-617-890-7889 *SHIPPING*

User Groups

Here are some pointers to Users Groups that may be in your area. Send info if you don't see your group here.

NEW THIS MONTH : A COLORADO UPDATE

Maryland/Northern Virginia/DC

Washington Apple Pi,

Hugh V. O'Neill, (301)-328-9510, Chairperson

POB 39036, Washington, DC 20016

The purpose of the SIG is to exchange information from an end-user's viewpoint on the capabilities (present, planned, and potential) of the NeXT computer/information system. The goals and objectives include the following:

1. Presentation of relevant, timely, accurate and complete information on the performance characteristics, hardware, and software of the entire system.

2. Discussion of applications in various areas including: education, research, medicine, law, decision making/management, policy making/analysis, science, engineering, mathematics, statistics, humanities, arts, business, governmental activity (federal, state, and local), executive information systems, artificial intelligence (AI), operations research/management science, systems analysis, desktop publishing, office use and home use.

3. Cost-effectiveness/benefits/performance evaluation from the viewpoint of the end-user.

4. Special topics such as networks, information system security/integrity, speech recognition, signal recognition, array processing, digital signal processing, and sound/music.

The NeXT SIG usually meets at 7:30 p.m. on the 2nd Wednesday of the month at the National Institutes of Health (NIH) in Bethesda. In addition, we have special joint meetings with other organization groups and SIGs at mutually acceptable times and places.

[Thanks **Keith & Lisa** for getting this to me! Hugh also mentioned on the phone that he's going to try to put together a Newsletter focussed on the Applications end-user.]

Phil Phuster ffuster@next.com is listed as another DC contact.

Georgia

BuzzNUG is sponsoring local demos and talks. Contact **Erica Liebman** at erica@kong.gatech.edu. (404)-352-5551 We **desperately** need a local activities organizer. Please contact me if you are interested in pitching in for things Buzz

BCS puts out a newsletter called WhAT's NeXT*.* ***Boy*** *would the Buzzies Staff love a complimentary subscription!*

Massachusetts

The ***Boston Computer Society*** (BCS) has a NeXT special interest group. Contact **Dan Lavin** at 1-617-969-6555, or **Jan McPeek** at 1-617-926-4027. BCS's NeXT SIG has 750 members. Contact Dan at BCS 1 Center Plaza, Boston, Mass 02108. They've been meeting monthly since January 1989 and have a Bulletin Board and NewsLetter aimed at the Business Community. There are member fees. BCS puts out the newsletter called *WhAT's NeXT.* **Boy** would the Buzzies Staff love a complimentary subscription*!*

Erik Kay reports via Conrad Geiger: BCS NeXT is an affiliate of BCS which has members in all 48 continental states and many foreign countries. The leader of the NeXT group is Dan Lavin. I've been working closely with Dan to set up a combined MIT/BCS NeXT FTP archive server (yes, I know there are 3 big ones already, but I think that because of BCS and MIT, we will have a lot to offer. We'd like to make it sort of a developers archive site, with the GNU source, lots of programming tools and classes, X Windows, etc.). Dan's phone number is (617)969-6555 if you'd like to get in touch with him yourself. The user group has a fairly large turnout which ranges from an average of 30-50 and has exceeded 100 at times. Partly because of it being BCS, it has attracted many commercial developers to come talk. People who have been there have included Cayman, the people who make the Scan-X scanner, Media-Logic, Stone Design (Andy Stone came himself), the people who do Digital Eyes/Ears, and others. ... Next month's meeting will have WingZ making it's world product release announcement. Erik Kay

.*California*

Robert D. Nielsen, nielsen@everest.sjsu.edu, 1-408-995-5775 coordinates **BaNG**! out of San Jose, loosely associated with Stanford. Robert is the San Jose State University NeXT Campus Consultant. (Atta-boy!) Definitely call if you live within fifty miles of San Jose. Or are willing to drive further. Or have your own 'copter. A newsletter is planned. ***See earlier article about the first meeting.***

Paul Lowe (714)787-3883 at the University of California at Riverside is interested in seeing what others are doing with their NeXT Cubes to distribute to the other NeXT Users (six so far) on campus. write to: plowe@ucrac1.ucr.edu

Texas (yee-hah!)

The report goes that there "is some massive new NeXT user group down there". Says **Jerry Goode**: "Hi there! Just wanted to get back to you with some info about the NeXT user's group down here in Dallas, Texas...

here is the name of the fellow who is heading up the group down here: **Dirk Hardy**/Hofbauer Information Systems/5080 Spectrum Drive/Suite 912W (Lock Box 21)/Dallas, Texas 75248/Phone: 214-385-2991 *(Late breaking news, additional contact name, no guarantees:* ***Charlie Lindahl - Dallas- Ft. Worth Texas....lindahl@evax.arl.utexas.edu***)

Hofbauer is a NeXT registered developer doing courseware authoring tools, and Dirk is working in conjunction with Dr. Ali from North Texas State U. to get the group going. I encourage you to get in touch with him - he's got a lot of creative ideas! Hope to have an email address for him soon.....Finally, we plan to meet the third Thursday of every month at 7 pm, somewhere! The logistics are still a bit up in the air as we try to figure out how much this thing will grow. If the response after our first meeting is any indication, we could settle in at around 50 people. By the way, I have had Buzzings forwarded to me and it's GREAT. Dirk has copies of both issues so far and was really impressed. You may want to talk with him about organizing a Texas contingent contribution as we try to get things rolling here. "

Washington (the state)

University of Washington (with over 110 cubes) has a NeXT User group dating back to March 1989. Contact **Corey Satten** (corey@cac.washington.edu).

Oregon

It looks like Bryce may be starting something up there, contact: **Bryce Jasmer** at jasmerb@cs.orst.edu

Minnesota

Scott Hess,scott@gacvax1.bitnet says: "I was wondering if you have had anything about the Minnesota Next User's group in Buzzings. I can talk to them and see what they want to put in, as we have our March meeting on Tuesday (the 12th). It would be nice to get our name out

there". *Hopefully, there will be more information, next issue.*

Canada (O!)

Vancouver NeXT User's Group (in BC, CA) is headed up by **Lionel Tolan** (lionel_tolan@cc.sfu.ca). **Tom Poiker** of Simon Fraser Universit is starting a new newsletter NeXTVieW (name tentative) with **Shirley Chan**. (poiker@whistler.sfu.ca) Tom's particular interests are geographic imaging on the NeXT. [Pretty cool, eh?] The first issue is out and available via the nets -- a very nice job, focussing on the applications level of NeXT use. **Tom, by the way, owes me a review of TopDraw!**

Colorado

The NeXT users in Colorado have just formed the **Rocky Mountain NeXT Users Group** (**rmNUG**). The first meeting was held in Boulder on April 11th. The users came from a varied background including NCAR, UCAR, University of Colorado (several sites), Colorado State University, Colorado School of Mines, US West, Hughes, Martin Marietta, etc.

*The NeXT users in Colorado have just formed the **Rocky Mountain NeXT Users Group** (**rmNUG**). The first meeting was held in Boulder on April 11th.*

Both Mitch Green and John Pierce of NeXT Inc. were on hand to show their support, discuss things with the users and provide great demos.

Our next meeting will feature Glenn Davis of UCAR demonstrating his Interface Builder application which displays and manipulates weather satellite data/pictures

Our meetings will be held in various places and feature the different members as they give presentations on what they are currently developing.

David Hieb (davehieb@boulder.Colorado.EDU, (303)492-5720) is the leader of rmNUG and is a System Administrator for the Computer Science Department at the University of Colorado, Boulder.

Brad Green (green_bk@cubldr.Colorado.EDU, (303)786-0081) is the NeXT Campus Consultant for the University of Colorado, Boulder, and provides support and leadership for rmNUG as well.

We plan on providing an environment for the NeXT users in Colorado to share, learn and profit from the collective expertise of the group.

p.s. To all the NeXT users out there in Colorado that haven't been contacted by one of us : CALL ME at (303)492-5720 or email me at davehieb@boulder.colorado.edu

---

Utah

Mitch Green of NeXT reports of the emerging Salt Lake City group, "The contact in Salt Lake City is probably going to be Tom Pier, but I don't have an email for him right now."

Buzz's Hint Corner

- Use the *open <filename>* command (typed) in the Browser dialog that opens under the main browser in the Workspace to simulate double-clicking on that file. If Edit is *already* running, say open <*filename*> and it will automatically be loaded into Edit - *Dave Joerg*
- Run any application with "<appname> -NXShowPS >& <somefile>" on the command line, and all the PostScript output of your program (ALL of it, a lot) will be dumped to "<somefile>". For even more fun, view this file with Preview (Yap and Scene don't work out as well) and you will see an instant replay of everything that happened in your application! *Dave Joerg*
- Are you getting your NeXT mail on a Sum? Just extract each program as described in previous hint. For the tiff picture, I converted the tiff file using Jef Poskanser's excellent PBMPLUS package. The proper incantation is

```
tifftopgm < Zings5.tiff | pgmtopbm | pbmtoxwd | xwud
```

I tried converting to PostScript, but there appears to be a bug in the PBMPLUS stuff somewhere. The image appears sheared somehow, both when viewed using Sun's pageview program, and on a LaserWriter.

For voice mail on a SPARC send the vox file to the audio device.

e.g. cat VoiceMail_erica5.vox > /dev/audio

There's no volume control with that method, but there are programs around to manipulate the audio device in various ways. The audio requires a SPARCStation-1. Everything else is generic Sun. - *Skip Montanaro*

- In WriteNow, you can get it to automatically put create evenly spaced "leaders" between tabstops. That is, suppose you wanted to create a table of contents that looks like this (very much like the first few BuzzNUG issues):

AI/X and NextStep 48

Buzz's Hint Corner 53

The way to do this is to type the first part. Then, hit the TAB key, BUT DON'T LET GO. While holding the tab key down, hit the period key and WriteNow will automatically fill in the periods for you. You can do hit any printable character on the keyboard while holding the tab key down and WriteNow will use that character. *Joel McClung,(joel@next.com),Systems Engr, NeXT, Inc.,Washington, DC.,(703) 938-NeXT*

• In FrameMaker, shortcuts for viewing Next Page and Previous Page are cmd-7 and cmd-6 respectively. When using these shortcuts, pages will always align exactly and neatly- even if you have scrolled up and down a bit. *EJL*

NeXT Support Answers,
Care of Doug Kieslar

Speaker Listener parameters byte array (QA486)

Q: I've tried everything, and I can't seem to get the Speaker/Listener connection to work correctly for byte-arrays. I can transmit null-terminated strings, but byte-arrays just won't work. I even read all the documentation for the Speaker and Listener classes. How does this work?

A: The documents are wrong. The man page on msgwrap has the solution, though. You need to declare the parameter as (byte *) IN YOUR .MSG FILE, but keep the parameter declared as (char *) in your method declarations. Note that you still should follow every (byte *) with a (int) that specifies the length of the byte array. Note also that, to receive byte-arrays, you should declare the parameter as (byte **), and follow it with a (int *), but should leave the declaration as (char **) for your methods.

Submenu menu item (QA494)

Q: How do I get a handle on the submenu of the current menu item? I read the documentation, and there appears to be no way to access the submenu to be called from a given menu item. It's easy to find out that the item HAS a submenu, but appears impossible to find out what object that submenu actually IS.

A: If [aMenuCell hasSubmenu] is true, [aMenuCell target] is the id of the submenu.

Speaker Listener parameters port rights (QA487)

Q: Why doesn't msgwrap let me declare a parameter as type (port_t) or (port_t *)? I thought Speaker/Listener was designed to let me pass around port send and receive rights easily.

A: You need to declare the parameters as (port_t_send) and (port_t_send *) respectively, IN THE .MSG FILE ONLY. Keep your method parameters set to (port_t). Of course, if you are giving or taking port receive rights, your .msg file should have (port_t_rcv) or (port_t_rcv *). Don't forget to use port_deallocate() when you're done using a port!

Submitted by: Dave Joerg, 708.879.1500, joerg@alliant.mcs.anl.gov

Pasteboard timeout paste board time readTypes (QA463)

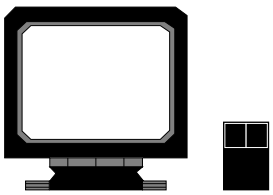
Q: The Pasteboard sometimes returns nil when I ask it for data--what might this mean?

Q: Should I check the Pasteboard return value when asking it for data?

Q: My program crashed when I pasted from the Pasteboard--what might have gone wrong?

A: When you use the readTypes:data:length: method to get data from the Pasteboard, the inter-process communication done to get the data might time out. This would normally happen if the copier put the data in the Pasteboard lazily, but then was very slow in providing the data when another app finally did the paste.

When such a timeout happens, the readTypes:data:length: method will return nil. Apps should check this return value, and probably put up some alert about not being able to get the data, since the copier is either slow or dead, and that it might work if you try again later. NOTE: Apps should NOT try to use the data returned, since they will indirect through a nil ptr and coredump!

You should check the return value for another reason: someone could put something new in the Pasteboard between the time you enquire the type list and ask for data. In this case your data is no longer there, and you'll get a nil back. This is documented in the spec sheet at least

.