- ana-systems (415.341.1768):
Documenter's Workbench (troff, etc.)
- Design Technology (info@bydesign.com):
OO-Browser (multi-language object-oriented class browser)
- Information & Communications (800.776.9966):
AKGH (context-sensitive on-line help tool)
- Lighthouse Design (800.366.2279):
Exploder (object-oriented database)
Schematic Entry (EE-CAD tool)
- Motorola (512.891.2030):
SIM56000 (DSP simulator program)
- Objective Technologies (212.227.6767):
[OT Palettes 1.0] (GUI object library/palette)
- ONyX Systems (817.468.2695):
BugByte (graphic interactive symbolic debugger)
- Professional Software (617.246.2425):
Objective DB Toolkit (obj. library, database support)
- Informix (913.599.7100):
Math++ (C-language math library)
- WeDesign Inc (415.479.1105):
TheLibrary (on-line help information system)

MEDICAL
- BioMedical Design Group (612.645.9062):
BioTRACE 8 (8 channel acquisition and monitoring)
- LMER (717.531.7589):
Gray's Anatomy-digital (digital version of Gray's Anatomy)
- Toltec Human Services (405.840.4254):
Transcriber (medical record transcription tool)

OTHER
- Alva Computer (804.851.6830):
Airfoil Design Kit (aerodynamic engineering tool)
- BIX (800.227.2983):
NeXT special interest group
- Boylan Enterprises (404.271.8305):
NeXUS Magazine: (NeXT oriented magazine)
- Baran's Tech Letter (208.265.5286, nbaran@well.sf.ca.us)
(newsletter)
- Deltos Fleet Computing (214.540.2301):
GeoKit (object-toolkit for cartographic map rendering)
- Halchin and Fleming (217.348.0917):
Spring (differential equation modeling program)
Taylor (Taylor polynomial software)
- Insignia Solutions (408.522.7600):
SoftPC (IBM-PC/AT MS-DOS emulator)
- International Data Group:
NeXTWORLD (magazine)
- Lighthouse Design (800.366.279):
Public Domain Disk #1 (asstd pd software)
- NeXTConnection (800.800.6398)
(NeXT-specific mail order)
- public domain:
iwf (ImageWriter II printer-driver)
(cs.orst.edu)
JumpBack (backup facility)
(sonata.cc.purdue.edu)
News (NeXTstep newsgroup news-reader, Beta) (cs.orst.edu)
- shareware:
iwscript (ImageWriter II printer-driver)
(sutro.sfsu.edu)
Stuart (full-featured NeXTstep VT100 terminal app) (cs.orst.edu)
- Software Engineering Solutions (512.343.2828):
AtTheBeep (NeXTstep message handling system)

#1 Football Champeens of the World!

HARDWARE
- Ariel (201.249.2900):
QuintProcessor (multiprocessor DSP NeXTBus board)
DM-N Digital Microphone
- Daewoo Telecom (508.836.4800):
Daewoo Graphics & Imaging System (image processing board)
- Dazzl (309.674.9317):
16/12 Analog to Digital Converter (NeXTBus board)
- DIT (505.662.1459):
CubeFloppy 2.9 (floppy, Mac file transfer software)
CubeDigital I/O, SCSI488/N (NeXTBus boards, controllers)
- Extron Electronics (800.633.9876):
Extron Board (NeXTBus large screen projector board)
- IOtech (216.439.4091):
ADC488, DAC488 (A to D, D to A converters)
Digital488/80, Serial488/4 (IEEE to digital, IEEE to serial)
Driver488/N (IEEE488 Obj-C driver)
- Pacific Microlectronics (800.628.3475):
PMHIDE (external SCSI hard disk drive enclosure)
- Pers. Comp. Periph (813.884.3092):
JETSTREAM Tape Backup (high-perfromance tape backup)
- PLI (800.288.8754):
SuperFloppy 2.8, Mach One (floppy, high-speed hard disk drives)
- Singular Solutions (818.792.9567):
A/D64X (analog-digital interface

BUSINESS
- Adamation (415.452.5252):
Live Wire, Who's Calling (business info management)
- Boss Logic (515.472.7740):
Contact! 1.0 (business contact database)
- Chadwyck Healey (800.752.0515):
MundoCart/Optical (cartographic map database)
- DIT (505.662.1459):
OnDuty (office task management)
- Stained Glass Software (408.249.3337):
Calendoscope (appointment calendar program)
- TransGraphics Systems (415.283.5750):
TransManager (terminal management software)

PROGRAMMIMNG LANGUAGES
- Absoft (313.853.0050):
FORTRAN77 (with object oriented extensions)
- Acucobol (619.271.7097):
ACUCOBOL-85 (COBOL)
- ana-systems (415.341.1768):
Modula-2
- Franz Inc (415.548.3600):
AllegroCL (Common Lisp)
- GNU (prep.ai.mit.edu):
C, C++, Objective-C, Smalltalk
- Interactive Software Engineering (eiffel.com):
Eiffel
- Iverson Software Inc (416.925.6096):
J
- Jefferson Software (602.234.3106):
Modula-2
- Motorola (512.891.2030):
DSP 56000 Assembler
- Oasys (617.890.7889):
C, FORTRAN, Pascal
- Plus Five Computer Services (314.426.3900):
MUMPS
- public domain:
FORTRAN-to-C converter
Standard ML (princeton.edu in pub/ml)
- Toltec Human Services (405.840.4254):
UNIX MUMPS
- Univ. Va (804.982.2209):
Uvapc (Pascal)

DEVELOPMENT
- Adobe Systems (415.961.4400):
Displaytalk (PostScript development environment)
- Ariel (201.249.2900):
Bug-56 (DSP debugger)

4010/4014 emulation)
- Asante (408.734.4844):
TR/SC (NeXT (SCSI) to Token-Ring)
- Atlantix (800.262.6526):
XWave (integration software, NeXT to PC LANs)
- AVATAR (508.435.3000):
InSession 3270 (IBM 3270 connectivity)
- Cayman Systems (617.494.1999):
GatorBox, GatorMail, GatorShare (NexT to Mac net)
- Conextions (508.475.5411):
3270Vision (IBM 3270 connectivity)
- DataViz (203.268.0030):
MacLinkPlus/PC (NeXT to Mac file transfers)
- HSD (415.964.1400):
OCR (fax software-scanner)
- Morning Star Technologies (800.451.1883):
Synchronous SCSI Communication Interface
- Novell (408.434.2300):
Novell NetWare integration software
- Objective Software Engineering (604.261.0186):
Mirage Fax (fax modem)
- Software Ventures (415.644.3232):
MicroPhoneII, WatchMe, ScriptEditor (telecommunications)
- Touch Communications 408.374.2500):
Worldtalk/400 (X.400 messaging gateways)
- Transarc (412.338.4400):
AFS (distributed file system)

COLOR
- Analytical Graphics (215.337.3055):
STK (Satellite Tool Kit, satellite planning)
- HSD (415.964.1400):
Scan-X Color (color scanner)
- Tektronix (503.221.1063):
Phaser PX (color printer, serial, parallel and AppleTalk ports)
- Oce (800.545.5445):
OceColor (PostScript printer)
- QMS (415.363.4300):
ColorScript 100 (color PostScript printers)

GRAPHICS
- McGill Univ (peterd@cs.mcgill.ca):
X windows 11R4
- Pencom Software (512.343.1111):
X windows 11R4
- Ohio Supercomputer Center (614.292.0006, michelle@osgp.osc.edu):
apE (visualization and image processing package)

SOUND
- Univ. Virginia (uvaarpa.acc.virginia.edu):
CSound-NeXT v2.0 (digital synthesis software interface)

MULTIMEDIA
- Ariel (201.249.2900):
IRCAM (multi-RISC NeXTBus board for vector/signal processing engine)
- Boss Logic (515.472.7740):
Boss DMS (document management software)
- Coda Music Software (800.843.2066):
MusicProse (music notation software)
- DisCopyLabs (415.651.5100):
software duplication services in 2.88 MB format
- Icarus Software (617.247.1350):
LiNK-IT (hypertext system)
- Imagine, Inc (313.487.7117):
MediaStation, MediaBrowser (multimedia database)
FirstChair (MIDI and sound sequencer)
- Knowledge Transfer Intl (214.233.7693):
SimuLEARN (instructional multimedia tool kit)
- Metaresearch (503.238.5728):
Color Digital Eye (high-res. color frame grabber)
Digital Eye, Digital Ears (gray-scale frame, sound grabber)
- NVT (415.285.8744):
High Density Video Drive (video playback device)
- Thoughtful Software (303.221.4596):
HyperCube (hypermedia environment)
- Visus (412.687.3800):
Fax DEMON, PaperSight (fax, document image management system)

AVAILABLE

Chuck Herrik

The following is a partial listing of products for the NeXT. This is not an official publication of NeXT Computers, Inc. and no warranty, either express or implied, is given. Up-to-date info is always welcome. Copies for review are encouraged.

ANALYTICAL
- Affinity Systems (316.636.5100):
Tactician Plus (multiuser spreadsheet)
- Ashton-Tate (800.437.4329):
PowerStep (spreadsheet)
- Informix (913.599.7100):
INFORMIX-TURBO (database engine for OLTP)
Wingz (spreadsheet)
- Ingres (800.446.4737):
RDBMS (database)
- KnowledgeSet Corp (415.968.9888):
KRS (database search and retrieval)
- Lotus (617.577.8500):
Improv (spreadsheet)
- Microstat (604.228.1612):
OMEN III (investment information retrieval and manipulation)
- Oracle (800.345.3267):
RDBMS (database)
- Professional Software (617.246.2425):
Objective DB Toolkit (30 classes link NeXT to Sybase)
- SAS (919.677.8000):
statistical analysis and presentation software
- Stone Design (505.345.4800):
DataPhile (database)
- Sybase (800.879.2273):
SQL Server (database)
- Triakis (505.672.3180):
DAN (data analysis and plot system)
- Wolfram Research (217.398.0700):
Mathematica (mathematical analysis

PUBLISHING
- Abaton (800.444.5321):
Scan 300/GS (scanner)
- Adobe (415.961.4400):
Illustrator, Plus Pack, SmartArt
- Altsys (214.680.2060):
Stealth (advanced drawing, graphic arts)
- Canon (800.848.4123):
IX-30F Image Scanner (scanner)
- Data Transforms (303.832.1501):
InDia (graphical decision-making)
GEMS (flexible modeling system, ex: economics)
- Epitome(615.675.0910):
Redline (multiple reviewer)
- Flash Graphics (415.331.7700):
Flash Graphics (screen, slide, paper presentations)
- Font Company (602.998.9711):
extensive PostScript font library
- Frame Technology (408.433.3311):
FrameMaker (desktop publishing environment)
- HSD (415.964.1400):
Scan-X (scanners, optical character recognition software)
- Innovated Data Design (415.680.6818):
Dreams (drawing and drafting)
- Lighthouse Design (800.366.2279):
Diagram (graphical diagramming)
- Media Logic (213.453.7744)
Artisan (paint and image processing system)
TopDraw (drawing package)
- Quark (303.934.2211):
QuarkXPress (publishing software)
- RightBrain Software (415.851.1785):
TouchType (typesetting; desktop publishing)
- Stone Design (505.345.4800):
ArtDraw, TextArt (desktop publishing)
- T/Maker (we need a phone and address):
ClickArt (PostScript images library)
- WordPerfect (801.225.5000):
WordPerfect 5.0 (word processor)

CONNECTIVITY
- Abaton (415.683.2870):
InterFaxNX (fax modem)
- Active Ingredients (617.576.2000):
Communicae (VT220 and Tektronix

N.E. 545 Kamiaken St. #8
Pullman, WA 99163
Phone: (509) 334-9594
Email: gerkman@bongo.csc.wsu.edu or gerkman@wsuvm1.bitnet

Other NeXT related User Groups
Mathematica: Special InterestGroup
Email: mathgroup-request@yoda.ncsa.uiuc.edu

Frame: FUN (Frame Users Network)

Email: framers-request@drd.com

NeXT Bulletin Board (comp.sys.next)
Archive to subscribe: email text "SUBSCRIBE NeXT-L" to LISTSERV@BROWNVM.BROWN.EDU
For further information on how to start your NeXT Group, email to the following address
user_groups@next.com

NEW

Chuck Herrik

Product Reviews

PRODUCT: **At The Beep**
AVAILABLE: now
CONTACT: Software Engineering Solutions
PHONE: 512.343.2828
EMAIL: mbrown@math.utexas.edu
(No NeXTmail)

Imagine that you're out of town and you want to pick up the telephone and talk to your NeXT computer... or suppose you want your NeXT to talk to anyone who calls you. **At The Beep** from Software Engineering Solutions in Austin, Texas lets you do all this and more.

At The Beep is a combination of especially-designed hardware and NeXTstep sofware which uses the sound record-and-play-back features of the NeXT to develop a full-featured telephone answering system. This new application has three versions, the Personal System which is shipping now, the Departmental System which should ship in Q1'91 and the Information Retrieval System which is still in the concept stage.

Designed to make operation and maintenance of your telephone answering system as transparent as NeXTstep does the views of your directories, **At The Beep** lets you save messages to files on your disk or include the messages in other applications (like NeXTmail). The Announcements window lets you record and store several different messages, and selection of the active announcement is a click on a button, so to switch from "I'm at lunch" to "I've gone fly-fishing" is effortless.

At The Beep has all the features you've come to expect from telephone answering systems. You can call in and get your messages by remote control and you can interrupt **At The Beep** during a call to speak to a caller. And you can record an outgoing conversation to a disk file. Imagine what Richard Nixon could have done with **At The Beep** if he had had a NeXT in the White House. By the way, the system comes with on-line help built into the application and sports what the developers call an "intelligent icon". When you hide **At The Beep**, the icon updates to let you know how many messages are pending.

How about indulging in some immediate gratification. You can call Software Engineering Solutions and talk to their NeXT at 512.343.2828, and if you leave your name and address you'll get a copy of their **At The Beep** brochure. Or better yet, have your NeXT call my NeXT and let's do lunch. This interpersonal computing is going to be amazing.

Texas

Austin NeXT User Group
Lorne Wilson
950 Capitol of Texas Highway N
Suite 300
Austin, TX 78759
Phone: (512)343-1111
Email: pensoft!lorne@cs.utexas.edu

! DaNG (Dallas Area NeXT Group)
Dirk Hardy, president
Hofbauer Information Systems
5080 Spectrum Drive
Suite 912W (Lock Box 21)
Dallas, TX 75248
Phone: (214) 385-2991
Email: blackbox!kti!dirk@uunet.UU.net

Charlie Lindahl
Automation and Robotics
Research Institute
University of Texas at Arlington
7300 Jack Newell Blvd. S.
Ft. Worth, TX 76118
Phone: (214)284-6122
Email: lindahl@evax.arl.utexas.edu

Meet the last Thursday of the month atBozell Inc. at 201 Carpenter Frwy from 7 to 9PM

hAng (Houston Area Next Group)
John R. Glover, president
E.E. Department
University of Houston
Houston, TX 77204-4793
Phone: (713)749-1820
Email: glover@uh.edu

Meet first Wednesday of the month at the Engineering College, UH, from 6 to 7:30PM

TexNeXT (Texas A&M NeXT User Group)
Keith Perkins, President
Walter Daugherity, Vice President
Knowledge Systems Research Center
Texas A&M University
College Station, TX 77843-3112
Phone: (409)845-1308
Email:daugher@cs.tamu.edu
or keithp@cs.tamu.edu

Utah

SLC NUG (Salt Lake City NeXT User Group)
Gary Mackelprang
150 Nimue
North Salt Lake City, UT 84054
Phone: (810)240-1017
Email: tarbet@chemistry.chem.utah.edu

Washington

Seattle Area NeXT Group
Peggy Thompson, chair
Paget Press
2125 Western Ave., Suite 300
Seattle, Washington 98121
Phone: (206) 448-0845
FAX: (206) 448-2350

University of Washington NeXT User Group
Corey Satten
Network and Distributed Computing
University of Washington
3737 Brooklyn Avenue, NE
Seattle, WA 98105
Phone: (206)543-5611
Email: corey@cac.washington.edu

Meet third Wednesday of month at 3:30PM iin Parrington Hall, room 106 on theUniversity of Washington campus

Washington State University NeXT Group
Joe Gerkman, president

New York

! GUN (Gotham Users of NeXT)
Tim Reed, chair
Objective Technologies, Inc.
7 Dey St.
Suite 711
New York, NY 10007
Phone: (212)227-6767
Email: treed@object.com
Group email: gun@object.com

Robb Allan, co-chair
Helical Systems, Ltd.
595 Madison Avenue
New York, NY 10022
Phone: (212)980-3408
Email: D0808@applelink.apple.com

GUN (continued)
Catherine Pavlov, co-chair
Clearlight International
11-1545 Avenue, #3D
Long Island City, NY 11101
Phone: (718)361-2327
Email: cpav@object.com

North Carolina

SAS NeXT User Group
Pranav Patel
SAS Institute
SAS Circle, Box 8000
Cary, NC 27512-8000
Phone: (919)677-8000
Email: pranav@unx.sas.com
Fax: (919)469-3737

Ohio

! Columbus NeXT User Group
Chuck Dyer
Sr. Systems Analyst
The Ohio State University
1971 Neil Ave-IRCC
Columbus, OH 43210
Phone: (614)292-4843
Email: dyer-c@osu-20.ircc.ohio-state.edu

Oregon

Portland NeXT User Group
John Corey
6105 SW Bonita Rd.
Lake Oswego, Oregon 97035
Phone: (503) 624-8499
Email: baker@next.com

Oregon State University NeXT User Group
Tom Leach, president
Ocean Admin 104
College of Oceanography
Oregon State University
Corvallis, Oregon
Email: leach@satchmo.oce.orst.edu

Meet second Thursday of month at 2:30pm in
Milne Computer Center, Faculty Development
Lab

Pennsylvania

IBUG (Interface Builder User Group)
Dr. Joel Smith
Allegheny College
Meadville, Pennsylvania 16335
Email: IBUG@music.alleg.edu

Rhode Island

NeXT-BUG
Axel Merk
Box 53
Brown University
Providence, RI 02912
Phone: (401)272-2262
Fax: (401)272-2262
Email: agm@cs.brown.edu

Boston, MA 02108
Phone: (617)969-6555
Email: nextwrld!dlavin@uunet.uu.net

Michael Burress, newsletter editor
BCS NeXT Group
One Center Plaza
Boston, MA 02108
Phone: (617)277-5245
Email: bjwalter@bu-pub.bu.edu

Barbara Walter
Email: bjwalter@bu-pub.bu.edu

StrataNUG (Stratus NeXT User Group)
Eric Williams
Stratus Computer
MS M21CAC
55 Fairbanks Blvd.
Marlboro, Ma. 01753-1298
Phone:(508)460-2915
Email: edw@es.stratus.com
FAX: (508)485-8818

Michigan

Michigan State University NeXT User Group
Ray Silverman
Department of Art
Michigan State University
East Lansing, Michigan 48824
Phone: (517)353-9114
Email: bonduku@ibm.cl.msu.edu

Minnesota

Minnesota NeXT User Group
Mike Tie, President/Treasurer
Math/CS Department
Carleton College
One North College Street
Northfield, MN 55057
Phone: (507)663-4067
Email: mtie@carleton.edu

Meet on the second Tuesday of the month

Missouri

St. Louis NeXT User Group
John Bartley
Deloitte & Touche
One City Centre
St. Louis, Missouri 63101
Phone: (314)343-4996
Email: 71511.125@compuserve.com

New Jersey

Princeton NeXT User Group
John James and Declan McCullagh
326 Forbes College
Princeton University
Princeton, NJ 08544
Phone: (609)258-8683
Email: jwjames@pucc.princeton.edu or declan@remus.rutgers.edu

New Mexico

Albuquerque NeXT User Group
Jeff Jortner
Sandia National Laboratories
Division 1424
PO Box 5800
Albuquerque, NM 87185
Phone: (505)846-2613
Email: jnjortn@cs.sandia.gov

Los Alamos NeXT Users Group
Dwight Barrus, chair
Group C-6, MS-B272
Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, NM 87545
Phone: (505)667-8870
Email: dmb@lanl.gov

Joe Kleczka, co-chair
Phone: (505)667-4584 (page number)
Email: jhk@lanl.gov

Box S1117, CSU Campus Activities Center
Ft. Collins, Colorado 80523
Phone: (303)491-6477
Email: dpattie@kontiki.cfnr.colostate.edu

District of Columbia

! WANSIG (Washington Area NeXT SIG)
Hugh O'Neill, president
P.O. Box 39036
Washington, D.C. 20016
Phone: (703)938-NeXT (Joel McClung)
Email: joel@next.com

Meet 2nd Wednesday of month -Bldg 12A
Room B-51 of NIH Complex in Bethesda,
Md. at 7:30PM

Newsletter, monthly
Phillip Fuster, editor
8300 Boone Blvd., Suite 558
Vienna, VA 22182

Naval Research Labs NeXT User Group
Richard Pitre
NRL, Code 5160
Washington, D.C. 20375
Phone: (202)767-3524
Email: pitre@ccf.nrl.navy.mil

Florida

Miami NeXT User Group
Marshall Gilula
2510 Inagua
Miami, Florida 33133
Phone:(305)854-8954
Email: mgilula@miasun.miami.edu

Georgia

! BuzzNUG
(Georgia Tech NeXT User Group)
Erica Liebman
1150 Collier Road NW Apt L-12
Atlanta, GA 30318
Phone: (404)352-5551
Email: erica%kong@gatech.edu

Local Meeting Coordinator :
Bert Lindgren 370-0968 (h)

NeXT Users' Journal, newsletter
Editor-in-Chief, Erica Liebman
(Issues run forty to sixty pages)

Illinois

Argonne NeXT User Group
Mark Henderson
Building 203, Room C-246
Advanced Computing Research Facility
9700 South Cass Avenue
Argonne National Laboratory
Argonne, IL 60439
Phone: (708) 972-5963
Email: henderson@mcs.anl.gov

! ChiNUG (Chicago NeXT User Group)
Bill Parod, president
Northwestern University
627 Dartmouth Place
Evanston, IL 60208
Phone: (708)491-5368
Email: bill_ parod@nwu.edu

Kentucky

Kentucky NeXT User Group
Neil Greene
330 Clifton Circle
Lexington, KY 40508-3405
Phone: (606)258-8655
Email: neil@s.ms.uky.edu

Massachusetts

! Boston Computer Society
(BCS) NeXT SIG
Dan Lavin, president
BCS NeXT Group
One Center Plaza

Berkeley, CA 94709-1496
Phone:(415)549-2684 or (415) 243-9140
Email: bfd@meta-x.stanford.edu

NeXTUCLA
Allen Denison
747 Gayley Ave. #412
Los Angeles, CA 90024
Phone: (213)824-1174
Email: denison@euphemia.math.ucla.edu

Orange County NeXT Group
Jim Brownfield
Suite Software
801 East Katella Ave., Suite 210
Anaheim, CA 92805
Phone: (714)938-8850
Fax: (714)938-8854

Santa Barbara NeXT User Group
Amir Gharaat - president
775 Camino Del Sur Apt. B4
Goleta, CA 93117
Phone: (805) 968-5584
Email: erone%pumpkin@hub.ucsb.edu

Nuggets
Gary Novak, president
Department of Geology
Cal State
5151 State University Drive
Los Angeles, CA. 90032
Phone: (213)343-2400

JPL/Caltech NeXT User Group
Carlos Salinas
Dabney 1-58
Caltech
Pasadena, CA 91126
Phone: (818)356-9258
Email: carlos@eeyore.caltech.edu

UC Riverside NeXT User Group
Paul Lowe, president
University of California
Computing and Communications
Riverside, CA 92521
Phone: (714)787-3883
Email: plowe@ucrac1.ucr.edu

! SCaN (Southern California NeXT Group)
Michael Mahoney
Dept of Computer Science and Engineering
California State University, Long Beach

Long Beach, CA 90840
Phone: (213) 985-1550
Email: mahoney@grafix.cse.csulb.edu

! SNuG (San Diego NeXT User Group)
Nicholas MacConnell, president
1135 Stratford Ct.
Del Mar, CA 92014
Phone: (619)481-7535 or (619)565-9738
Email: tfinn@next.com

Bruce Webster, newsletter editor
3973 Catamarca Drive
San Diego, CA 92124

CP-NUG (Cal Poly NeXT User Group)
Marcus von Engel, president
441 Wodbridge
San Luis Obispo, CA 93401
Phone: (805) 541-6879
Email: mengel@data.acs.calpoly.edu

Colorado

! rmNUG (Rocky Mountain NeXT Users Group)
Dave Hieb, chair
4521 Wellington Rd
Boulder, CO 80301
Phone: (303)530-2560
Email: davehieb@boulder.colorado.edu

Brad Green, co-chair
4600 South Ulster Street, Suite 700
Denver, CO 80237
Email: green_bk@cubldr.colorado.edu

CSU NeXT Computer Users Club
Douglas Pattie

Medical: NeXTMed SIG
Bill Barker and Jim Brinkley
Biological Structure, SM - 20
University of Washington
Seattle, WA 98195
Phone: (206)543-7315
Email: NeXTMed-request@
ulnar.biostr.washington.edu

Music: NeXT Music SIGs (two of them!)
Email:

nextmusic-request@silvertone.Princeton.edu

and
next-music-request@usc.edu

Network and Security Management for Installed Labs and Large Installations
Email: next-lab-request@cs.ubc.ca

Programmers: NeXT Programmers SIG
Email:
next-prog-request@cpac.washington.edu

UNITED STATES

Alaska

Arctic Circle NeXT User Group
Aaron Morse
P.O. Box 900-641
University of Alaska
Fairbanks, AK 99775
Phone: (907)479-2247
Email: fsapm@alaska.bitnet

Arizona

Phoenix NeXT User Group
Gary Frederick, president
20826 N 16th Ave
Phoenix, AZ 85027-3531
Phone: (602)869-0316
Email: jsoft!ggf@uunet.UU.NET

Jim Ames
Arizona State University
CIM Systems Research Center
Tempe, AZ 85287-5106
Phone: (602)965-2906

Tucson NeXT User Group
Robert W. Layhe
CCIT/User Support
University of Arizona
Tucson, AZ 85721
Phone: (602)621-2284
Email: layhe@rcnext1.rc.arizona.edu

California

! BANG (Bay Area NeXT User Group)
BaNG
P.O. Box 8858
Stanford, CA. 94309
Email: BaNG-request@meta-x.stanford.edu

Rick Reynolds
1442A Walnut Street #62
Berkeley, CA 94709-1496
Phone:(415)549-2684 or (415) 243-9140
Email: bfd@meta-x.stanford.edu

Joe Barello
1505 Grand Avenue
Piedmont, CA. 94611
Phone: (415)652-0769
Email: joeba@lll-lcc.llnl.gov

STuN (Stanford Users of NeXT)
Chris Overton
P.O. Box 2628
Stanford, CA. 94309
Email: louiex2@portia.stanford.edu

Berkeley Mac User Group - NeXT SIG
Rick Reynolds
1442A Walnut Street #62

poiker@whistler.
sfu.ca

Tao, monthly newsletter
Robert Lin, editor
Objective Software
1701W 64th Ave
Vancouver, B.C.
Canada V6P 2P3
Phone: (604) 261-0186
Email: rlin@cs.ubc.ca

Ontario

TANG (Toronto Area NeXT Group)
David LaVallee
705 King Street W.
Ontario, Canada M5V 2W8
Phone: (416) 365-2133
Email: David_LaVallee@next.com

Ottawa NeXT User Group
Daniel Cayouette, president
86-632 King Edward Street
Ottawa, Ontario K1N9N2 Canada
Phone: (613)594-3352
Email:
danielc%nx1%dciem@uunet.UU.NET

Quebec

! Montreal NeXT Section of Club Macintosh
Robert Paulhus, president
2250 Guy Street, Room 303
Montreal, Quebec H3H2N2 Canada
Phone: (514)939-0382
Email: paulhus@calvin.cs.mcgill.ca

FRANCE

FaNG - French area NeXT Group
Thierry Charles
32, Rue de Porto - Cite Mion
34000 Montpellier
France
Phone: (33) 67 64 07 97
Email: 73020.2442@compuserve.com

JAPAN

NeXus (Japan NeXT User Society)
Katsuhiro Ohashi
4-4-2
Ebara
Shinagawa-ku
Tokyo, 142 JAPAN
Email: NeXus-office@etl.go.jp
Fax: +81-03-351-0880

NeXus Memo, monthly newsletter

MEXICO

Quetzalcoatl - Mexico NeXT Group
Efrain Rocha
ITESM, Servicios Computacionales
Sucursal de Correos "J"
Monterrey N.L. Mexico 64849
Phone: (83) 58-20-00 ext 4071 or 4073
Fax: (83) 58-89-31
Email: efrain@mtecv2.mty.itesm.mx

NETHERLANDS

Netherlands NeXT User Group
Lex van Sonderen
Email: lex@ica.philips.nl

SWEDEN

SnAG (Svenska NeXT-AnvandarGruppen)
Bjorn Backlund
Box 1263
164 28 Kista, Sweden
Phone: +46 8 752 15 45
Email: snaginfo@sics.se

NeXT SPECIAL INTEREST GROUPS (SIG)

Classroom: NeXT Courseware SIG
Email: send the following text message to
MAILSERV@gac.edu -
"Subscribe next-classroom <your name>"

up or down in a half a year, depending on how much surplus or deficit money we are carrying at that time. Again, no checks will be cashed until we are set up as not-for-profit but do send them in.

Review Materials

If you are a 3rd party developer, please do send in review materials to the address above. We will make every effort to get your product reviewed in the next issue. Upcoming reviews : **Mathematica : Quick Reference**, **Lotus Improv** and **Diagram!**

Software/Hardware Loaners and Donations

The BuzzNUG staff can really use donations and loaners of Hardware and Software materiels. Donations are not tax-deductible, but will be greatly appreciated, treated as Pledges (and you will be sent an adorable stuffed *Buzz*), and we will gratefully thank you at the end of each issue. Please contact me at the above address.

The Evolving Manifesto

As, and when, appropriate, more administrivia will be published in this column. So keep your eyes peeled.

And to every one of you who have helped out, pitched in and made this group effort a real success, all of us here at Tech say THANK YOU!

USER GROUPS

Conrad Geiger

This is the official issue of the NeXT User Groups Directory. The Golden Nugget Award winning user groups have a "!" before their name. These groups are recognized for their outstanding contribution to the NeXT community.

AUSTRALIA

Queensland

OzNeXT (Australian NeXT User Group)
Paul Davis, chair
P.O. Box 65
School of Business
Bond University
Queensland 4229 Australia
Phone: (075)952 291
Fax: 953-3220
Email: paul@terrapin.bus.bu.oz.au

CANADA

Alberta

Calgary NeXT User Group
Alan Dibb, chair
135 Wood Oak Way SW
Calgary, Alberta, Canada T2W3R3
Phone: (403) 220-6320
Email: dibb@cpsc.ucalgary.ca

British Columbia

! Vancouver NeXT Group
Lionel Tolan, chair
Computing Services
Simon Fraser University
Vancouver, B.C.
Canada, V5A 1S6
Phone: (604) 291-4702
Email: lionel_tolan@cc.sfu.ca

NeXTVieW, quarterly newsletter
Tom Poiker and Shirley Chan, editors
Dr. T.K.Poiker
Department of Geography
Simon Fraser University
Burnaby, B.C. V5A 1S6
Phone: (604)291-4515
Email: usertoni@cc.sfu.ca or

NeXT Users' Journal Manifesto

Changes, Goals, Directions

It is the goal of the NeXT Users' Journal to deliver intelligent, well-written articles to an audience of technical readers. We focus on how-to, review and NeXT-information articles. We seek feedback and articles from our readership.

We look for custom objects, application code, disk-tabs, and any other NeXT related code snippets. Hints are eagerly welcomed.

Authors Rewarded

We very much need regular columnists as well as good technical writers. And we are prepared to reward those who pitch in. As soon as we can get our not-for-profit status fixed and can start taking in money, we will be using some of these funds to say "thank you" to our writers, staff and interviewees. For anyone submitting two technical articles, four columns or donating money in excess of $100 to BuzzNUG, we will send you an adorable stuffed *Buzz* in appreciation.

Mailed NUJ

We will now also produced mailed DOS-disk based versions of the NeXT Users' Journal for $36 for six issues (however long that is, anywhere from six months to twelve!). This means that we may have to cut some pictures from your 360 or 720 Kbyte disks to make 'em all fit. However we will include advertising in mailed versions. The monies received for mail subscriptions cover only materials and labor for mailing -- we also ask our mailed subscribers to join in our fundraising efforts through pledges. Send checks to Erica Liebman/1150 Collier Road NW L-12/Atlanta, GA 30318. Checks will not be cashed until we get set up financially as not-for-profit. If someone donates any other disk drive, we will offer those formats too.

Pledges

In order to pay for things like toner, paper, labor, etc, We are asking that our readers : from the sites and through direct mail make generous pledges on a *can-afford-to* basis. Think of this as a kind of Shareware-PBS-Newsletter-Drive. We are asking for pledges of $15/year for students, $30 and way, way, up for people who are actually pulling in salaries. For any pledge, I'll mail you out a *Buzz* related bumpersticker. Over $100 and you get an adorable stuffed *Buzz*. If you are looking for something more in the line of Tech t-shirts, etc then write or call and we'll negotiate. As before, no checks will be cashed until we get the not-for-profit set up but do start sending checks in. All pledgees will be listed at the end of each issue (unless you request otherwise) as a special thankyou.

Advertising

We will start to carry advertising in mailed issues. Due to the size limitations of the DOS distribution media, advertisements must be basic postscript or rich-text to start with. (In other words, no scanned images -- yet!) Advertisors will be listed as sponsors in the Internet version because 1. they are and 2. it is inappropriate to distribute advertising over Internet. Our starting basic rates per page per issue : 1/4 page $150 (sponsor), 1/2 page $ 250 (generous donor), 1 page $400 (honor list). These rates will go

```
arguments\n", PgmName);
        (void)fprintf(stderr, "Usage: %s [UNIT]\n",
PgmName);
        (void)fprintf(stderr,
    "\tUNIT is the OD unit number (e.g., 0a) to be
unmounted and ejected\n", DEVDIR);
        exit(1);
    } else if (argc == 2) {
        /* One argument provided */
        unit = argv[1];
    } else {
            /* No arguments provided; use defaults */
        unit = UNIT;
    }

    cmdbuf = (char *)malloc(MAX(strlen(UMOUNT),
strlen(EJECT) +1) + strlen(DEVDIR)   +

        strlen(DEVICE) + strlen(unit) + 3);
            /* +3 in above for ' ' and '/' before unit
name, and for '\0' to terminate
             * and the +1 is for the 'r' in /dev/rod0a
             */
    (void)sprintf(cmdbuf, "%s %s/%s%s", UMOUNT, DEVDIR,
DEVICE, unit);
    if (0 != system(cmdbuf)) {
        (void)fprintf(stderr, "%s: problems unmounting
%s.\n", PgmName, unit);
        exit(1);
    }

    (void)sprintf(cmdbuf, "%s %s/r%s%s", EJECT, DEVDIR,
DEVICE, unit);
    if (0 != system(cmdbuf)) {
        (void)fprintf(stderr, "%s: problems ejecting
%s.\n", PgmName, unit);
        exit(1);
    }

    exit(0);
}
```

/* mountod.c - mount an od in argv[1], change owner to user */

```
#include <stdio.h>
#include <text/text.h>

#define UNIT   "0a"
                   /* Default unit number */
#define DEVICE "od"
                   /* Default (only available) device
name */
#define DEVDIR "/dev"
                   /* Default (only available) device
directory */
#define MOUNT  "/usr/etc/mount"
        /* Command for mounting filesystem */

#define MAX(x, y)  \
    (x > y ? x : y)


char *getenv();

main(int argc, char *argv[])
{
    char        *cmdbuf;
    char        *PgmName;
    char        *unit;
    char        *user;


    PgmName = basename(argv[0]);
        /* So messages look nice */

    if ((argc > 3) || (argc < 2)){
        (void)fprintf(stderr, "%s: wrong number of
arguments\n", PgmName);
        (void)fprintf(stderr, "Usage: %s directory
[UNIT]\n", PgmName);
        (void)fprintf(stderr,
    "\tUNIT is the OD unit number (e.g., 0a) to be
mounted on directory\n", DEVDIR);
        exit(1);
    } else if (argc == 3) {
        /* Two argument provided */
        unit = argv[2];
    } else {
            /* One argument provided; use default unit */
        unit = UNIT;
    }

    cmdbuf = (char *)malloc(1024);

    (void)sprintf(cmdbuf, "%s %s/%s%s %s", MOUNT,
DEVDIR, DEVICE, unit, argv[1]);
    if (0 != system(cmdbuf)) {
        (void)fprintf(stderr, "%s: problems mounting %s
on %s.\n", PgmName, unit, argv[1]);
        exit(1);
    }

    if (user = getenv ("USER")) {
        (void)sprintf (cmdbuf, "/usr/etc/chown %s %s",
user, argv[1]);
        if (0 != system(cmdbuf)) {
            (void)fprintf(stderr, "%s: problems with
chown\n", PgmName);
            exit (2);
        } /* end if command didn't work */
    } else {
        fprintf (stderr, "%s: environment variable USER
not defined\n", PgmName);
        exit (3);
    } /* end if USER in environment var */

    exit(0);
}
```

/* unmountbackupod - unmount and eject an od */

```
#include <stdio.h>
#include <text/text.h>

#define UNIT   "0a"
                   /* Default unit number */
#define DEVICE "od"
                   /* Default (only available) device
name */
#define DEVDIR "/dev"
                   /* Default (only available) device
directory */
#define UMOUNT "/usr/etc/umount"
        /* Command for unmounting filesystem */
#define EJECT  "/usr/etc/disk -e"
        /* Command for ejecting disk */

#define MAX(x, y)  \
    (x > y ? x : y)

main(int argc, char *argv[])
{
    char    *unit,
            *cmdbuf,
            *PgmName;

    PgmName = basename(argv[0]);
        /* So messages look nice */

    if (argc > 2) {
            /* Arguments provided? */
        (void)fprintf(stderr, "%s: wrong number of
```

the setuid bit set to work properly. The makefile takes care of this for you.

How-2 restore files

You can grep through the .lst files to find out which tar file a certain data file was backed up into ("grep *filename* /tmp/backup/*user*/*.lst"). You can then restore that file from any of the tar files you want. That way, you can go back to any previous version that was stored. Remember that tar demands the complete file name, including the directory names, when restoring a file. Wild cards won't do, nor will file names without the directory specifier. If in doubt about what is acceptable, look at the .lst file and use the entire last string on the line.

Restoring every file in a certain directory is a little harder, but still doable. Type "tar xvf /tmp/backup/*user*/*tarfile.*tar 'tar tf /tmp/backup/*user/tarfile*.tar | grep /*directory*/`". (Note: the character in front of the second tar is the backward single quote. The last character you type is the same thing.) What this does is to make one pass through the tar file extracting the full names of all the files. It then uses grep to pick out only those filenames that contain the directory name you want. It then passes those names back to the other tar, which looks at the tar file again to extract the desired files. This takes a while, as it makes two complete passes through the tar file, but it is a simple way to do what is needed.

You can also restore a complete copy of your entire directory. To do this, start with an empty directory. Untar the full backup ("tar -xf /tmp/backup/*user*/fl.*month.day.hour*.tar"). Then untar each of the incremental backups ("tar -xf /tmp/backup/*user*/sv.*month.day.hour*.tar"), in order from oldest to newest. This way, each time a file is found in a tar file, it overwrites the older one that has already been restored. It is important to remember to untar the incremental files in the order from oldest to newest, instead of backwards. That would leave you with the oldest copy of the file, not the newest.

Things you can configure

What if you don't like ".backup.od" for the name of the timestamp file left behind. Just change the line that starts "timefile=..." to set timefile to some other name.

If you don't want the list file generated (it takes a while to do this), find the two places where there is a line saying "comment out the next 3 lines to avoid writing the list file." Then, comment out the three lines following it in each of the two places.

If you don't want the directory on the backup disk to have the same name as your user login name, edit the line that starts with "savedir=".

If you want to mount the optical disk somewhere other than /tmp/backup, edit the line that starts with "budir=". However, be careful that you don't mount it in any directory that you want to back up. If you do that, tar will fail with the following error "tar: *filename*: file changed size." Basically, it makes sense that you can't backup everything in a directory if one of the things in the same directory is the file that you are backing up to.

What if it didn't work?

If you are running automatically, you will get a mail message every day saying what the backup script did. Otherwise, you will get the output to the Shell window. If it works OK, you will see the following

"tar file written"
"backup.od done"

followed by a line showing how much disk space is left in the backup directory and a list of the files in the backup directory.

Sometimes, the backup procedure doesn't work. Usually, that means that the optical disk filled up during the backup. Or, it can mean that you forgot to leave an optical disk in the drive. Another possible problem is that you don't have the correct directory on the optical disk.

However, if you fill up the optical disk, you will get other messages, "/tmp/backup: write failed, file system is full", followed by "tar: tape write error: No space left on device". This indicates that the backup didn't work. You need to mount an empty optical disk and try again.

If you get a message that says "/tmp/backup not found" or "/tmp/backup/*user* not found", (with your user name) this means that either the optical disk is not mounted on /tmp/backup, or that the optical disk doesn't have a directory with the same name as your user name.

Another problem is that the backup.od script can't deal with optical disks that haven't been initialized. If you get a message saying something like "mount /dev/od0a on directory: No such device or address", then the disk you've tried to insert probably hasn't been initialized. Unmount it and use automount (i.e., just shove it back in again) to initialize it.

certain time, or just when you key in the command; the list file is easily disabled; and when the script runs automatically, it will mail the result of the backup to you.

Restoring files is a little more trouble than backing them up. Since you don't do it as often (you don't, do you?), that doesn't matter much. You can restore only a certain version of a certain file, not necessarily the newest version, but any one. Or, you can restore every file in a specified directory, or every one in your entire directory.

How-2 make backups:

The simplest way to use the script is to just ask it to mount the optical for you, make the backup, then unmount the optical. First, make sure that the script file is properly installed according to the instructions below. Then, open a Shell window, go to your home directory, and type "backup.od -m". What this will do is to put up a panel asking you to insert an optical disk and wait for you to insert the disk. It will then make a full or incremental backup of the files under your home directory to the disk you inserted. Then it will unmount and eject the disk.

The first time you run the script, it will make a full backup, a copy of every file in your home directory or anywhere under that. Every time after that, you will get only an incremental backup, a copy of only those files that have changed.

If you've been making incremental backups and want to make a full backup, then delete the file named ".backup.od" (note that period is the first character in the name.) Then, just make a backup the way you normally do. Instead of making an incremental backup, you will be making a full backup.

If, instead of making a backup of your entire home directory, you want to make a backup of only part of it, use the command "backup.od -m *directory*", where *directory* is the name of the directory you want to back up..

So far, I haven't said anything about automatic backups. Every time you want to make a backup, you have to go to the NeXT machine, type a command, and insert an optical disk. Well, that's the way some people around here do it, but that's not for me. I'm too lazy to remember to make backups that way. So, what I do is to leave the optical disk in the drive all the time. To mount a disk, I just type "mountod /tmp/backup". Then I run the script, telling it not to run now, but to schedule itself to run every day at the same time. Then, I go home. Everything else happens behind my back.

To do this, first mount the disk. Type "mountod /tmp/backup". This tells the system to mount an optical disk on the directory /tmp/backup. Since it is in /tmp, everyone that can log onto my NeXT could write onto it. To keep other people from messing up my backup files with their own, then I create a directory with the same name as my login name (mine is jpm), and change the permissions on that directory so that I'm the only person who can write in it.

Once a disk is mounted, then I open a Shell window, go to my home directory, and type "backup.od -a 10:00pm". This tells the backup script that I want it to run at 10:00 pm every day. When I mount a new disk, I always delete ".backup.od", so that the first backup done to that disk is a full backup. After that, the backups will be incremental. I just let it run every night. When it gets through making the nightly backup, it sends me mail that says that the backup got done OK, and tells me how much space is left on the disk.

I just let it run until, one night, it fills up the disk. When this happens, the backup script will still be futilely trying to write more stuff to the disk when I come in the next morning. I know this because I didn't get my daily reassuring mail message from backup.od. What I do next is to kill the process, eject the disk, mount and erase a new one, then delete ".backup.od". Now, things are set up so that tonight, the script will make a full backup onto the empty disk.

How-2 install the script

To use this script, first you need to put it onto your disk. Find the tar file on one of the archive servers, and transfer it to your computer. Untar the file ("tar xvf backupod.tar"). Then, su to root and type "make install". If you don't have root priviledges on your machine, give the file to your system manager and ask her to do it.

What this does is to compile two small programs, then move the executables of the programs and the script to /usr/local/bin, then move the man pages to /usr/local/man.

One of the two programs is mountod which mounts an optical on a specified directory; the other is unmountod, which unmounts the specified disk and ejects it. These are C programs, and must be owned by root and have

```
        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;
    volatile int foundOne = NO;

 NX_DURING
 [NXApp activateSelf:YES];
 /* here is where you check extensions of
iconPathList
    and determine which handler method to
call
    for EPS, TIFF or raw data */
 NX_HANDLER
 NX_ENDHANDLER
    *flag = YES;
 return 0;
}
@end
```

5.0 The DSP Chip

Experimentation with the DSP chip will quickly reveal that it is not highly optimized for image processing (rather than sound) application because the memory on the chip is so limited, the chip is optimized for integers (not floating point data) and the transportation back and forth of data takes time. However, the chip can be used for some simple applications.The best way to find out about this is to look at the sample applications on your NeXT in /NextDeveloper/Examples/DSP/ArrayProcessing directory.

Some of the best things to use the chip for? Mean/Median filtering and Fourier Transforms.

6.0 Where to Go from here

This paper will give you the bare essentials to start Image Processing programming on the NeXT. However your own experimentation and investigation into the NeXT Application Kit will reveal those objects most appropriate to your needs.I suggest that you proceed in this order :

o loading and displaying raw images.
o saving raw images in TIFF format
o loading and displaying TIFF images
o filtering and displaying loaded images.
o saving filtered images.
o loading images via dragging into your window.

You can use the code segments included in this report, although no warrantee is offered or implied.

Automatic Backups

or

Now that you've got a 256 Megabyte Optical Disk, what do you do with it?

Pat McGee
Computer Graphics Group
Los Alamos Nat. Lab.
jpm@lanl.gov
(LAUR 90-2436)

One thing that I'm pretty paranoid about is making backups. At least, I'm paranoid about it when it's convenient. If it isn't convenient, I often delay making a backup until it's too late. I haven't backed up my Macintosh in over a month. Since I don't have a tape drive, or a way to tell a particular program to run every night, I just don't seem to get around to it.

But, my NeXT gets backed up every night (well, almost every night). On blue (that's its name), I have a neat little shell script that takes care of most of the details for me. All I have to do is to replace the optical disk every time it fills up.

My backup script runs every night at 10:00, long after I'm gone home (at least usually), and copies all the files under my home directory that have changed that day. These files are copied into a tar file on the optical disk. The script also makes a list file of the files that it put into the tar file. This makes it easier to find out where the latest copy of a certain file is. Just grep through all the list files for the desired name, and you'll find out every place that the file was backed up to.

Actually, the backup script is somewhat more versatile that what I normally use it for. You can use it to do full and incremental backups of any directory; you can set the script to automagically make the backups at a

```
return outdata;
}
```

To call this method, you'd write some code similar to :

```
...
filteredPicture = [self mean:[myImage
data]:height:width];
...
```

This example illustrates that no special NeXT programming is really necessary for the intrinsic image processing functions. This, plus the templates given above for data i/o and image display should greatly ease the programming burden of the NeXT-naive (although IP-experienced) programmer.

4.0 Getting Fancy

Every NeXT programmer really should attempt to maintain the NeXT Style guidelines. For the Image Processor, this will probably include two tasks : Open/Save panel invocation and registering windows for dragging pictures into an application. Let's look at each of these.

4.1 Open/Save Panels

The Open/Save panels allow the user to select a file to open or a filename to save to. The results of this operation can be used in the data input and output described earlier.The following example is of the Open Panel. The Save Panel works along similar (if nearly identical) lines.

```
- performOpen:sender
{
    id my_open;

    /* Create and run appkit open panel.
       Exit if user presses cancel */
     my_open = [OpenPanel new];
     if (![my_open runModalForDirectory:"."
             file:NULL]) return self;
     /*... insert here a check for .TIFF
       or .EPS extensions then call
       appropriate
       loading code... */
```

4.2 Registering Windows for Drag

It makes intuitive sense to allow users to "drag" an image into your application or "through" filters such as Sobel, Canny, Thresholding, Median, and so forth. This approach can be a lot less clumsy than using menu options and dialog panels because people are solving their image problems in a more graphical domain.
Here is a basic skeleton for registering any application window with the main server. By the way, you will need to make your custom view a delegate of the Application so that the Speaker/Listener (interapplication communication, which is what we are describing here) will work.

```
#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 "AppropriateHeaderFile.h"

@implementation MyCustomView
/* ... other IP and View specific code ...
*/
-appDidInit:sender
{
unsigned int wn;
    id speaker = [NXApp appSpeaker];
    NXConvertWinNumToGlobal([theWindow
        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)) {
```

window descriptions for the NeXT. (Display PostScript (DPS) is something slightly different -- being a kind of "operating system level" imaging system for the NeXT). You can generally create a postscript file, foo.ps, and safely rename it to foo.eps. The EPS standard is generally used for non-bitmap images. However, you can use bitmaps in EPS.To convert your raw image to postscript follow this boiler plate (this example is for a square image):

%!PS-Adobe-2.0 EPSF-1.2
%%Origin:0 720
%%BoundingBox:0 0 Dimension Dimension (e.g. 256 256)
%%EndComments
/picstr Dimension string def
gsave
dimension dimension scale
dimension dimension 8 (8 refers to 8 bits)
[dimension 0 0 -dimension 0 dimension]
{currentfile picstr readstring pop}
PUT RAW IMAGE HERE
grestore

Here's an example for a 256 x 256 picture

%!PS-Adobe-2.0 EPSF-1.2
%%Origin:0 720
%%BoundingBox:0 0 256 256
%%EndComments
/picstr 256 string def
gsave
256 256 scale
256 256 8
[256 0 0 -256 0 256]
{currentfile picstr readstring pop}
image
grestore

3.0 Basic Image Processing

After the data is sucessfully loaded in from either TIFF or raw image files, it is time for you to start doing your filtering, region growing, etc. Your data will be stored in a continuous block pointed to by your myImage instance variable. You can find out the details of your newly loaded image by querying this object.

Getting information about the image will generally use these methods :

```
- bitsPerPixel
- samplesPerPixel
- bitsPerSample (NXImageRep)
- isPlanar
- numPlanes
- numColors (NXImageRep)
- hasAlpha (NXImageRep)
- bytesPerPlane
- bytesPerRow
- colorSpace
- pixelsWide (NXImageRep)
- pixelsHigh (NXImageRep)
```

Those methods that are notated `NXImageRep` are actually defined in the similarly named parent class of the `NXBitmapImageRep`.

Getting general image information is important, but critical is getting access to the data itself, which is accomplished through the -data method. The pointer to the data block returned by this method can be passed to any image processing routine that takes a block of data, so extensive rewriting of routines for the NeXT is eliminated. This is needless to say, a great convenience.For example, say you have defined a mean filter along the following lines :

```
#define ctoi(x) (char) (x & 0xff)
/* 3 x 3 mean filter : could be easily
modified for arbitrary filter size. PIX is
predefined as type char[height][width]; */
- (PIX)
mean:(PIX)data:(int)height:(int)width
{
int x,y;
int sum;
PIX outdata;

outdata = (PIX) (malloc(height*width));
for (y = 1; y < height-1; y++)
 for(x = 1; x < width-1; x++)
 {
        sum = 0;
        for (j = -1; j < 2; j++)
          for(i = -1; i < 2; i++)
            sum += data[y+j][x+i];
   outdata[y][x] = sum / 9;
 }
```

```
   [myImage drawIn:&bounds];
   return self;
}
```

Notice that the method drawIn: is predefined in the declaration of the NXBitmapImageRep object. This means that no matter how bit your image is to start with, it will automatically adjust to the size of your custom view object! Beware, as this means your aspect ratio may not be retained. You should also note that in version 2.0 of NextStep, the new message has generally be superceded by the init and alloc messages. Typically, to use a new image, you will want to do as follows:

```
{
...
    if (myImage) [myImage free];
    myImage = [[NXBitmapImageRep
           alloc] someInit];
...
}
```

and for deallocation of the view, don't forget that you'll be needing to deallocate any image you have been using,

```
- free
{
    // free all dependant inst. vars
    [myImage free];
    return [super free];
}
```

2.0 File I/O

File Input and Output may occur on either Tagged Image File Format (TIFF) files (the native NeXT Bitmap representation) or on raw data. TIFF files are the most easily read in, along the lines of :

```
- loadImage:(const char *)imageName
{
    if (myImage) [myImage free];
    myImage = [[NXBitmapImageRep alloc]
        initFromSection:imageName];
    [self scale:1.0:1.0];
    [self display];
    return self;
}
```

Raw files need to be handled differently. A large number of parameters define exactly how the data is structured, a stream will have to be opened to read in the data and so forth. This example assumes an 8-bit greyscale stream.

```
- loadImage:(const char *)imageName
    :(int)width:(int)height
{
FILE *f1;
NXStream *myStream;
    f1 = fopen(imageName, "r");
    fread(myImage, sizeof(char),
              width*height, f1);
    fclose(f1);

    if (myImage) [myImage free];
    myImage = [[NXBitmapImageRep alloc]
    initData:myImage /*must be i.v. */
         pixelsWide:width
         pixelsHigh:height
         bitsPerSample:8
         samplesPerPixel:1
         hasAlpha:NO /*no transparency */
         isPlanar:YES /* trust me */
         colorSpace:NX_OneIsWhiteColorSpace
         bytesPerRow:width
         bitsPerPixel:8];
    [self scale:1.0:1.0];
    [self display];
    return self;
}
```

Once the data is loaded (either way), it may be stored in TIFF format as follows :

```
- storeImage:(const char *)fileName
{
NXStream *myStream;
    myStream = NXOpenMemory(NULL, NULL,
        NX_WRITEONLY);
    [myImage writeTIFF:myStream];
    NXSaveToFile(myStream, fileName);
    NXCloseMemory(myStream, NX_FREEBUFFER);
    return self;
}
```

2.1 EPS Blather

Extended PostScript (EPS) is generally considered to be a proper (hah!) superset of PostScript (PS) and is used for creating page and

```
protos in the include files */
extern RETCODE dbrpcinit(DBPROCESS *dbProc, char
*rpcname, DBSMALLINT options);
extern RETCODE dbrpcparam(DBPROCESS *dbProc, char
*paramname, BYTE status,

      int type, DBINT maxlen, DBINT datalen,

      BYTE *value);
extern RETCODE dbrpcsend(DBPROCESS *dbProc);
extern void dbsetlname(LOGINREC *loginrec, char
*name, int field);
extern DBINT dbretstatus(DBPROCESS *dbProc);
```

-- insertaddress.proc --

```
create proc insertaddress
   @name varchar(100),
   @address varchar(100),
   @citystatezip varchar(100),
   @phone varchar(100),
   @email varchar(100)
as
   insert addressbook
   values(@name, @address, @citystatezip, @phone,
@email, NULL)
   if @@error = 0
      update addressbook set comments = NULL
where (name = @name)
   select * from addressbook where (name = @name)
```

-- create_addressbook --

```
create table addressbook
   (name varchar(100) not null,
    address varchar(100) null,
    citystatezip varchar(100) null,
    phone varchar(100) null,
    email varchar(100) null,
    comments text null,
   )

create unique clustered index name_ind
   on addressbook(name)

create index phone_ind
   on addressbook(phone)

create index email_ind
   on addressbook(email)
```

Quite a bit (but not all) about IP Programming on the NeXT

Erica J. Liebman

The NeXT is well suited for Image Processing (IP) programming for a number of reasons. First, it utilizes the unified PostScript imaging system easing display of monochrome or color pictures. Second, NeXT supplies a number of development tools such as the Interface Builder program to speed the programming task. Third, a Megapixel (780 x 1280 picture elements) display and an on-board Motorola 56000 Digitial Signal Processing chip (DSP) comes standard with each system.

1.0 The View Object

Display of any graphical elements on the NeXT will involve the View object or one of its descendents in the inheritance heirarchy. This object is responsible for drawing to the screen, for updating itself when obscured by other windows and so on.

1.1 Bitmap Image

IP Applications typically use Bitmaps since their data are primarily two-dimensional picture samples taken from cameras, digitizers, scanners or similar devices (such as radar, etc).The NeXT Application Kit (appkit) provides an object called NXBitmapImageRep which handles both Tagged Image File Format pictures and raw data streams. Bitmaps may be of any depth. Typically monochrome pictures are 8 bit greyscales. The NXBitmapImageRep can handle color pictures, pictures of arbitrary depth and pictures with an alpha (transparency) channel.
Bitmap Images are easily integrated into custom Views.

1.1.1 Using Bitmaps in Views

Bitmaps are usually drawn in a View's drawSelf:: method. For an image, say, stored in the myImage instance variable,

```
drawSelf:(NXRect *)
rects:(int)rectCount
{
   NXEraseRect(&bounds);
          /*clear the background */
```

```
message handler */
- _setSQLMessageLevel:(int)severity
line:(int)line msg:(int)msgno
             proc:(char *)procname server:(char
*)srvname
             state:(int)msgstate text:(char
*)msgtext
{
   [sqlLevelText setIntValue:severity];
   [sqlLineText setIntValue:line];
   [sqlMessageText setIntValue:msgno];
   [sqlProcText setStringValue:procname];
   [sqlServerText setStringValue:srvname];
   [sqlStateText setIntValue:msgstate];
   [sqlTextText setStringValue:msgtext];
   return self;
}

/* the func to handle messages from the server.
calls a method to do it.
 * this functions interface is defined by the DB-
Library.  It is called
 * whenever the sybase server has something
important to tell us.
 */
int msg_handler(dbProc, msgno, msgstate,
severity, msgtext, srvname,
             procname, line)
   DBPROCESS *dbProc;
   DBINT msgno;
   int msgstate, severity;
   char *msgtext, *srvname, *procname;
   DBUSMALLINT line;
{
   [NXApp _setSQLMessageLevel:severity line:(int)
line

   msg:(int) msgno proc:procname server:srvname

   state:msgstate text:msgtext];
   return(0);
}

/* Outlets */
- setSqlTextText:anObject
{
    sqlTextText = anObject;
    return self;
}

- setSqlLevelText:anObject
{
    sqlLevelText = anObject;
    return self;
}

- setSqlStateText:anObject
{
    sqlStateText = anObject;
    return self;
}

- setSqlServerText:anObject
{
    sqlServerText = anObject;
    return self;
}

- setSqlProcText:anObject
{
    sqlProcText = anObject;
    return self;
}

- setSqlLineText:anObject
{
    sqlLineText = anObject;
    return self;
}

- setSqlMessageText:anObject
{
    sqlMessageText = anObject;
    return self;
}

@end
```

/* dbfix.h OR sybfix.h*/

```
/*
 * Part of DirectoryAssistance.
 *
 * Routines that are not declared in the include
files and a message handler
 * external declaration.
 */

extern int msg_handler(DBPROCESS *dbProc, DBINT
msgno, int msgstate,

   int severity, char *msgtext, char *srvname,

   char *procname, int line);

   /* line is DBUSMALLINT but it may be necessary
to

    * declare it int for weird reasons

    */
/* db routines that for some reason have no
```

```
- setResultsScrollView:anObject
/* The scrollview in the nib file is a regular
text scrollview, but we want a
 * scrolling matrix of buttons, so we have to
monkey around with it.
 */
{
       NXRect frm;
       id myProto;
       NXSize asize;
       int i, rows, cols;

    resultsScrollView = anObject;

    [[resultsScrollView docView] getBounds:&frm];
    /* first make a prototype button */
    myProto = [ButtonCell newTextCell:""];
    [myProto setBordered:NO];
    [myProto setBezeled:NO];
    [myProto setAlignment:NX_LEFTALIGNED];
    [myProto setEditable:NO];
    [myProto setSelectable:NO];
    [myProto setHighlightsBy:NX_CHANGEGRAY];
    [myProto setShowsStateBy:NX_CHANGEGRAY];
    /* now make a matrix of those buttons */
    resultsMatrix = [Matrix newFrame:&frm
mode:NX_RADIOMODE prototype:myProto

       numRows:0 numCols:1];
    [resultsMatrix setFont:[Font newFont:"Times-
Roman" size:14]];
       asize.width = frm.size.width;
       asize.height = 16.0;
       [resultsMatrix setCellSize:&asize];
       [resultsMatrix allowEmptySel:YES];
       [resultsMatrix setAutoscroll:YES];
       [resultsMatrix
setBackgroundGray:NX_LTGRAY];
       [resultsMatrix
setCellBackgroundGray:NX_LTGRAY];
       /* set the double action and target of the
matrix */
       [resultsMatrix setTarget:self];
       [resultsMatrix
setDoubleAction:@selector(displayAction:)];

       /* make the matrix the docView of the
scrollview and free the old Text
        * docview.
        */
       [[resultsScrollView
setDocView:resultsMatrix] free];
       /* give it an arrow cursor instead of I-
Beam */
       [resultsScrollView setDocCursor:NXArrow];
    return self;
}

@end
```

/* SQLApp.h */

```
/* This class is necessary only for implementing
a message handler for sybase
 * All it has over Application is outlets for the
various text fields,
 * the message handler function, and a method
which is called by the message
 * handler function which actually does the work.
 */

#import <appkit/Application.h>
#import <appkit/TextField.h>
#import <stdlib.h>
#import <strings.h>
#import <sybfront.h>
#import <sybdb.h>
#import "sybfix.h"
   /* declare my message handler and some left
out db stuff*/

@interface SQLApp:Application
{
    id sqlTextText;
    id sqlLevelText;
    id sqlStateText;
    id sqlServerText;
    id sqlProcText;
    id sqlLineText;
    id sqlMessageText;
}

/* Outlets */
- setSqlTextText:anObject;
- setSqlLevelText:anObject;
- setSqlStateText:anObject;
- setSqlServerText:anObject;
- setSqlProcText:anObject;
- setSqlLineText:anObject;
- setSqlMessageText:anObject;

@end
```

/* SQLApp.m */

```
#import "SQLApp.h"

@implementation SQLApp

/* a private method which does the work of the
```

```
- deleteAction:sender
/* Action of the delete button.  Deletes the name
in the name field of the
 * address window if there is only one record
matching it, and the user
 * confirms it.
 */
{
	int retval, i;

	/* clear the search results window on each
new find */
	for (i = [resultsMatrix cellCount] - 1; i
>= 0; i--)  {
	    [resultsMatrix removeRowAt:0
andFree:YES];
	}
	[resultsScrollView display];

	/* first find the name to see if it
uniquely identifies one record */
	retval = [self findName:[theForms
stringValueAt:0]];
	if (retval == 0)  {
	    /* no records match */
	    [helpField setStringValue:"No matching
record exists.  "
		"Cannot delete."];
	    return nil;
	}  else  if (retval == 1)  {
	    /* one matches so ask if we should
delete and do it. */
	    retval = NXRunAlertPanel("Delete",
"Delete the record which is now "
		"shown in the address window?",
"OK", "Cancel", NULL);
	    if (retval == NX_ALERTALTERNATE)  {
		[helpField setStringValue:"Delete
canceled."];
		return nil;
	    }  else  {
		/* do it */
		if ([self deleteName:[theForms
stringValueAt:0]] == nil)  {
		    [helpField setStringValue:"Delete
unexpectedly failed."];
		    return nil;
		}  else  {
		    [helpField setStringValue:"Record
deleted.  "

		"Press add to undo."];
		    return self;
		}
	    }
	}  else  {
	    /* more than one matching record was
found, so don't do it */
	    [helpField setStringValue:"More than one
record matches that name.  "
			"Only one record at a time can be
axed."];
	    return nil;
	}


    return self;
}

- displayAction:sender
/* Action for Display button and double click on
scrolling matrix of names.
 * finds and displays the address info for the
name which is selected in
 * the results window
 */
{
	if ([self findName:[[resultsMatrix
selectedCell] title]] != 1)  {
	    [helpField setStringValue:"Unexpected
error.  Did not find name."];
	    return nil;
	}  else  {
	    [helpField setStringValue:"Found
record."];
	    return self;
	}
}

/* outlets */
- setTheForms:anObject
{
    theForms = anObject;
    return self;
}

- setCommentScrollView:anObject
{
    commentScrollView = anObject;
    return self;
}

- setHelpField:anObject
{
    helpField = anObject;
    [helpField setStringValue:"Welcome to
AddressBook"];
    return self;
}
```

```
        moreRows = dbnextrow(dbProc);
    }
    return 2;
}

- deleteName:(const char *)name
/* delete the given name from the database */
{
    dbcmd(dbProc, "delete addressbook where
name = '");
    dbcmd(dbProc, name);  dbcmd(dbProc, "'");
    dbsqlexec(dbProc);
    if (dbresults(dbProc) == FAIL)  {return
nil;}
    return self;
}

- addResult:(char *)name
/* add an entry to the results window's scrolling
matrix of buttons */
{
    int rows, cols;

    [resultsMatrix addRow];
    [resultsMatrix getNumRows:&rows
numCols:&cols];
    [resultsMatrix setTitle:name at:rows-1 :0];
    [resultsMatrix sizeToCells];
    [resultsScrollView display];
    return self;
}

- addAction:sender
/* Action of the add button.  gets the info from
the forms, and calls
 * insertname:... to do it.
 */
{
    const char *name, *ad, *csz, *ph, *em;
    char *comments;
    int length;

    name = [theForms stringValueAt:0];
    ad = [theForms stringValueAt:1];
    csz = [theForms stringValueAt:2];
    ph = [theForms stringValueAt:3];
    em = [theForms stringValueAt:4];
    length = [[commentScrollView docView]
textLength];
    comments = (char *)malloc(length+1);
    [[commentScrollView docView]
getSubstring:comments start:0 length:length];

    if (![self insertName:name address:ad
citystatezip:csz
            phone:ph email:em
comments:comments])  {
        [helpField setStringValue:"Add failed.
The problem could be that the"
                " name you entered already
exists."];
        free(comments);
        return nil;
    }

    [helpField setStringValue:"Record added."];
    free(comments);
  return self;
}

- findAction:sender
/* Action of find button.  Finds the name in the
name field of the address
 * window.
 */
{
    int retval, i;

    /* clear the search results window on each
new find */
    for (i = [resultsMatrix cellCount] - 1; i
>= 0; i--)  {
        [resultsMatrix removeRowAt:i
andFree:YES];
    }
    [resultsScrollView display];

    /* fill the helpfield with helpful text */
    retval = [self findName:[theForms
stringValueAt:0]];
    if (retval == 0)  {
        [helpField setStringValue:"Find failed.
No records were found for "
            "that name."];
        return nil;
    } else if (retval == 1)  {
        [helpField setStringValue:"Found one
record."];
        return self;
    } else {
        [helpField setStringValue:"Found more
than one matching record.  The "
            "first is displayed above.  The
names of the others are listed "
            "in the Search Results window."];
        return self;
    }
  return self;
}
```

```
	strlen(csz), (BYTE *)csz);
	   dbrpcparam(dbProc, "@phone", (BYTE)0,
SYBVARCHAR, -1,

	strlen(ph), (BYTE *)ph);
	   dbrpcparam(dbProc, "@email", (BYTE)0,
SYBVARCHAR, -1,

	strlen(em), (BYTE *)em);
	dbrpcsend(dbProc);
	dbsqlok(dbProc);
	dbresults(dbProc);
	dbnextrow(dbProc);
	/* see if it worked */
	if (dbretstatus(dbProc) != 0)  {
	   NXRunAlertPanel("Database Message", "The
attempt to add the record "

	   "failed.", "OK", NULL, NULL);
	   return nil;
	}
	/* now save the comment text */
	length = strlen(comments);
	dbwritetext(dbProc, "addressbook.comments",
dbtxptr(dbProc, 6), DBTXPLEN,
		dbtxtimestamp(dbProc, 6), YES,
length, comments);

	return self;
}

- (int)findName:(const char *)name
/* finds and returns any records which contain
name.  This is case insensitive
 * even though SQL is case sensitive because of
the call to makeFindString.
 * In the case of multiple records being
returned, the first is displayed in
 * address window, and the names of all records
are listed in the results
 * window, from which they can be individually
found.  Returns 0 if no
 * records are found, 1 if a single record is
found, and 2 if more than one
 * record is found.
 */
{
	DBCHAR nm[101], ad[101], csz[101], ph[101],
em[101];
	char firstname[101];
	char *comments;
	int length;
	STATUS moreRows;
	char *findName;

	/* first get a case insensitive version of
name */
	findName = [self makeFindString:name];
	/* find the records */
	dbcmd(dbProc, "select * from addressbook
where name like '%");
	dbcmd(dbProc, findName); dbcmd(dbProc,
"%'");
	dbsqlexec(dbProc);
	free(findName);
	if (dbresults(dbProc) == NO_MORE_RESULTS)
{return 0;}
	/* bind the results to our variables */
	dbbind(dbProc, 1, STRINGBIND, (DBINT)0,
nm);
	dbbind(dbProc, 2, STRINGBIND, (DBINT)0,
ad);
	dbbind(dbProc, 3, STRINGBIND, (DBINT)0,
csz);
	dbbind(dbProc, 4, STRINGBIND, (DBINT)0,
ph);
	dbbind(dbProc, 5, STRINGBIND, (DBINT)0,
em);
	moreRows=dbnextrow(dbProc);
	if (moreRows == NO_MORE_ROWS)  {return 0;}

	/* handle first row */
	[theForms setStringValue:nm at:0];
	strcpy(firstname, nm);
	[theForms setStringValue:ad at:1];
	[theForms setStringValue:csz at:2];
	[theForms setStringValue:ph at:3];
	[theForms setStringValue:em at:4];
	/* get the comments.  Since it is a text
column it should be read
	 * directly, not bound.
	 */
	length = dbdatlen(dbProc, 6);
	comments = (char *)malloc(length + 1);
	strncpy(comments, (char *)dbdata(dbProc,
6), length);
	comments[length] = '\0';
	[[commentScrollView docView]
setText:comments];
	free(comments);

	/* handle other rows if any */
	moreRows = dbnextrow(dbProc);
	if (moreRows == NO_MORE_ROWS)  {return 1;}
	/* add the first name to the results window
*/
	[self addResult:firstname];
	/* add the rest */
	while (moreRows != NO_MORE_ROWS)  {
	   [self addResult:nm];
```

```
	dbexit();
	[super free];
}

- lowerString:(char *)string;
/* alter contents of string to change all upper
case letters to lower */
{
	char *stringPtr;
	stringPtr = string;
	while (*stringPtr != '\0')  {
	   if (*stringPtr >= 'A' && *stringPtr <=
'Z')  {
		*stringPtr = *stringPtr + 32;
	   }
	   stringPtr++;
	}
}

- (char *)makeFindString:(const char *)origStr
/* return a pointer to a string which is a
suitable non-case-sensitive string
 * for the origStr in sybase's syntax (ie change
each alpha char like 'a' to
 * "[aA]")  Freeing the string is the caller's
responsibility.
 */
{
	char *findStr;
	char *lowerOrig;
	char *indexOrig, *indexFind;
	int findStrLen=0;

	/* make an all lower case copy of the
original string */
	lowerOrig = (char
*)malloc(strlen(origStr));
	strcpy(lowerOrig, origStr);
	[self lowerString:lowerOrig];

	/* find out how long findStr will be */
	indexOrig = lowerOrig;
	while (*indexOrig != '\0')  {
	   if (*indexOrig >= 'a' && *indexOrig <=
'z')  {
		findStrLen+=4;
	   }  else  {
		findStrLen++;
	   }
	   indexOrig++;
	}

	/* now malloc some memory and then go
through and construct */
	findStr = (char *)malloc(findStrLen+1);
	indexOrig = lowerOrig;
	indexFind = findStr;
	while (*indexOrig != '\0')  {
	   if (*indexOrig >= 'a' && *indexOrig <=
'z')  {
		*indexFind = '['; indexFind++;
		*indexFind = *indexOrig;
indexFind++;
		*indexFind = ((*indexOrig) - 32);
indexFind++;
		*indexFind = ']'; indexFind++;
	   }  else  {
		*indexFind = *indexOrig;
indexFind++;
	   }
	   indexOrig++;
	}
	/* cap the string */
	*indexFind = '\0';

	free(lowerOrig);

	return findStr;
}

- insertName:(const char *)name address:(const
char *)ad
		citystatezip:(const char *)csz
phone:(const char *)ph
		email:(const char *)em
comments:(const char *) comments
/* Calls an SQL remote procedure with the
arguments given to add the
 * information to a new record.  In order to
allow unlimited length comments,
 * the comment field is added separately through
a call to dbwritetext.
 */
{
	int length;

	/* add the easy stuff with the rpc */
	dbrpcinit(dbProc, "insertaddress",
(DBSMALLINT)0);
	   dbrpcparam(dbProc, "@name", (BYTE)0,
SYBVARCHAR, -1,

	strlen(name), (BYTE *)name);
	   dbrpcparam(dbProc, "@address", (BYTE)0,
SYBVARCHAR, -1,

	strlen(ad), (BYTE *)ad);
	   dbrpcparam(dbProc, "@citystatezip",
(BYTE)0, SYBVARCHAR, -1,
```

```
#import <sybfront.h>
#import <sybdb.h>
#import "sybfix.h"

/* declare my message handler and some left out
db stuff*/

@interface SQLInterface:Object
{
    id theForms;
    id commentScrollView;
    id helpField;
    id resultsScrollView;
    id resultsMatrix;

       /* dbProc Variables */
       DBPROCESS *dbProc;
       LOGINREC  *login;
}

+ new;
- free;

/* SQL DB-Library interfaces */
- insertName:(const char *)name address:(const
char *)ad
             citystatezip:(const char *)csz
phone:(const char *)ph
             email:(const char *)em
comments:(const char *) comments;

- (int)findName:(const char *)name;
- deleteName:(const char *)name;

/* Interface Utilities */
- addResult:(char *)name;

/* Action methods */
- addAction:sender;
- findAction:sender;
- deleteAction:sender;
- displayAction:sender;

/* Outlets */
- setTheForms:anObject;
- setCommentScrollView:anObject;
- setHelpField:anObject;
- setResultsScrollView:anObject;

@end
```

/* SQLInterface.m */

```
#import "SQLInterface.h"

@implementation SQLInterface

+ new
/* this method logs in to the server as user "sa"
and assumes no password.
 * it assumes the addressbook table resides in
the master database, for
 * simplicity, although this practice is not
advised.
 */
{
       self = [super new];

       /* init DB-Library */
       if (dbinit() == FAIL)  {
          NXRunAlertPanel("Database Problem",
              "Make sure the sybase dataserver is
running.",
              "OK", NULL, NULL);
          exit(ERREXIT);
       }
       /* login to the server */
       login = dblogin();
       DBSETLUSER(login, "sa");
       DBSETLPWD(login, "");
       DBSETLAPP(login, "AddressBook");
       dbProc = dbopen(login, NULL);
       if (dbProc == NULL)  {
          NXRunAlertPanel("DB Error",

       "Can't log in to database server.  Is the
server running?"

       "  Ask system administrator for help.  It
could also be "

       "an incorrect user name or password.",

       "Exit", NULL, NULL);
          exit(ERREXIT);
       }
       /* set up the message handler */
       dbmsghandle(msg_handler);
       /* uncomment the line below this one and
put in the right database name
        * if the table you're accessing is on
another database besides the master
        */
       /* dbuse(dbProc, "DatabaseName"); */

       return self;
}

- free
/* quit from the server */
{
```

they are listed in the matrix of buttons in the results window.

The `-deleteName:` method merely sends a batch to SQL telling it to delete the given name. It is the simplest of the DB interface methods.

Next comes a small utility method to add a name to the results window which is fairly self explanatory.

Now come the action methods. There are four buttons in this application and each has an action. The add button's action is `-addAction:`. It collects the strings from the various forms in the address window, and sends them to `-insertName:` ... The `helpField` is filled with text that indicates the outcome of the operation.

The `-findAction`: method gets a name from the name form and calls `-findName:`. It also fills `helpField` with text indicating the outcome.

`-deleteAction:` first finds the name in the name form, and if there is one and only one match, asks for confirmation and deletes the corresponding record. If more than one match is found, it's action is almost identical to `-findAction:`, and nothing is deleted.

`-displayAction:` finds the name which is selected in the results window. It is both the action of the display button and the double action of the matrix of buttons in the scroll view in the results window. Names that are listed in that matrix should be in the database because they are only put there in the event of a find with more than one match.

All that is left are the outlet setting methods, and the only non-standard one is `-setResultsScrollView:` because it needs to set up the matrix (and get rid of the default Text object which interface builder sticks in it). This is standard Appkit code, and I will leave it to the reader to figure it out. (It's not hard.)

Other Files and Miscellaneous Matters

There are two other files of interest in this application both necessary for smooth functioning and compiling.

The first is `sybfix.h`, an include file which declares the prototype for the message handler function as well as some definitions for DB-Library functions that for some reason are left out of the standard Sybase include files.

The other file is `Makefile.preamble` which adds a couple of lines necessary for the compiler to locate the Sybase include and library files.

I know this article has not explained everything you need to know about Sybase and the DB-Library, but it is meant as an explained example, not a description of the DB-Library package. The documentation for Sybase and the DB-Library are very good, and I suggest you buy a set or print out a set (from the on-line version).

I have heard from people running NeXT System Release 2.0 (in a beta version) that Sybase won't run. I have also heard that this is being fixed, or is fixed, or something. I hope so, because Sybase is a very useful thing to have around. It allows one to build sophisticated database applications in a small amount of time. Just like the Appkit allows one to build sophisticated interfaces in a short time.

This program took me 4 hours to get running, as I mentioned before, which I see as a great proof of the claims of NeXT. I doubt that a comparable terminal version of this program could have been developed in much less time. The Appkit and DB-Library are easy to integrate, and while I'd love to see a more sophisticated interface to SQL, like a kit of objects, what I have now is pretty good.

I hope that lots of people will build database applications for the NeXT, both specific like this one, and general (a good database definition tool, and several other things come to mind). Have fun.

/* SQLInterface.h */

```
/* This class contains the meat of the
application.  It logs onto the server,
 * has methods for finding, adding, and deleting
records, and contains the
 * action methods for all the buttons.
 */

#import <objc/Object.h>
#import <appkit/Form.h>
#import <appkit/ScrollView.h>
#import <appkit/Text.h>
#import <appkit/Matrix.h>
#import <appkit/ButtonCell.h>
#import <appkit/Cursor.h>
#import <appkit/Panel.h>
#import <stdlib.h>
#import <strings.h>
```

so I included one in this example.

The remote procedure insertaddress takes data for all the columns except the comments column adds a new record with that info, and selects it. It selects it so we can then insert the comment data with a call to a special DB-Library routine `dbwritetext.`

Once the table and single remote procedure have been defined, you are ready to start the actual C programming.

The SQLApp Class

The first class in our application is SQLApp. The only reason we need this class is that `NXApp` is a global variable which can be referenced from outside the context of the appkit. The DB-Library has a feature which allows you to specify a function which is called with certain information every time SQL has something important to say. This has to be a regular C function adhering to a specific prototype. But we want to display that information in a window in text fields. We can only do this by writing the function so that it calls a method of one of our objects which knows about the window and fields through the standard outlet mechanism. The only one of our objects readily accessible is `NXApp`, so we put a method there which the message handler function can call which will display the information.

The SQLApp class, then, contains this method, the definition of the actual message handler function (whose status has been demoted to glue), and outlets for the text fields we want to fill.

The SQLInterface Class

The main workhorse of our application is the SQLInterface class. It is responsible for maintaining the connection to the server, and for responding to action messages from all the buttons in our windows. We will go through it method by method.

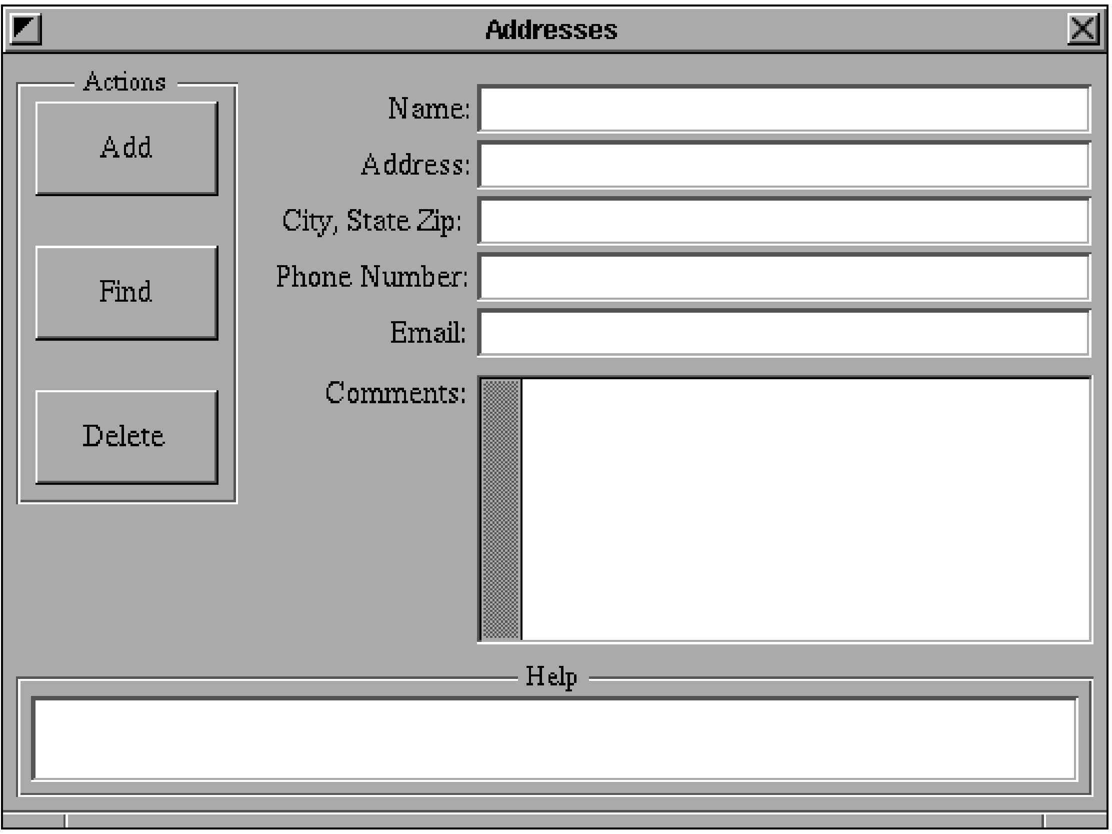
The `+new` method is overridden to add the functionality of logging in to the SQL server. Again, an empty password is assumed for the "sa" account.. The code is fairly straight forward for logging in to the server, and not much needs explanation here. We set up a login structure and open the connection, and unless this fails, we continue and set the message handler, and then we're done.

Next is `-free`. This only exits from the server and calls its super.

Two string utility functions follow. The first accepts a string and converts it to lowercase. The second is a little more involved. It takes a string and returns a new string which is a case insensitive version of the original. This means that the letter 'a' in the original would be converted to the string '[aA]' in the new string. SQL is case sensitive, but the address book shouldn't be. These utilities make it possible.

Next come the three procedures which actually call DB-Library routines (besides new and free). First is `-insertName: address: citystatezip: phone: email: comments:`. This procedure calls the remote procedure insertaddress on the SQL server. Then it must add the comments information with a call to `dbwritetext.` There are three calls which are used to set up a remote procedure call `rpcinit`, `rpcparam`, and `rpcsend`. With these you specify the procedure and its parameters. Then it must wait for the server to come back (`sqlok`) and process the results (`dbresults` and `dbnextrow`). `dbretstaus` is checked to see if the procedure succeeded. Then we stick in the comment text through a call to `dbwritetext` which is a procedure used for putting data into text or image columns.

`-findName:` is the next method. It searches the database for names containing the name it is given (so you can search for all the mike's, and not just Mike Ferris). To do this we search for the string surrounded by '%'. The '%' character works as a wildcard like '*' in Unix. The first thing that this method does is make a case insensitive string by calling `makeFindString`. Then it sends an SQL command batch to the server and gets the results. The DB-Library has a routine for binding the results of a select statement to local variables. This is called `dbbind`, and it should be called after `dbresults`, but before `dbnextrow`. Each call to `dbnextrow` will change the values of these bound variables, so the method can cycle through all the results easily. Once it binds the variables, it calls `dbnextrow` and gets the first row of data. The information is put into the address window. As usual text columns have to be dealt with separately. We read the comments column separately (first getting its length and `malloc`ing memory for it, then getting the data). The comment is put into the scrolling view. Once the first record is processed, there may be more, and if so



The SQL End

I assume in this article that Sybase is up and running on your cube (or slab) and that you have access to the "sa" account. In fact, the program assumes that you have the password for the "sa" account set to "" (an empty string).

The first step in writing a database application is setting up the tables in Sybase. The files `create_addressbook` and `insertaddress.proc` are batch files for SQL. They can be run through `isql`. In this example I decided to go the easy route and define the database table directly on the master database in Sybase. This is not advisable for sizable projects.

The table has six columns or fields. Name, address, city state zip, phone, email address, and comments. These are all `varchar` type columns except comments which is `text` type. The difference is that with `varchar` you have to give an upper limit <= 255 chars for each field whereas `text` type is of unlimited length. The comment doesn't really have to be that long, but text fields are useful and they need to be handled somewhat differently through the DB_Library,

Sybase, the DB-Library, and the Appkit

by Mike Ferris
mike@opus.portal.com

I've been working a lot lately with Sybase. I must say I'm pleased at the ease with which it is possible to access Sybase servers through the DB-Library, and at how simple it is to integrate the DB-Library into an Appkit program. My purpose in writing this is to provide a simple, yet complete sample of how this integration can be accomplished. When I set out to do this article, I knew I would need an example program, so I decided to write it first. I thought about what the example should be, and came to the conclusion that it should be a real application with some utility, and that it should demonstrate all the major DB-Library calls which I find most frequently used.

Address Book

Every computer needs an electronic rolodex which looks neat, works well, and never gets used, so that is what I wrote. It took me a little over four hours to write this application, and it seems to work fine. Any further necessary debugging is left as an exercise.Here is what the main window looks like:

The address is entered and displayed in the forms and scroll view. If you enter information and press the add button, the information is added to the database as a new record. Duplicate names are not allowed. If you enter just a name or part of a name and press find, addresses containing that name are found. If more than one is found, the first is displayed, and a list of all the names found is given in a separate window. Names in this list can be double clicked to find them separately. If you type a name and press delete, the record matching that name is deleted (after confirmation). On deletes, the name is first found, and if there are more than one match, the delete fails.

Helpful text is displayed in the help field. Messages from the server are caught and displayed in a separate window. These are normally error messages. Records can be modified by simply deleting them, making the changes and re-adding them. A modify button is left as an exercise. Finding on criteria other than the name is also left to the reader.

NeXT demonstrated its commitment to Mathematica in many ways, notably, through Richard Crandall's new book, Mathematica for the Sciences, and by Richard's scheduled lecture, entitled "Great Problems of History," attended by virtually all conferees. This lecture covered the period 50 A.D. to the present, and how Mathematica can be used to study the historic changes in science.

lX. PARALLEL DISTRIBUTED COMPUTING

Before the holidays Richard Crandall and the "NeXT community supercomputer" discovered a factor of a huge mystery number called F15. Mathematicians around the country verified that indeed the factor was legitimate, but it turned out it had been found recently elsewhere. No cigar -- but the system works!

Zilla was allowed to roam free over the holiday break. On January 5, 1991, Zilla caught another factor, this time of the mystery number called F13, which has 2500 digits. The factor is: 2663848877152141313. A previous factor was found in 1974.

Mathematicians around the country were queried once again. In particular, Prof. Wagstaff of Purdue University, who keeps a central archive on factorization results, has informed us that this discovery is new and genuine.

The New York Times recently took note of randall's work with Zilla.

Wednesday, January 30, 1991

Creating Computer Godzillas

"Researchers at the computer maker NeXT, Inc. have developed a potent program that harnesses the unused computing power in a network of computers.

"Affectionately named Godzilla or just Zilla, the program permits users to break up the thorniest scientific and other computation problems into small parts and then runs them in parallel on unoccupied machines. Godzilla will take up residence on any computer in a networked NeXT work stations whose owner gives permission. It waits until the computer is not being used and then runs part of the problem. If the user returns and taps his or her keyboard, Godzilla has the manners to stop what it's doing and wait for the computer to be free again. Godzilla has already been used for a variety of mathematical explorations, like factoring very large numbers. Richard E. Crandall, a physicist who directs NeXT's scientific computation group, said many other uses were possible. Projects as diverse as quantum theory calculations, designing new medicines, or creating high-resolution computer animations are possible.

"Taken by itself, each NeXT computer has the power of a scientific work station. Yet when harnessed together by Zilla a group of NeXTs can match the speed of a supercomputer on some problems. To define the scale of Godzilla, the NeXT researchers like to speak of one Zilla unit, or a Z.U., which is equivalent to the power of100 NeXT workstations. The performance of one Z.U. is roughly equal to the speed of a Cray Y/MP computer, one of the world's fastest. Zilla is not confined to NeXT computers. A network of NeXTs running Zilla can be connected with a genuine supercomputer, creating a very fast computer."

X. MISCELLANEOUS

NeXT Campus Consultants

NeXT's Campus Consultant program continues to support campuses nationwide. The idea for the program began 18 months ago as a way to provide additional campus technical support. It has blossomed into a full-blown program; As independent contractors, Campus Consultants may do any or all of the following: assist in academic project development; help support and manage the installed base of NeXT computers; support the campus bookstore and computing center; participate in technology fairs and conferences; support the local NeXT user group.

NeXT is committed to educating and training these students. Recently, NeXT held developer camp for Campus Consultants where they learned to program on NeXT computers. Via programs such as campus consultant training, NeXT is committed to providing academic developers with excellent expert support and advice.

year. Here's what they said about NeXT:

"Steve Jobs has finally delivered on the original promise of the NeXT computer. These machines are fast and very inexpensive for what they do. Also, the applications coming out for them are, as promised, very hot. . .

"These new systems point the way for advanced use of color display PostScript. NeXT has taken a leading role in the migration of expensive, powerful workstation technology to cheaper, desktop platforms...."

V. QUESTIONS ABOUT NeXT

The NeXT keyboard

As you may know, NeXT changed its keyboard when we began shipping the NeXTstation and the '040-based NeXTcube. As NeXT has grown and expanded distribution channels, it's become important that we be able to ship our systems to Europe and other parts of the world. In Europe, there are much more stringent ISO standards for keyboard layouts. Our original keyboard did not meet those standards.

One of the key requirements was that the return key had to be "L" shaped. To accomplish this, one of the keys had to be removed and the key codes remapped. In lengthy discussions with our engineers and our European and Asian teams, the new key layout was approved.

During this transition period, you may recommend to the UNIX folk that they remap their keyboard. In the developer directory, there is an application called Keyboard which allows anyone to remap the keyboard.

Vl. SUPPORT FROM NeXTedge

NeXT has brought all of its customer service and support programs together into one, new organization -- NeXTedge. NeXTedge includes hardware service, software support, training, leveraged support partnerships, and more.

NeXTedge's Customer Education group offers four training courses ranging from basic orientation to advanced network management. These courses have been redesigned to include Release 2.0 and have been expanded to allow for more comprehensive discussion and increased hands-on lab time. All classes are held regularly at NeXT locations around the country. Call 1-800-848-6398 for more course or schedule information or to register for any NeXT education course.

Courses currently offered:

1. Introduction to Interpersonal Computing
A two-day course focusing on the fundamentals of using and maintaining a NeXT computer, with emphasis on effective personal and interpersonal computing skills.
2. Improv 1.0
A two-day course designed for end users with previous experience on a NeXT machine. The course concentrates on the fundamentals of using Lotus Improv on the NeXT computer to analyze data and create powerful graphical presentations using Lotus' charting tool, Presentation Builder.
3. NeXT System Administration
This five-day course teaches one how to use, administer, and network NeXT computers.
4. NeXT Hardware Service
This two-and-a-half day course focuses on general usage, maintenance, and hardware service.

Vll. NeXT on Campus

The winter issue of NeXT on Campus will be released in the second week of February. Look for the spring issue in late April. We are always interested in learning more about innovative research and teaching applications of NeXT technology. Please contact David Spitzler, managing editor of NeXT on Campus, if you you'd like to discuss a project on your campus. David can be reached at David_Spitzler@NeXT.com

VIII. 1991 Mathematica Conference

Wolfram Research and Addison-Wesley hosted the second annual Mathematica conference in San Francisco, from January 12-15, 1991.

In addition to monochrome and color NeXTstations, the NeXT booth demonstrated custom front-ends to Mathematica created by faculty and commercial developers. Among the front-ends demoed were Gourmet, a supercalculator, and Objective Technologies' MathPalette, which allows any programmer to create a Mathematica application from within Interface Builder.

lll. NeXT HARD DISK DRIVES

340 MB Hard Disk Drive Replaced by 400MB HD

NeXT has replaced a 340MB Hard Disk Drive with a 400MB disk drive -- at no added cost to NeXT customers. NeXT is one of the first computer systems vendors to be shipping a 3.5-inch drive of this capacity.

As stated, there will be no price increase for this larger capacity drive and all customers who ordered 340MB drives will be shipped this larger capacity drive at no extra cost. The disk is preloaded with the extended version of Release 2.0 software and with the extended version loaded, has more than 180MB of free space.

Order and pricing information: Existing orders for 340MB systems are being converted to 400MB drives.

Summary of hard disk product availability:

Drive Availability
105 MB Hard Drive : now
200 MB Hard Drive : now
400 MB Hard Drive : now
660 MB Hard Drive (NeXTcube Only) : now
1GB Hard Drive (NeXTcube Only) : late Q1' 91

lV. NeXT IN THE NEWS

1. **Wall Street Journal**, February 1, 1991, p. B1

NeXT Scores Big With Campuses On Second Try

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

Mr. Jobs, who helped to pioneer personal computing when he ran Apple,started NeXT with the idea of supplying college educators and students with an advanced computer capable of mixing huge amounts of audio, video and text. Mr. Jobs. who oversaw the creation of Apple's novel Macintosh computer, seems to be showing signs of his old spark. NeXT's newest models, which began rolling off the assembly line late last year, contain state-of-the-art hardware and sophisticated software. They are selling to students and faculty for as little as $3,000, less than half the price of the original machine and much less than comparable computers sold by rivals.

Mr. Jobs said NeXT is posting monthly sales equivalent to an annual rate of more than $100 million. Vicki Brown, an analyst at International Data Corp., thinks NeXT is on target to sell from 36,000 to 44,000 units this year. University customers are the most eager buyers. Mr. Jobs has deep ties with academics from his days running Apple, the longtime leader in the education market. Some, who have stuck with Mr. Jobs through his recent ups and downs, say he is making good on his vow to set a new standard for academic computing. "I think he has met many of his original goals, '' said Brian Hawkins, vice president for computer services at Brown University. He said purchases of NeXT machines by Brown staff and students "have picked up considerably" in recent months. "NeXT finally has the machine it promised; that's fantastic," added Edward Barboni, vice president of information at Allegheny College. He thinks NeXT has staying power. "We've had opportunities to peek at NeXT's future products," he said, adding, "What's coming down the, pipe is more good stuff."

2. **INFOWORLD**, January 7

Paul Saffo's predictions for 1991

"Unix will break out of niche markets - and NeXT will make it into the big time. Wrapped in interface software that shelters users from its user-vicious core, Unix will finally make it onto the desktops of nonengineering knowledge workers. The company that will work this minor miracle will not be Sun, but NeXT, which will emerge as a major computer player."

3. **BYTE Magazine**, January 1991

Last month, Byte Magazine awarded the NeXTstation and NeXTdimension its "Award of Distinction." Award recipients are chosen by Byte editors based on the following criteria: the product or technology is innovative, has had significant impact on its respective market niche, advances the state of the art, or provides a superior price-to-performance ratio. The three award categories are Awards of Excellence, Distinction and Merit. Byte editors voted 67 products worthy of awards this

job! I'm looking forward to the final release. . . . "

Charlie S. Lindahl, Automation and Robotics Research Institute, University of Texas at Arlington

Internet E-mail: lindahl@evax.arl.utexas.edu

NeXTConnection

A mail-order service for software and peripherals has been established - NeXTConnection, the same folks who have made PC Connection and MacConnection so successful. Products began shipping in November. The existence of NeXTConnection will not in any way preclude individual campuses from forming distribution and pricing arrangements directly with software or peripherals vendors.

Vendors currently selling through NeXTConnection: Abaton, Absoft, Adobe, American Power, Avery, Digital Instrumentation Technology, Dove, Frame Technology, Hayes, Intel, Kensington Microware, Lighthouse Design, Maxell, Moustrak, Practical Peripherals, Pacific Microelectronics, PLI, Safe Power, Sony, Stone Design, T/Maker, Toshiba, Tripp Manufacturing, WordPerfect.. Vendors soon to be available through NeXTConnection: Canon, Coda, Dataviz, Informix, Insignia Solutions, Lotus, Software Ventures.

FORTRAN from Abasoft

Absoft's Object-Oriented FORTRAN for Release 2.0 is now shipping and is available from: Absoft Corporation/2781 Bond Street/Rochester Hills, MI 48309/313 853-0050/313 853-0108 fax/Attn: Sales or Bill Clephane

FrameMaker Update

A new, interim release of FrameMaker 2.0 is now available, FrameMaker 2.0d. This new version, which replaces existing versions, fixes a number of small bugs and ensures that FrameMaker 2.0 works well with NeXTstep 2.0

To obtain the new release, FrameMaker 2.0d, Higher Education customers should obtain a copy from a Campus Support Center. Customers who are unable to contact a campus support center should mail an optical disk to: FrameMaker 2.0d Upgrade/NeXT Telebusiness/900 Chesapeake Drive/Redwood City, 94063

Please include your return address to insure that your FrameMaker 2.0d is routed correctly. Customers with floppy disk drives can mail should request the floppy version and will receive FrameMaker 2.0d on floppies.

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

MediaStation 1.5

MediaStation 1.5 is a revised version of Imagine's multimedia database, storage and retrieval application. MediaStation 1.5 takes full advantage of NeXT software release 2.0 , including support for JPEG image compression. Imagine is shipping this product to existing customers as an upgrade free of charge.

II. NeXT COLOR PRODUCTS

NeXTstation Color computers will be shipping soon. Buyers of these computers will have a choice of two MegaPixel Color Displays: a Philips 17-inch and Hitachi 21-inch. Both monitors display the same number of pixels. The 17-inch Phillips has a list price of $2,995; the Hitachi has a list price at $3,995. Your actual campus price will depend on the discount level available to your campus, based on your campus contract with NeXT. Both monitors will be available immediately when NeXTstation Color products ship.

The **17-inch Color Display** is manufactured by Philips. It is black-cased and displays 1120 x 832 bits. This monitor has features, performance, and overall quality which compares favorably to the Sony monitor that originally anticipated to ship with NeXT color workstations. This product supports both NeXTstation Color and NeXTdimension systems/Pricing: $2,995 (list price)/Resolution: 1120 x 832 at 92 dpi

The **21-inch color display** offers the same number of pixels (1120 x 832) but at 75 dpi. Small fonts are very readable! We're offering this monitor at a very aggressive $3,995 list price. This compares with Radius, offering this new monitor for the Macintosh for $4500. This monitor, too, supports both NeXTstation Color and NeXTdimension systems/Pricing: $3,995 (list price)/Resolution: 1120 x 832 at 75 dpi

Because of the enthusiastic response to the original Improv offer, Lotus has extended the offer of a free copy of Improv with the purchase of every NeXT system through March 31, 1991. As stated above, customers must send their NeXT system registration cards to NeXT to receive this free copy of Improv.

Lotus Improv API

Improv 1.0 includes, as a yet undocumented and 'unofficial' feature, an Application Program Interface (API) allowing programmers to message Improv tools from within their application. For developers interested in doing Objective C programming, the API provides access to a series of internal Improv entry points in either of two ways: through Speaker-Listener from another application, or through user-written code dynamically linked into Improv. Application developers can build sophisticated, automated applications by using the API in concert with NeXT's Interface Builder, other objects (such as external databases) and C programs. Improv's API will be available in April. Please wait to hear more details from NeXT; do not contact Lotus directly.

Mathematica

Mathematica 2.0 is in the process of being completed by Wolfram Research, Inc. and will ship sometime late winter or early spring. NeXT education users may use the existing version of Mathematica (1.2) on NeXT System Release 2.0. Your NeXT campus resale location will receive the 1.2 version soon and users will be able to make a copy of the software by bringing either one 2.88 MB floppy diskette or two 1.44 MB diskettes to your reseller.

Questions about Mathematica 1.2

Q: I want to install Mathematica version 1.2 on my Software Release 2.0. What is the best way to proceed?

A: It's a good idea to keep Mathematica and its associated files in the same places in which they are found on the 1.0 Software Release. For sake of example, the following assumes you are copying Mathematica from a Release 1.0 optical disk. You should allot a little more than 4 MB for Mathematica.app and about 2 MB for the notebooks and packages found in NextLibrary.

```
localhost> su
Password:
# cp -rp /1.0Disk/NextApps/Mathematica.app /NextApps
# cp -rp /1.0Disk/NextLibrary/Mathematica /NextLibrary
```

If you want to run Mathematica either from a shell or remotely, you'll need to move these files from the Mathematica file package to /usr/bin:

```
localhost> su
Password:
# cp -rp /1.0Disk/NextApps/Mathematica.app/Kernel/math /usr/bin
# cp -rp /1.0Disk/NextApps/Mathematica.app/Kernel/mathremote
        /usr/bin
```

X for NeXT

Announced at Uniforum in Dallas last month, Pencom Software released Co-Xist, an X Window System for NeXT computers. Co-Xist provides full X Window System Version 11 Release 4 (X11R4) client and server support within the NeXTstep environment. For more information, contact:Pencom Software/9050 Capital of Texas Highway North, Suite 300/Austin, TX 78759/512 343-1111/512 343-9650 fax/uunet!pencom!pensoft!co-Xist

Here is what one NeXT user group had to say about Co-Xist:

"Last night Pencom came to our user group (DaNG -- Dallas Area NeXT group, based at University of Texas, Arlington) from the USENIX/UNIFORUM conference and demoed and spoke to the group."

Some impressions:

"It definitely WORKS! They had all of the standard X-clients running: XEYES, XTERM, etc.

"Clients did NOT have to be modified to work under their system.

"Distribution currently takes 20-30MB (this may change in the actual release; please double-check with the company for specifics). This includes the X man pages & X fonts, which take up the bulk of the space.

"One of the most impressive things: the developer was asked, 'How long was this in development?' The answer was, 'I started in October and we had a version running by the end of November.'"

"CONGRATS to Pencom for doing a bang-up

illustration from T/Maker Company to customize and enhance presentations. Improv spreadsheets and graphs are linked dynamically so that any changes made to a worksheet are automatically reflected in the associated graph.

Improv imports and exports Lotus 1-2-3 spreadsheets and reads any other spreadsheet that writes .WK1, .WK3, or ASCII files, enabling users to make use of existing 1-2-3 data, exchange data, and combine efforts with users working on other platforms.

System Requirements, Pricing And Promotion Extension

Lotus Improv ships on 3.5 inch diskettes and runs on all NeXT computers with the NeXTstep 2.0 system software. Improv supports the NeXT 400 dots per inch PostScript Laser Printer and the Apple LaserWriter.

Lotus Improv has a suggested retail price of $695. However, Lotus and NeXT have extended a worldwide promotion through March 31, 1991, offering a free copy of Lotus Improv to customers who buy and register any of NeXT's new 68040 computers or who upgrade their current NeXT computer with the 68040 upgrade kit. Customers may request their free copy of Lotus Improv when they register with NeXT by calling 1-800-848-NeXT. Customers will receive the product directly from NeXT. After the worldwide promotion expires, Improv will be available through resellers and VARs that carry NeXT hardware and other NeXT products. Further details on these plans will be available at the end of March. For additional product information, customers in the U.S. can call 1-800-343-5414.

Lotus Development Corp., founded in 1982, develops, markets and supports business software and CD-ROM databases that help users access, analyze, communicate and share information. The company's first product, Lotus 1-2-3, is the most popular software program in the world with more than 14 million users. Lotus markets its broad range of products in more than 65 countries and offers users comprehensive support options, including 24-hour-a-day, 7-day-a-week support in the U.S. for PC versions of its 1-2-3 spreadsheet.

(All prices are for the U.S.A. only.)

Note:

Purchasers of new 68040 NeXT products between our September, 1990 introduction and March 31, 1991 (including '040 upgrade boards) will receive a FREE copy of Improv. In order to receive their software, all users MUST have registered their new computer or upgrade. If you have purchased a system and have any questions about the registration process, please call NeXT at 1-800-848-NEXT.

Also, **WordPerfect**'s new version, WordPerfect for NeXT, began shipping recently (last week, I believe). This program is file compatible with all WP versions (DOS or Mac)--but it is their first WYSIWYG implementation. Beta users loved it! It is available from **NeXT Connection** (1-800-800-NEXT), along with most of the third party softward and hardware peripherals.

Finally, the new NeXT specific magazine has hit the news stands, **NeXTworld**. It is pretty hot...you might want to find one and check it out. Plans are to publish it every other month initially -- Steve

Even More News

News from NeXT

From: Ron Weissman, Director, Higher Education, NeXT Computer, Inc.

Date: February 6, 1991

1. **THIRD-PARTY SOFTWARE/HARDWARE UPDATE**

WordPerfect

WordPerfect is shipping. For more information about pricing and availability, contact WordPerfect directly : WordPerfect Corporation/1555 N. Technology Way/Orem Utah 84057/(801) 225-5000

Lotus Improv

Lotus Improv began shipping on February 4. If you purchased an 040-based NeXT computer or upgrade board, or know someone who has, please make sure you or your colleague register that computer with NeXT. You must complete and return your NeXT registration card in order to get Improv. (There is no special Lotus card to send in.)

this server is a Sony News Workstation. (files which are sent by ftp should be tar-ed and compressed).

I hope you will enjoy the capabilities of this server, and all suggestions are welcome. Phil

PS: add a CONTENT file describe the content of the dir where it is in. forgive my english ! :)

More News

c/o Steve Siezcko

FOR IMMEDIATE RELEASE

Lotus Announces Shipment Of Lotus Improv: Early Users Respond Enthusiastically To Spreadsheet's Innovation

CAMBRIDGE, Mass., February 4, 1991 --- Lotus Development Corp. today announced it has begun shipping Lotus Improv, its new spreadsheet for the NeXT computer, on schedule. Improv introduces a new approach to spreadsheets combining traditional spreadsheet metaphors, such as rows and columns, with new, breakthrough features, such as dynamic spreadsheet views, plain English formulas, and presentation graphics incorporating data, text, graphics, images and sound.

"Reaction to Lotus Improv has been enthusiastic," said Don Casey, vice president, Lotus Spreadsheet Division. "Early users have found that Improv enables them to view and analyze information faster and more dynamically than ever before, and agree Improv represents an exciting and significant step forward in spreadsheet technology."

Improv has been instrumental in driving customer interest in NeXT's new 68040-based computers. According to Todd Rulon-Miller, vice president of sales for NeXT, "Improv's unique approach to analyzing and viewing data offers our customers powerful, new capabilities not available on any other platform. It provides the level of innovation our customers need and expect from Lotus and NeXT."

Improv has been well received among spreadsheet users in corporations and higher education who rely on spreadsheets for financial planning, merger and acquisition analysis, market analysis, sales forecasting, data management, course administration, and grants management.

According to Eric Spahr, vice president at Shearson Lehman Bros., "In my opinion, Lotus has truly broken new ground with Improv. I need a spreadsheet that allows me to build models for forecasting and accounting and to restructure my data in any way I want. Improv enables me to do this quickly and easily, and the result is a spreadsheet my traders and customers can understand."

New Spreadsheet Capabilities For Improved Data Manipulation & Analysis

Improv enables users for the first time to quickly reformat the spreadsheet by creating multiple views of the same spreadsheet without re-keying information. With Improv's dynamic views feature, data can be rearranged to compare and gain greater insight into otherwise hidden data relationships; expanded to show more detail; hidden or grouped for clarity; and formatted for presentation needs. Any changes made to data in one view are automatically reflected in all related views.

Improv also allows users to name data cells and formulas using English words instead of letters and numbers. With English formulas and cell names, the structure and logic of the spreadsheet are immediately understood, while data accuracy and the ability to spot errors are enhanced significantly. Formulas are displayed separately in a window below the spreadsheet for easy viewing and modification and increased spreadsheet maintainability. Improv formulas can be applied to many cells at once reducing a spreadsheet's complexity. If there are errors in the spreadsheet formula, Improv immediately identifies them and, where possible, suggests solutions.

Lotus Improv helps users create advanced presentations combining data, text, graphics, images and sound for voice annotations. For building presentations, Improv offers full-featured drawing and charting capabilities. Charting options include 3D bar, stack, line, bar, pie, scatter, and area graphs. Users can rotate 3D bar graphs to show the bars from any perspective, and easily change font types, background colors and fill-in patterns of their graphs. Users also can add TIFF or EPS images such as those provided in a sampler of ClickArt

```
    int     i = 0;
       //  for keeping track of rows in a column
    DIR   *myDir;   //  for reading a directory; see
sys/dir.h
    struct direct
    *myEnt;   //  for reading a directory; see
sys/dir.h

    [browserBox setTitle:"Browser's Unlimited"
ofColumn:0];

    myDir = opendir(NXHomeDirectory());
    // I want to read the user's Home Directory.

    while ( (myEnt = readdir(myDir)) != NULL)
    {
      if (strcmp(myEnt->d_name, ".") == 0 ||
strcmp(myEnt->d_name, "..") == 0)
       // Get rid of the dot dirs.
      continue;
      else
      { // Put some data into the NXBrowser.
        // matrix is given to us when this method is
messaged by NXBrowser.
        [matrix insertRowAt:i];
        [ [matrix cellAt:i :0] setStringValue:myEnt-
>d_name];
        [ [matrix cellAt:i :0] setLeaf:YES];
        [ [matrix cellAt:i :0] setLoaded:YES];
        i++;
      }
    }
    closedir(myDir);

    return i;    // How many items were processed;
required by NXBrowser.
}

- userSelected
{
// This method is called by the NXBrowser when an
action (click, doubleclick) happens
// on an item in the NXBrowser.  This method is
connected to the NXBrowser object
// by setAction: in appDidInit:.

// Get the string in the cell that was selected in
the NXBrowser and put that string into
// the textBox.

    [textBox setStringValue:[ [ [browserBox
matrixInColumn:[browserBox selectedColumn] ]
selectedCell] stringValue] ];

    return self;
}

@end
```

Call For Discussion

Brad Cox offers the concept of *Software IC's*. Steve Jobs promised us software tools that we could buy off the shelf. Why hasn't the market been flooded with object-libraries? Why hasn't NeXT's object library been flooded with submissions. Why haven't the sites filled up with custom-objects?

Got an opinion? Willing to venture a guess? Take this opportunity to become a pundit. Send mail to *erica@kong.gatech.edu* before 28 February 1991 to participate in the discussion. Responses (edited as necessary) will appear in the Feb-March issue.

With any luck, we'll be carrying reviews of two such object libraries. And remember : Your feedback is, as always, greatly appreciated.

Notre Philippe

Philippe Provoste

hi NeXTusers ! My ftp server is up ! :)

It will be primarily used to collect classes (both in C++ or ObjC), but it will accept sources files. No binary-only are accepted since this server is aimed to provide a good support to people (beginner/developpers) who want to program NeXTStep.

Because, the idea of collecting classes is a bit new, i need your help to feed the server, *your* server. the address is: 192.33.156.81the account is ftp, your login name is the passwd. It is running 24/24 h all the year.

Because using class is not so straightfoward, you must include what your class is used for, and other additionnal info into an INFO or README file posted with the class.Having a GNU licence or not isn t a problem.

You can submit your work either by ftp in the pub/submission directory or by mail (to me) to: phil@\[192.33.156.81\]

Because this is the first server in france dedicated to the NeXT, and because *alot* of programmers begins to work on the NeXT, you will find in the near future, few French products...:)

But i need you re help...to succeed in this action of having a dedicated server.so please, feed it feed it FEED IT !!!

All files must not be in a NeXT mail format since

here is to show off the NXBrowser class. After building the interface, it's a good time to save the interface file and create a project.
Next, create a subclass of object called MyBrowser. This class is the heart of the application. In the class Inspector of IB, add two outlets: textBox and browserBox. These are how we will communicate with the TextField and NXBrowser interface objects. Move back to the class inspector and unparse MyBrowser. Click OK to create the [hm] files, and to add to them to the project.
Time now to connect the MyBrowser class to the UI objects. With the MyBrowser class highlighted in the Class browser, Instantiate to create MyBrowserInstance. MyBrowserInstance will show up in the File Window.
The first connection is between File's Owner and MyBrowserInstance. Control-drag a connection from File's Owner to MyBrowserInstance. The Connections Inspector shows that **delegate** is the only outlet for File's Owner. We want MyBrowserInstance to be a delegate of File's Owner. Delegation allows surrogates to do work for others. In this specific case, we want MyBrowser to do the appDidInit: method for File's Owner. This delegation allows us to do some special things in the MyBrowser appDidInit: method. This should become more clear when you review the accompanying code.
Next, we want to connect MyBrowserInstance to the UI objects created earlier. Control-drag from MyBrowserInstance to the Text Field object in the application's main window. Again, the Connections Inspector shows the outlets and actions; MyBrowser has only two outlets. Choose the textBox outlet, and click connect in the Inspector. Control-drag from MyBrowserInstance to the NXBrowser in the main window, choose browserBox outlet, and click connect in the Inspector. Save your work. This completes the IB portion of the project.

MyBrowser Source Code

The source code for MyBrowser.[hm] is presented below. You will notice some differences from the typical 1.0 code. For example, the unparse command in IB no longer creates the "set" methods--a welcome relief from source code clutter. Have Fun!

```
/* MyBrowser.h */
#import <objc/Object.h>
@interface MyBrowser:Object
{
    id  browserBox;
    id  textBox;
}

- appDidInit:sender;
- (int) browser:sender fillMatrix:matrix
inColumn:(int)column;
- userSelected;
@end

/* MyBrowser.m */

#import "MyBrowser.h"
#import <appkit/appkit.h>
#import <stdio.h>
#import <sys/dir.h>

@implementation MyBrowser
-appDidInit:sender
{
// MyBrowser must be a delegate of File's Owner for
the appDidInit: to catch.
// Create this delegate relationship in IB with a
connection from File's Owner to this
// class.

// Delegation is necessary for the app to fill the
Matrix in the NXBrowser.
// Targeting connects the NXBrowser with the app so
the NXBrowser knows where to send
// certain messages.
// Setting the Action tells the NXBrowser which of
the Target's methods to message when
// it receives an action (click or doubleClick).

    [browserBox setDelegate:self];
    [browserBox setTarget:self];
    [browserBox setAction:@selector(userSelected)];

// These are what actually cause the NXBrowser to
load and display data.
// The NXBrowser will message the delegate to load
the data.
    [browserBox load];
    [browserBox displayColumn:0];

    return self;
}

- (int) browser:sender fillMatrix:matrix
inColumn:(int)column
{
// This is called by the NXBrowser via delegation;
stuff the NXBrowser with appropriate
// data here.
// This method is defined by the NXBrowser class;
don't change the method declaration.
```

Each column consists of a ScrollView or a ClipView containing a Matrix filled with NXBrowserCells. The NXBrowser's delegate is queried to put data into columns as the user navigates through the hierarchy.
The delegate manages data displayed in the NXBrowser's columns. (NXBrowser Class Description, Preliminary Release 2.0 Copyright ©1990 NeXT Computer, Inc.)

How the NXBrowser Class Works

The NXBrowser class works somewhat differently than other Application Kit classes. The NXBrowser class uses other classes via delegation and targeting to both display data and to take action on data that is selected in the browser. These classes can be the same, or different. These necessary relationships are described more fully below.
To put data into a browser a *delegate* of NXBrowser is required. The NXBrowser class defines a number of methods that the NXBrowser will message for putting data into a browser, *but these methods are implemented by the NXBrowser's delegate*. In short, the NXBrowser provides the method names, and your code is where these methods actually do the work. Hopefully, the accompanying example code will clear up some of this mystery.
In addition, the NXBrowser requires a *target* that will do something with the data that is selected in the browser. This is similar to delegation (as defined above), but more flexible in that the programmer decides what method is called to handle the data selected in the browser.
NeXT says it like this:
The delegate must know how to retrieve data and place it in the NXBrowser. The target must know what to do with that data once the user has selected it. (NXBrowser Class Description, Preliminary Release 2.0 Copyright ©1990 NeXT Computer, Inc.)

A Simple Example

Since the browser class in 2.0 is a tool that we can use in our work here, I decided to take a closer look at it, thinking I might get a head start on a few 2.0-based applications. As I got into this project, I began to see that the NXBrowser object behaves somewhat differently than other objects I was used to working with. Once I created a working example, I saw the light, so to speak. Hence, I've decided to share my insights with the NeXT programming community.
The example below is a simple demonstration of one possible browser incarnation. The application reads the user's home directory, and displays it in a NXBrowser. If the user selects an item in the browser, by way of either a click or a double-click, the selected item is displayed in the TextField. This example is intended to show how to load a browser with data, how to read any data that is selected in the browser, and how to set up such an application in IB. As you will see, not much code is necessary, but there are a few tricks.
First, let's take a look at the finished product:

As you can see, a pretty simple application.

Building the Application

Please note: You must be running 2.0 to use the NXBrowser class.
Start IB in your usual way. Create the application's UI by dragging a TextField from the basic views palette and a NXBrowser from the scrolling views palette into the application's main window. (Note: The scrolling view palette is new in the 2.0 version of IB.) Size all the elements to get the desired effects. I chose to do a few extra things like eliminating the close box in the main window; your application can do what you want. The real purpose

someone's face along with the other attachments in NeXT Mail. Maybe in release 3.0…

Why All the Fuss?

Setting aliases is an extremely useful feature of UNIX mail, and NetInfo supports it in a similar fashion. Allowing users to type "Bob" instead of "bob@foobar.uta.edu" is much easier on the human users of electronic mail. The BSD UNIX way of supporting aliases is handled in a backhanded fashion, but that NeXT supports it at all says something about the "Not Invented Here" syndrome and NeXT's commendable efforts for compatibility. Since NetInfo is a NeXTstep feature, it represents the best route to assigning aliases to a NeXT computer's mail router. Individuals can still create their own pseudonyms for other users (in the .mailrc file if they use the BSD UNIX mail program from the shell, or in the ***Addresses*** panel in NeXT Mail), but these methods are not as transparent and sophisticated as using NetInfo. Until further notice, the picture database is effectively limited to local networked NeXT users.

For further information:

UNIX System Administration Handbook, Nemeth, Snyder, and Seebass. Prentice-Hall. Complete discussion of sendmail and it's administration. ISBN 0-13-933441-6.

!%@:: A Directory of Electronic Mail Addressing And Networks, Frey and Adams. O'Reilly and Associates. How to address e-mail; complete map of major networks and the Internet. ISBN 0-937175-39-0.

sendmail, aliases, newaliases, niload, and nidump man pages.

Introduction to the NXBrowser Class in 2.0

William Barker
Systems Administrator
Department of Biological Structure, SM-20
University of Washington
Seattle, WA 98195
wrb@ulnar.biostr.washington.edu

Overview

Many of us are familiar by now with some form of a browser. Using a browser in our code wasn't supported in 1.0, however there have been some unsupported browser classes available. Thankfully this situation has been addressed in 2.0 with the introduction of an official browser class.

I've been beta testing 2.0 on my cube at the University of Washington for the last couple of months. During that time, I've been exploring some of the new programming features found in this release. A browser object would be very useful for some of the research we do here. The lack of a supported browser object in 1.0 has been a drawback. Not anymore.

The browser in 2.0 is a set of tools for building, supporting and doing fun things with browsers. Interface Builder (IB) now includes a browser object that makes building browsers into the user interface (UI) a snap. In addition, two Application Kit classes--NXBrowser and NXBrowserCell--provide support for using browsers in your applications.

Definition of a Browser

The best example of a browser is right under the NeXT icon on your dock. The Workspace Manager uses a browser for displaying the file system. An Open panel uses a browser to display a list of file names in a similar way. Just about any list of items can use a browser.

The 2.0 documentation describes NXBrowser this way:

NXBrowser provides a user interface for displaying and selecting hierarchically organized data such as directory paths.

new data to it using Edit. The Addresses alias capability begs the question:

Do the NetInfo aliases override the Addresses aliases?

"Sendmail will only recognize aliases that are either in NetInfo or in the flat file. The aliases file [~/.NeXT/.mailalias] that is kept around for Mail doesn't influence sendmail. Thus if an alias appears in Mail's aliases file, but not in NetInfo or the flat file, the mail will be returned with user unknown."

The flat file is the /etc/sendmail/aliases file. In other words, the entries made in ***Addresses*** (under ***Private Users***) are merely book markers, and if they are not used to address a message via the cc or To: buttons on the ***Addresses*** panel, then they serve no other purpose in life, and they do not actually contribute to the alias database anywhere else in the system. ***Private Groups*** *are* checked for an alias match and are swapped for "real" addresses. It is important to note that NeXT Mail will not allow a ***Private Group*** to be specified if it is also a NetInfo group alias, and all NetInfo group aliases are automatically inserted into the ***Addresses-Private Groups*** area.

I wanted the hierarchy cleared up and stated baldly, so I asked:

With three potentially different alias databases, which is king, queen and prince?

"My recommendation: only use NetInfo to maintain consistency. The aliases file for Mail is updated from NetInfo using MailManager, don't play with it by hand."

By this time it should be clear that NetInfo is the *major domo* method of determining alias decoding on the NeXT. The BSD UNIX mail method of specifying aliases is supported in a backhanded way, and should be avoided. A bridge between the old and new ways of dealing with aliases is to run the shell script below to make changes in the alias database without running NetInfo:

```
#! csh: Change Aliases (9/11/90) Deltos Fleet Computing
##NOTE: This assumes your host is local; replace the '.'
with your host name if on a net!
nidump aliases . > aliases
vi aliases
niload aliases . < aliases
```

Non-gurus should use NetInfoManager. The utility of the shell script is in being able to execute it quickly from a shell. If you don't use shells, then don't bother. Besides, real UNIX gurus use EMACS, and *not* vi. ■

The Picture Database

Finally, aliases affect one other area of NeXT Mail: the picture database. Apparently, NeXT Mail only knows how to put the picture of a local user on a mail message. The author asked if there was any way to offer external user's images:

*I want the picture database to show in my Mail app. So far, only messages I generate on local accounts, for which I have pictures stored in **/LocalLibrary/Images/People** show in Mail. Mail sent by the very same account names but from different hosts do not get the same picture icon, and mail from external hosts' accounts do not get their picture shown even though the account name is aliased to the TIFF image in NetInfo.*

"Mail pictures are only supported for local Mail users, however there is a work-around that might work, but is not officially supported. Let me know if you are interested in the details."

The work-around is to make local aliases of all external messengers. In all honesty, it is not worth the trouble. Life would be much easier if NeXT would simply send the TIFF file of

The *members* property holds the actual mail address of the alias (see figure 3.) in the Value browser. Both properties are easily altered in this browser panel in NetInfoManager.

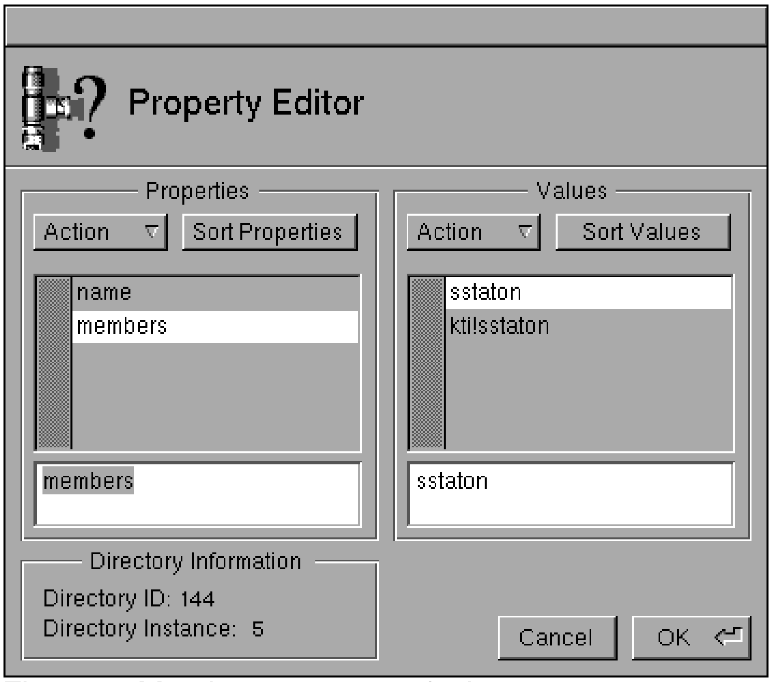


Figure 3. Members property of alias.

This article has already covered three of the four ways to establish mail aliases on the NeXT, and at this point, some important questions arise. There is a substantial overlapping of features in the BSD and NetInfo methods for providing aliases to mail users. In researching this article, the following questions came up (which were sent to Ask_NeXT):

*Does NetInfo automatically override **/etc/sendmail/aliases**?*

Andreas Sautter of NeXT replied for Ask_NeXT:

"I haven't tried this in 2.0 yet, but in 1.0, you could theoretically use /etc/sendmail/aliases to store your aliases. I strongly recommend that you use NetInfo for all your aliases and leave /etc/sendmail/aliases untouched."

Obviously, NeXT wants everyone to use NetInfo instead of the BSD method. The implicit answer appears to be "yes." This led to a more specific question about the priority of the BSD method versus NetInfo:

*When I do a **newaliases** command from the shell, does this tell NetInfo to look at the **/etc-/sendmail/aliases** file, or does it merely tell NetInfo to overwrite the same file with its internal alias file? What does it do?*

Andreas replied:

"What happens is that the sendmail program in 1.0 looks in both the flat file and in NetInfo to find aliases information. Again, don't rely on the fact that it does, NetInfo is the place to look for aliases."

Newaliases operates independently of Netinfo, and the data file it maintains is used separately by sendmail.

The fourth and final area in NeXT Mail where aliases are specified is within the NeXT Mail application itself. Under the ***Addresses*** button is a NetInfo-styled database browser that allows individuals to add their own aliases. Essentially, this feature of NeXT Mail emulates the .mailrc feature of BSD UNIX mail. This feature inspired some more questions for Ask_NeXT concerning the priority of ***Addresses*** versus NetInfo:

Where does Mail store the stuff you enter in Addresses?

Andreas replied:

"This is documented in Mail's Help system: Advanced --> Pictures."

Well…sort of. The actual location of the NeXT Mail aliases file is in ~/.NeXT/.mailalias. The file is an ASCII file that is world readable and follows the format of a NetInfo dump of the aliases database. Presumably, it is possible to add

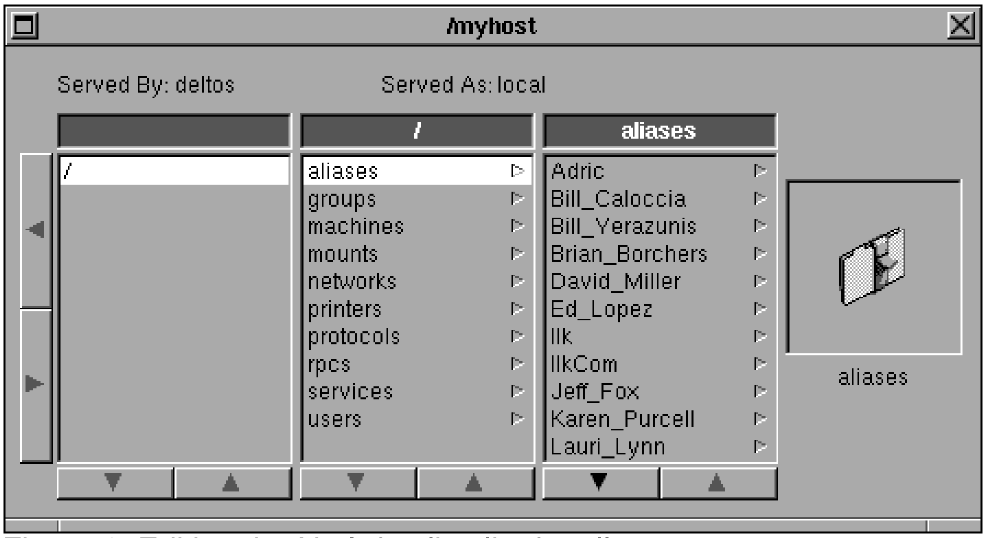


Figure 1. Editing the NetInfo alias list for all users of system Deltos.

Once a new entry is made (for example, "Steven_Staton[3]"), it has two Properties: *name* and *member*. The *name* property holds all the variations of the alias name (typically the case changes and variants of a name; see figure 2.) in the Values browser.

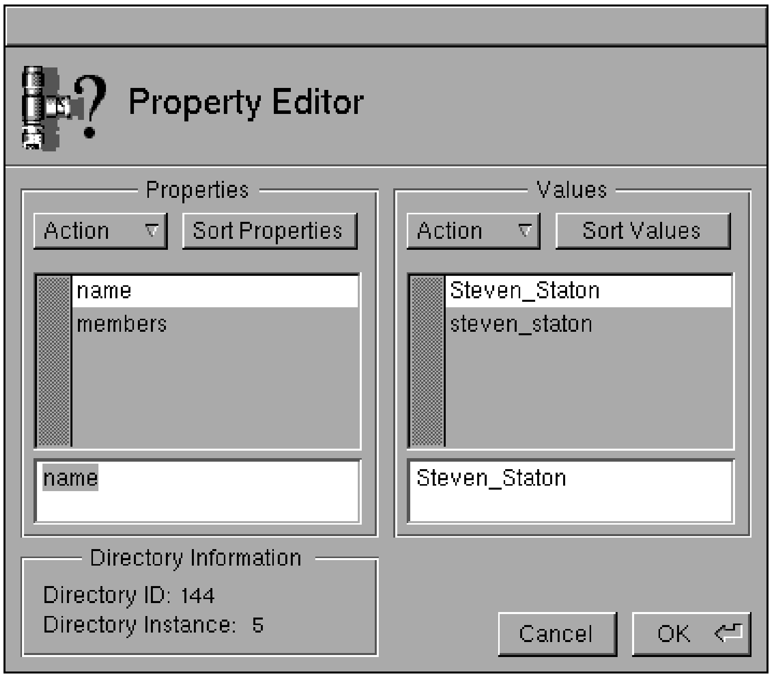


Figure 2. Name property of alias.

[3]The underscore is required - spaces are not allowed in alias names.

the message to where it is going. Local mail is routed through the local mail daemon, and external mail is sent via UUCP and the rmail command.

It is sendmail that is responsible for deciphering aliases. To do this, sendmail checks a file called /etc/sendmail/aliases for the current list of aliases. A program called newaliases updates the binary database file that sendmail uses to identify aliases. To add a new alias, the system administrator adds the alias to the /etc/sendmail/aliases file and issues the newaliases command[1].

Each individual user can create his own list of private aliases by editing the .mailrc file in his home directory (the directory one logs into). A line of the form:

alias Steve_Staton deltos!sstaton@blackbox.busl.com

instructs mail to translate the string "Steve_Staton" into the target address within the context of UNIX mail. These aliases should not conflict with the sendmail alias list, and are only specific to a given user within the local UNIX shell (the aliases do not work for other users).

NeXT NetInfo Aliases

NeXT changed the rules with their variant of BSD 4.3 UNIX. In order to group together all the various UNIX system database files that a system administrator must handle in organizing a UNIX machine, NeXT created the NetInfo database. NetInfo holds all the various mail, password and other system administration parameters of a given system (or systems) in a shared database that other NeXT computers can examine over the network. This allows any NeXT user to log into a network account on any physical NeXT computer in a NetInfo network, and access all his mail and account information. NetInfo is a great idea, but it causes some confusion among those more familiar with the older standard of BSD UNIX.

NetInfo maintains the alias list in an internal data structure that is normally only accessible from NetInfoManager,[2] a NeXTstep application found in /NextAdmin. To load new values into the NetInfo alias database, the system administrator runs NetInfoManager and edits the aliases database, thus adding system wide aliases (see figure 1).

---

[1]See the manual pages for aliases to discover the format of this file.

[2]NetInfo data can be extracted from the UNIX shell with the nidump command.

68030-based computers in the field, and can be purchased separately.

"When we unveiled a family of four new products, all 68040-based, on September 18 in San Francisco, we made some aggressive shipping commitments that we are now meeting," said Steven P. Jobs, president and CEO of NeXT. "Our volume shipments of NeXTstation should also put to rest any doubts about the Motorola 68040 being a reality."

"At our September introduction, we announced 15,000 first-day orders," said Todd Rulon-Miller, NeXT's vice president, sales. "Now we can start delivering on these orders, as we aggressively pursue even more new sales."

NeXTstation is more powerful and compact than the company's original NeXT Computer, which was based on the 68030 processor. The new computer's retail price of $4,995 also makes it the most affordable NeXT computer and gives it one of the most impressive price/performance ratios of any computer in any class.

NeXTcube, with a suggested retail price of $7,995, is based on the one-foot cube of the original NeXT Computer. The new NeXTcube offers users the greatest number of options within the NeXT family in terms of expandability, storage, memory and pathways to high-end color. It can be used as either a network file server or a high-end desktop computer.

Both products run NeXT's Release 2.0 system software and include the NeXTstep graphical user interface and development environment; NeXTmail, NeXT's multimedia electronic mail application; the unified imaging model of Display PostScript, which makes images on the screen appear just as they will look when printed; built-in thin and twisted-pair Ethernet and TCP/IP networking; a 2.88 MB floppy disk drive as standard; digital signal processing (DSP) for CD-quality sound, signal and image processing, and voice recognition; and an improved MegaPixel Display that weighs less than its predecessor and which adds a built-in microphone.

NeXT Computer, Inc. designs, manufactures and markets "Interpersonal Computing" professional workstations that promise to enhance the way groups work together in the 1990s. NeXT sells its products directly and through selected retail and VAR channels in North America, Asia and Europe. The company is headquartered at 900 Chesapeake Drive, Redwood City, California, 94063.

Using NeXT Mail with Aliases

Steven Staton
Deltos Fleet Computing

The NeXT™ Computer provides a remarkable improvement in interpersonal communication with the introduction of NeXT Mail, a NextStep® based electronic mail system that functions on top of standard BSD UNIX™ mail. While NeXT Mail provides advanced capabilities like Lip Service™ and the ability to insert any file or document into a mail message, all the fundamental operations of NeXT Mail are funnelled to standard UNIX mail. Because of this, there are some features of UNIX mail that are best left to NeXT Mail.

In particular, both forms of mail service offer the ability to define aliases, which are pseudonyms for real mail addresses. For example, the author's Usenet mail address is deltos!sstaton@blackbox.busl.com. Another user on a different system might refer to that address as "Steve" (in other words, as something that a human remembers as opposed to what a computer requires). UNIX mail provides two services for doing this, and NeXT Mail provides two additional ones. How do you decide which is right for a given need?

Let's find out.

BSD UNIX Aliases

BSD UNIX uses a program called sendmail to decide where to route mail messages. It is responsible for identifying the destination of a message and for sending a message to the appropriate program responsible for transferring

instead be an inset and have a "live link," such that I could double-click the image in the WordPerfect window and end up editing the chart in Diagram!. Then I could save the chart and be placed back into working on my WordPerfect document.

Insets are not currently possible on most machines—yeah, yeah, System 7.0, if it ever happens—but the NeXT software architecture makes it nearly trivial. A good general inset standard will boost integration and thus productivity. As with any bright high-tech idea these days, insets are being actively pursued by a variety of companies developing new-generation operating systems.

NeXT has a chance to take yet another lead here. Fortunately, NeXT understands the need and opportunity and I think we'll see something soon—probably before 3.0.

The banquet was the highlight of the conference for me. After we convinced the hotel security thugs that my associates would not cause a scene similar to the one in the bar, we got to sit through a brilliant presentation by Bill Buxton, from the University of Toronto. Bill is a musician, in addition to a computer researcher. He has a problem with computers: he thinks they stink. To his mind, accustomed to playing instruments that capture infinite nuances of action, computers communicate with people through the thinnest of sensual channels.

Bill offered a challenge: could anyone in the banquet hall make his $100k workstation capture the subtlety of expression that his saxophone could? Needless to say, not one person raised a hand. Then Bill presented some astounding examples of user interface technologies that he felt deserved to be considered for general implementation.

My favorite was a Xerox PARC video about using sound in interfaces. For the first time, I saw a UI that used sound without being obnoxious—heck, it almost felt natural to have "status sounds" droning quietly in the background. Watching this demo, I felt the icy finger of progress caress my spine. Once again, I got that feeling that we're on a train hell-bent for...well...your guess is as good as mine.

And once again, NeXT has an edge on this UI technology: the DSP. Steve Jobs has said he wasn't sure just what we'd use the DSP for, but he new in his heart that it was important. I think many people are getting an inkling of what to use it for. Hopefully, end users will soon reap the benefits.

By the way, some clever person reacted to the chilly environment of this year's conference, and made warmer plans for '91: Arizona. What are we trying to prove here?

Roger Rosner is director of engineering at Lighthouse Design, Ltd., creators of Diagram! and other NeXTstep products, and wishes it to be known that some of his best friends are lumberjacks. He can be reached by electronic mail at rock@lighthouse.com or ...!uunet!lighthouse!rock, by physical mail at 6516 Western Avenue, Chevy Chase, MD 20815-3212, or by telephone at 1-800-366-2279 or 301-907-4621.

Better Late than Never Department

NeXTstation Begins Volume Shipments

REDWOOD CITY, Calif., November 26, 1990 P NeXT Computer, Inc. announced today that it has begun on-schedule volume shipments of its new NeXTstation 68040-based computer. The 15 MIPS NeXTstation computer has a list price of $4,995 for a complete system that includes floppy and hard disk drives and NeXTstep, NeXT's revolutionary software environment.

In addition, NeXT announced that it has commenced shipments of its NeXTcube computer, a 68040-based version of its well-known cube-shaped computer, as well as 68040 CPU boards to upgrade its existing installed base of 68030-based computers.

NeXT also announced that its two new color computers, NeXTstation Color and NeXTdimension, will ship on schedule in Q1 1991.

All the new 68040-based products, including the upgraded CPU boards, contain NeXT's latest release of its software, NeXTstep Release 2.0. NeXTstep 2.0 is also compatible with all NeXT's

NeXT BOF at OOPSLA

Roger Rosner

Flying North for Winter

October in Ottawa is not nice. Mind you, Ottawa is truly a beautiful city, but it just happens to be about three feet south of the Arctic Circle—at least from my reptilian perspective. Nonetheless, some twisted individual chose to hold this year's Object-Oriented Programming: Systems, Languages, and Applications (OOPSLA) and European Conference on Object-Oriented Programming (ECOOP) conferences there and then.

Knowing that OOPSLA and ECOOP are the world's biggest and most important conferences on OOE (object-oriented everything), over 2000 well-insulated people attended. The conference attendance register read like a Who's Who in OOS (object-oriented society). The attendees presented scores of papers, and shared thousands of experiences, anecdotes, and cups of hot chocolate. Me, I mostly drank the demon gin, to numb my sense of the cold. Hot chocolate is nice for a little warmth, but does not come close to matching the blaze of glory and heightened level of understanding ignited by a few stiff shots.

The conference had an accompanying product show, where the coolest of OOC's (object-oriented companies) strutted their stuff. A one word summary: expensive. Based on prices alone, the OOR (object-oriented revolution) has yet to hit the main stream. Personally, I think some of these companies need to take the hint and produce mass market products—third round financing negotiations do not bring out the best in people.

Most prominent in the product booths were the OODB (object-oriented database, a real acronym) vendors. Obviously, OODB's will solve many tough problems and be big business someday, but for now they're too expensive and too incompatible. Useful standards built on real-world experience are a long way off. This however has not halted the proliferation of OODB companies claiming to offer the best of all possible solutions. Take my word on this: on an ice-cold projected market of less than ten million dollars in 1991, and with the wintry clouds of recession blotting out the sunlight, most of these companies are not going to be found at the 1992 OOPSLA.

NeXT did not have a booth. And most people I spoke to seemed ignorant of the basics of NeXT computers. I found this odd, as these same vendors and researchers were all scurrying around like rats trying to address many of the issues already successfully implemented (and, of equal significance, integrated) by NeXT. As an associate of mine said just before being thrown bodily from the hotel bar by a group of enraged lumberjacks, "NeXT has all the answers, but no one knows it."

However, the conference did generate a good deal of warmth for NeXT enthusiasts (and salespeople, software developers, etc.). NeXT Canada held a "BOF," conference lingo for a "birds-of-a-feather" session, wherein people sharing a particular interest gather to chat. On one day's notice, and with only a handful of posters, the BOF drew dozens more than our meeting room could hold.

The BOF started with a presentation of the new NeXT software and hardware by David Lavallee, a Canadian NeXT system engineer. After the preview, we split into two groups: novices interested in more demos, and NeXT programming veterans seeking hard technical information.

With representatives from NeXT's engineering department all the way to—no hissing now—Microsoft employees, how could the NeXT veterans BOF help but be productive? Well, by getting into philosophy mostly. For the most part, we spent our time arguing over such urgent issues as the marketability of a sea-of-objects paradigm, where application programs no longer exist, and the future of object persistence.

I had intended to turn the meeting toward discussing an inset standard, and toward getting some consensus about what we developers felt NeXT should do in 3.0 with regards to system services, Interface Builder, class libraries, etc. The inset standard, however, was my prime goal.

Insets are documents excerpted in other documents. For instance, if I create a structure chart in Diagram! and want to display it in WordPerfect nowadays, I copy and paste an EPS representation that has no connection to the original Diagram! document. Optimally, the pasted image would

```
}

- (id)copy
{
   return [super copy];
}

- read:(NXTypedStream *)stream
{
   [super read:stream];
   NXReadTypes( stream, "*", &stringPointer );
   return self;
}

- write:(NXTypedStream *)stream
{
   [super write:stream];
   NXWriteTypes( stream, "*", &stringPointer );
   return self;
}

- (const char *)name
{
   char    *type = "String";
   return type;
}
@end
```

[**Smartlist.h**]

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

@interface SmartList:List
{
}

- (id)swap:(unsigned)x :(unsigned)y;
- (id)sort;

@end
```

[**Smartlist.m**]

```
#import <objc/objc-runtime.h>
#import <objc/Object.h>
#import <objc/List.h>
#import <string.h>
#import "SmartList.h"
#import "Atom.m"

@implementation SmartList

- (id)swap:(unsigned)x :(unsigned)y
{
   id  temp;

   temp = [self objectAt:x];
   [self replaceObjectAt:x with:[self objectAt:y]];
   [self replaceObjectAt:y with:temp];
   return self;
}

- (id)sort
{
   unsigned k, j;
   BOOL sorted;

   k = [self count] - 1;
   sorted = NO;

   while( ( k > 0 ) && !sorted )
   {
      sorted = YES;
      for( j = 0; j <= ( k - 1 ); j++ )
      {
        if( [[self objectAt:j] isGreaterT
          han:[self objectAt:j+1]] )
         { [self swap:j :j+1]; sorted = NO;  }

      }
     k--;
   }
   return self;
}
@end
```

```
#import "Atom.m"

@implementation String

+ new
{
    self = [super new];
    stringPointer = (char *)malloc( 1 * sizeof( char ) );
    return self;
}

+ (id)newWithValue:(char *)temp
{
    self = [super new];
    stringPointer = (char *)malloc( (strlen( temp ) + 1) *
        sizeof( char ) );
    strcpy( stringPointer, temp );
    return self;
}

- (char *)assign:(char *)temp
{
    free( stringPointer );
    stringPointer = (char *)malloc( (strlen( temp ) + 1) *
        sizeof( char ) );
    strcpy( stringPointer, temp );
    return stringPointer;
}

- (char *)retrieve
{
    return stringPointer;
}

- (int)address
{
    return (int)stringPointer;
}

- (BOOL)isEqualTo:(id)otherObject
{
    if( !strcmp( [self name], [otherObject name] ) )
    {


if( !strcmp( stringPointer, [otherObject retrieve] ) )
    return YES;
else  return NO;
    }
    else
    {
        return NO;
    }
}

- (BOOL)isLessThan:(id)otherObject
{
    if( !strcmp( [self name], [otherObject retrieve] ) )
    {

if( strcmp( stringPointer, [otherObject retrieve] ) < 0 )
    return YES;
else
    return NO;
    }
    else
    {
        return NO;
    }
}

- (BOOL)isGreaterThan:(id)otherObject
{
    if( !strcmp( [self name], [otherObject retrieve] ) )
    {

if( strcmp( stringPointer, [otherObject retrieve] ) > 0 )
    return YES;
else
    return NO;
    }
    else
    {
        return NO;
    }
}

- (BOOL)isSequential
{
    return YES;
}

- (id)strcat:(id)secondString
{
    int len1, len2;
    char *oldStringPointer;

    len1 = [self strlen];
    len2 = [secondString strlen];

    oldStringPointer = stringPointer;

    stringPointer = (char *)malloc( ((len1 + len2) + 2) *
        sizeof( char ) );

    strcpy( stringPointer, strcat( oldStringPointer,
            [secondString retrieve] ) );
    free( oldStringPointer );

    return self;
}

- (int)strcmp:(id)secondString
{
    return strcmp( stringPointer, [secondString retrieve] );
}

- (id)strcpy:(id)secondString
{
    [self assign:[secondString retrieve]];
    return self;
}

- (int)strlen
{
    return strlen( stringPointer );
}

- (id)free
{
    self = [super free];
    return self;
```

```
      return (int)&c;
   }

   - (BOOL)isEqualTo:(id)otherObject
   {
      if( !strcmp( [self name], [otherObject name] ) )
      {

          if( c == [otherObject retrieve] )   return YES;
         else  return NO;
      }
      else
      {
         return NO;
      }
   }

   - (BOOL)isLessThan:(id)otherObject
   {
      if( !strcmp( [self name], [otherObject name] ) )
      {

if( c < [otherObject retrieve] )

   return YES;
else
   return NO;
      }
      else
      {
         return NO;
      }
   }

   - (BOOL)isGreaterThan:(id)otherObject
   {
      if( !strcmp( [self name], [otherObject name] ) )
      {
        if( c > [otherObject retrieve] )  return YES;
        else return NO;
      }
      else
      {
         return NO;
      }
   }

   - (BOOL)isSequential{return YES;}
   - (BOOL)isalnum{return isalnum( c );}
   - (BOOL)isalpha{return isalpha( c );}
   - (BOOL)isascii{return isascii( c );}
   - (BOOL)iscntrl{return iscntrl( c );}
   - (BOOL)isdigit{return isdigit( c );}
   - (BOOL)isxdigit{return isxdigit( c );}
   - (BOOL)isprint{return isprint( c );}
   - (BOOL)ispunct{return ispunct( c );}
   - (BOOL)islower{return islower( c );}
   - (BOOL)isupper{return isupper( c );}

   - (id)tolower{tolower( c ); return self;}
   - (id)toupper{toupper( c ); return self;}
   - (id)free{self = [super free]; return self;}
   - (id)copy{return [super copy];}

   - read:(NXTypedStream *)stream
   {
   [super read:stream];
   NXReadTypes( stream, "c", &c );
   return self;
}

- write:(NXTypedStream *)stream
{
   [super write:stream];
   NXWriteTypes( stream, "c", &c );
   return self;
}

- (const char *)name
{
   char    *type = "Character";
   return type;
}

@end
```

[String.h]

```
#import "Atom.h"

@interface String:Atom
{
   char *stringPointer;
}

// Class methods
+ (id)new;
+ (id)newWithValue:(char *)temp;

// Assigning and retrieving values
- (char *)assign:(char *)temp;
- (char *)retrieve;
- (int)address;

// Standard comparison methods
- (BOOL)isEqualTo:(id)otherObject;
- (BOOL)isLessThan:(id)otherObject;
- (BOOL)isGreaterThan:(id)otherObject;
- (BOOL)isSequential;

// ANSI C compatibilty methods
- (id)strcat:(id)secondString;
- (int)strcmp:(id)secondString;
- (id)strcpy:(id)secondString;
- (id)strlen;

// Memory management methods
- (id)free;
- (id)copy;

// Archive methods
- read:(NXTypedStream *)stream;
- write:(NXTypedStream *)stream;


- (const char *)name;


@end
```

[String.m]

```
#import <objc/objc-runtime.h>
#import <objc/Object.h>
#import <string.h>
#import "String.h"
```

```
    return NO;
}

- (const char *)name
{
    char *className = "Atom";
    return className;
}

- free
{
    self = [super free];
    return self;
}

- copy
{
    return [super copy];
}

- write:(NXTypedStream *)stream
{
    [super write:stream];
    NXWriteTypes( stream, "i", &bugStats );
    return self;
}

- read:(NXTypedStream *)stream
{
    [super read:stream];
    NXReadTypes( stream, "i", &bugStats );
    return self;
}

- startDebugging
{
    bugStats = YES;
    return self;
}

- stopDebugging
{
    bugStats = NO;
    return self;
}

@end
```

[Character.h]

```
#import "Atom.h"

@interface Character:Atom
{
    char c;
}

// Class methods
+ (id)new;
+ (id)newWithValue:(char)temp;

// Assigning and retrieving values
- (char)assign:(char)temp;
- (char)retrieve;
- (int)address;

// Standard comparison methods
- (BOOL)isEqualTo:(id)otherObject;
- (BOOL)isLessThan:(id)otherObject;
- (BOOL)isGreaterThan:(id)otherObject;
- (BOOL)isSequential;

// ANSI C compatibilty methods
- (BOOL)isalnum;
- (BOOL)isalpha;
- (BOOL)isascii;
- (BOOL)iscntrl;
- (BOOL)isdigit;
- (BOOL)isxdigit;
- (BOOL)isprint;
- (BOOL)ispunct;
- (BOOL)islower;
- (BOOL)isupper;

- (id)tolower;
- (id)toupper;

// Memory management methods
- (id)free;
- (id)copy;

// Archive methods
- read:(NXTypedStream *)stream;
- write:(NXTypedStream *)stream;

- (const char *)name;

@end
```

[Character.m]

```
#import <objc/objc-runtime.h>
#import <objc/Object.h>
#import <string.h>
#import <ctype.h>
#import "Character.h"
#import "Atom.m"

@implementation Character

+ new
{
    self = [super new];
    return self;
}

+ (id)newWithValue:(char)temp
{
    self = [super new];
    c = temp;
    return self;
}

- (char)assign:(char)temp
{
    c = temp;
    return temp;
}

- (char)retrieve
{
    return c;
}

- (int)address
{
```

elements in the list. With standard techniques this is all but impossible. It is hard to write a routine to make comparisons, due to the fact that you compare different types in different ways. An integer compares different than a string which is different that a character and so on. So I thought, well, leave it up to the data type to compare itself. Now a sort routine would be possible.

Bubble Sort example for Object List:

```
self = List

-sort
{
 sorted = NO;
 k = [self count] - 1;
 while( (k-- > 0) && !sorted )
 {
  sorted = YES;
  for( j = 0 ; j <= k ; j++ )
  {
   if( [[self objectAt:j] isGreaterThan:[self objectAt:j+1]] )
               <=- IMPORTANT
   {
    [self swapObjectAt:j with:j+1];
    sorted = NO;
   }
  }
 }
}
```

This is a simple Bubble Sort, but other algorithms could also easily implemented. The actual algorithm is not as important as the concept suggested by in the marked line. The phrase "[self objectAt:j]" returns an object in the list. Then that object checks to see if it is greater than the following object in the list. Past programming techniques have put all responsibilty on the algorithms. The algorithms need to retrieve the data, make comparisons, and possibly re-assign data. Now some responsibility rests with the data structure. The advantages stretch beyond comparisons. For instance printing, archiving, displaying, etc... could all be handled by the objects themselves. The possibilties are endless.

And here are a few objects that are included in the Data Block collection to give you an idea of what the library is all about.

[Atom.h]

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

@interface Atom:Object
{
   BOOL bugStats;
}

+ (id)new;

//- (void)assign:(void)temp;
//- (void)retrieve;

- (BOOL)isEqualTo:(id)otherObject;
- (BOOL)isLessThan:(id)otherObject;
- (BOOL)isGreaterThan:(id)otherObject;
- (BOOL)isSequential;

- (const char *)name;

- (id)free;
- (id)copy;

- write:(NXTypedStream *)stream;
- read:(NXTypedStream *)stream;

- startDebugging;
- stopDebugging;

//- displayToCLI;

@end
```

[Atom.m]

```
#import <objc/objc-runtime.h>
#import <objc/Object.h>
#import "Atom.h"

@implementation Atom

+ (id)new
{
   self = [super new];
   bugStats = NO;
   return self;
}

//- (void)assign:(void)temp
//{
//   return temp;
//}

//- (void)retrieve
//{
//   return self;
//}

- (BOOL)isEqualTo:(id)otherObject
{
   return NO;
}

- (BOOL)isLessThan:(id)otherObject
{
   return NO;
}

- (BOOL)isGreaterThan:(id)otherObject
{
   return NO;
}

- (BOOL)isSequential
{
```

Data Objects

Ed Hill

Data Structures + Objects = Programs

Nikalus Wirth's "Algorithms + Data Structures = Programs" was an invaluable resource for computer scientis in the 1980's and it is my feeling that Brad Cox's book, "Object Oriented Programming" will be the computing almanac of the 1990's. Both books present the idea of reusabilty. Wirth demonstrates the technique on a theoretical level, a simple algorithm can apply to different circumstances. Cox goes one step further, and shows us how actual code could be reused.

College courses teach data structures and how to implement all kinds of handy dandy algorithms. I have used Stacks, Queues, and Trees over and over again. The problem is that everytime I use a data structure, I rewrite the procedures to implement them. Not that I am too stupid to cut and paste old code into my new program, but I typically never need the exact data structure twice. Sometimes I want a tree of integers, the next time it's real numbers, character strings, etc... . The idea of reusability is still present. I use the same alogrithm for each traversal, but it is implemented in different ways.

OO languages give the current programmer tools that Wirth did not have. Two important features, Polymorphism and Inheritance, are missing in early languages. With the use of these new tools, we are able to take Wirths ideas and apply them in such a way so that code behaves more and more like an algorithm. By making one object called a list, the programmer would never have to re-invent the wheel, he could just use the list object over and over again. A list holds objects, not string, integers, etc..., and since everthing in the perfect OO world is an object, different versions of the same list are not needed. The reusabilty extends even further. A Stack is nothing more than a list with special features. Once the list object is invented, the programmer would only need to modify it slightly in order to make a Stack object.

I would think that traditional data structures such as these (Stack, Queues, Trees, etc...) would be the first objects to be implemented by the companies that create the class libraries. Some companies create libraries and ship them with the compilers (Smalltalk). Others try to squeeze extra money out of you (StepStone IC Libs.). NeXT gave us an extensible object library (Appkit, Soundkit, Musickit). The drawback is that with the exception of a couple of objects, these kits are useful only for taking advantage of NeXT specific features (graphic UI, sound, printing, etc...)

NeXT has laid the groundwork for a library of data structure objects. The List object could be the basis for the library that I would like to see. Some people think that the list object is useless in standard programming situations, because it only handles objects, so they use the Storage object instead. I propose a different approach. Wrap data structures in OO shells. That is, create OBJECTS such as Integer, Character, String, etc... . I will be the first person (besides Doug) to point out the flaws. #1 Bulk. By replacing and int variable declaration with an id, you will increase the size of your program. #2 Speed. Since assigning data is achieved through methods, the process has to go through another pointer, and thus slower by a constant (O(1)).

Now that I have outlined the obvious disadvantages, let me outline what I think are advantages that outweigh the flaws.

#1 Reuseability of common algorithms due to improved polymorphism (Sorts, searches, etc...).
#2 More robust applications in a smaller amount of time due to the reuse of standard objects.
#3 Better documentation providing information about the actual algorithms used to implement the Objects. For example: "Retrieval of the count of objects in class List is accomplish in constant time O(1), while the retrieval of an actual element is achieved in linear time O(n)."
#4 A built in OO debugger
Advantages specific to NeXT users:
#5 Conforms to appkit conventions for memory allocation and deallocation.
#6 Objects implement methods for reading and writing to typed streams.

My motivation is based on the following; I thought it would be great if I could write a routine that would sort a list independent of the actual

a good move for smaller developers who aren't sure about marketing channels to instead come up with a custom application. There are certainly people who make a good living at this with PCs, Macs and Suns. It seems like a real good opportunity because of the relatively small software body and the lack of specialized software. Someone with a good idea for a niche market application--office management, medical office, taxes, whatever--could come up with a software package, develop it, and then go around selling NeXT systems with the application. It is a means by which you create you own market for your software. However, I've never been in that marketplace, so I don't know what the dangers, pitfalls and other difficulties are.

Tell me about NeXT's present sales channels.
NeXT is in a bit of a quandary. I can't speak for the current situation, but most of the reports that I've gotten from people who've actually tried to work with Businessland to buy NeXT are fairly negative. They say that Businessland has done a really poor job of marketing the NeXT. One could argue about what they had to work with, but I'm sure the situation was a frustration for NeXT. What I expect you'll see happen is far more active direct sales by NeXT representatives and a big push for VAR channels. But it's still a big problem for them. Case in point. I had a call from one of our company executives, who lives back in New Jersey. He has a friend who has come over to visit and has seen his NeXT many times and now wants one of his own. This friend can't find out where to buy one. He's having no luck with the local Businessland. He tried to contact the New York NeXT representative and had no luck there. I finally had to call up the NeXT representative here in San Diego in order to help this person. Without those broad distributions channels, it is a real trick for NeXT to penetrate the market. I don't know if NeXT is locked into Businessland, or if they could switch. I don't know if they could find anyone else to switch to. I'm not sure what their problems and limitations are.

Are the problems in marketing, distribution, and production of a financial nature, as some have suggested, or is NeXT simply understaffed?

I seriously question that there is a financial problem. The biggest problem NeXT had was simply the time schedule of the 68040 chip and the limited numbers of chips that they could get in November and December. In the meantime, they not only had their initial 15,000 orders, but they had more orders coming in after the introduction. Since the introduction, they have continued to sell systems, and in their case, it's almost an embarrassment of riches: they have more orders than they can deliver. I think they'll catch up probably by the end of this quarter, by the end of March, and be on a roll. Again, I have my ear on the ground in a lot of places inside and outside of NeXT. They are being financially conservative, but they've been financially conservative all along. They've run a fairly tight ship. They've kept salaries down. They've kept personnel down. This is a firm that's trying to compete with Apple and Sun, who have 12,000 employees each, and it hasn't been that long since NeXT went over the 400-employee mark. Their total employee base for design, sales, marketing and production is smaller than some divisions within Apple and Sun, and that again is another constraint. There is a classic problem with a company that's trying to grow, especially if it's trying to put out a product. There are some key points in the growth curve that are actually dangerous. You're trying to increase staff , and so need the revenues to support your staff; but if you don't have enough staff, you can't develop, market and support your product well. It is a real 'catch-22' situation. I've seen a lot of companies, both software and hardware, founder on that transition--trying to go from a small firm to a big one--and a lot of them just don't make it. If there is anything inherently dangerous to what NeXT is doing, it's this situation. But that's not unique to NeXT or indicative of any serious financial problems. It's just that it's a tough thing to do, and a lot of companies don't make it through that curve. NeXT has the advantage of a lot of cash in the bank. I think they're using it very frugally, and I believe that will get them through.

There are bookstores presently selling NeXTstations for $3300 to $3500. That is an incredible price considering what you are getting. You've got the 17-inch monitor, 8 megabytes of RAM, 105 megabyte drive, tons of software. It's a stunning price. And it's the target price. The original target price was to have a NeXT computer out there for around $3000 to $3500 for the students, and their original machine came in at double that price. But they've reached that target now, and I'm willing to bet that there are going to be a whole lot of NeXT systems sold on a whole lot of campuses. You've got a machine that is cheap enough, quiet enough, reliable enough, and powerful enough. And it has all the right tools to entice a lot of universities and a lot of departments to buy large numbers of these. College students will also sacrifice and spend what is essentially one semester's tuition on a system like this. The irony is that more than ever you are going to see NeXT get big on campus, even though some analysts had concluded--prematurely--that NeXT had abandoned the university market.

With the wide spread acceptance of Windows 3.0 and, subsequently GUI's, how can NeXT continue to differentiate itself?
Essentially, NeXT can differentiate itself just by having someone run Windows and then use a NeXT. I mean, the differentiations is simply quality: quality of the software, quality of the display, quality of the hardware. Even as we talk, I'm sitting in front of a '386 running Windows 3.0. With the exception of some key technologies like DDL(dynamic data link), the Windows interface is about where the Macintosh was five years ago. It is not cutting-edge technology; it's frankly a memory hog, it's slow, and it's crude. I was talking recently with a programmer who is developing a Windows application for a major software vendor. He told me that he can't even compile his application under Windows. Windows is so resource-intensive that he has to get out of it to do a compilation and then get back into Windows and launch [the compiled application]. That's really stupid. I have been somewhat suspicious of all the great cries of how Windows is going to take over and how Windows is going to harm the Macintosh or NeXT or whatever. What I've heard from a number of sources, including two or three software companies which are developing Windows applications, is that all their market research says that Windows is selling well but that Windows applications are not. People use Windows and play around with it, but when they go to run an application, they run their old MS-DOS applications. That may change as some of the major applications come out set up for Windows, but it raises some serious questions. A journalist that I was talking to was actually raising issues about the Macintosh versus Windows running on a '386 or '486 clone. When I listed the pricing and configuration of a NeXTstation he came away very impressed and decided he wanted to take a look at it.

Is it still possible, with the industry maturing and installed base such a key factor in purchasing decisions, for a piece of software, such as Improv, to sell machines?
Yes, I think it's still possible. I think on the corporate level what you're going to see is key people--executive power users--who suddenly decide to get Improv, and if they want Improv, they're going to have to buy a NeXT. This is only a limited number of sales within a corporation. But what's going to happen is that once you have this person with a NeXTstation using Improv, WordPerfect, the laser printer and other NeXT utilities, you are going to start seeing corporate envy. People will say, "He can do this, and he can do that. Hey, gee, I should get one." It is actually the same strategy that Apple used to get into the corporate market, where you had key people, usually graphics arts production types, introducing the Mac into the corporation. A good friend of mine in Utah literally smuggled Macs into his workplace and used what the machines were producing to impress others. I think that you're going to see much the same curve with the NeXT.

Should more developers consider becoming VAR's, and do you have any information on how NeXT organizes this process?
I don't know the process offhand. It may actually be

would be well over double what NeXTdimension costs. And you might have a hard time getting the boards to all work together. There is nothing like the NeXTdimension board in either the Macintosh or PC world. Again, Apple, Microsoft, and IBM have been running around, raising hoopla for multimedia, and I think NeXT has just stepped in and said that this is the best multimedia platform you can get for the money.

Is multimedia a viable market, or do we still need to wait?
The problem with multimedia depends on what you mean by the term. If you are talking in terms of selling CD-I or CD ROM drives to the home market, it is not there yet. The people who have made the best stab to date are Commodore, but given their track record, I'll have to wait and see how successful they are. They have released a version of the Amiga 500 which looks like a stereo or video component; your standard sleek black box with a CD ROM drive built in. It's going have to be something at the component level where people have it sitting on their VCR's or in their stereo stack. It can't be a 'clunky' computer that someone has to drag out, with cables strewn all over the unit. In terms of home multimedia, I think the market isn't going to develop until we see more than what we've got. With the other type of multimedia, business multimedia, the biggest problem is that you are increasingly overwhelmed as more and more tools become available. Well, let me give you an analogy from our product. The problem we're addressing is that it's easy to turn out an ugly document. What we are offering is a tool to help make it easier for turning out a good looking one and harder to turn out a bad looking one. When you get to video and multimedia and slide presentations, it is much easier to create something that looks stupid; it requires much more work to create something that looks good. You can put a lot of time and effort into sixty seconds of a multimedia presentation just trying to get it to look good, and it may well be that a product parallel to ours, in the sense of something that's a multimedia expert system, may be what's required to make multimedia a real viable market.

One component that had tremendous implications for multimedia was the magneto-optical drive. Now that this drive is no longer bundled, will we see a cheaper MO drive, may be the 125 megabyte in addition to the 2.88 floppy?
I'm sad to see the optical drive go. I'm not sure why the technology was so slow to develop in speed and low-cost, but I can understand why it didn't catch on. Part of the problem was corporate consideration. NeXT was in a bind. To really successfully make the corporate penetration, they had to have easy media-exchange between NeXT systems and IBM PC's. That may the single most compelling reason for the switch to the floppies. The second most compelling reason is the cost of media for third party software firms. A start-up like NeXT is an opportunity for small software firms to come out with innovative and relatively inexpensive software and market it, and those $50 to $100 opticals were a real stumbling block to a lot of firms. It was a nightmare in terms of production, costs, and inventory. This is sad, because of the comment Steve [Jobs] made--which bears out my own experience and that of most NeXT users that I know--is that people who actually have the NeXT like the optical. I like having this device that I can keep backing up on for a good long time. I have archives going back almost two years, stored on three or four opticals. I'd love to see NeXT offer a 125 MB 3.5-inch optical as a drive option. Again, for whatever reason, the price in the whole optical drive market place is steep. Have you ever tried to price optical drives for PC's or Mac's? They're expensive. People gripe about the price of the NeXT, but the peripherals for PC's and Mac's are horribly expensive. You're looking to spend three to five thousand dollars just for your optical disk drive. So, until those price can come down, I think optical drives will remain an expensive option.

When NeXT first started, it was going to be the academic computer, and students were going to carry their worlds around on optical disks. Where is this commitment now, and is there room for growth in this market?
I think you are going to see NeXT become more viable as an academic computer than ever before.

processor. He didn't say any of that, but his answers left that impression. What I tried to explain was that there are certain times that you want to dedicate a processor to a given task--such as imaging for a printer--so that it has no impact whatsoever on the operation of the rest of the system. Like it or not, there are real-time CPU-intensive tasks that would be best served by having a processor all to themselves, and it would be nice to get them to run without having the rest of the system slow down.

Wasn't the DMA system designed to manage the bottleneck issue?
DMA helps with moving data around, but I'm talking about system tasks. For example, you have issues such as graphics acceleration, where if you have a dedicated processor (as with the NeXTdimension board), then you have graphics performance that isn't dependent upon how many CPU cycles you can get. By contrast, when you start dragging windows around on the NeXTstation Color, it is sluggish simply because you're moving more bits around. If you go to print something, especially a long document, you get hit with these little pauses as the bitmap imaging process drains resources. It would be real nice to simply plug in a board of one more more extra processors and have them do what is required. You might have a network server, or you might be trying to run real time data acquisition or transaction processing. These things are very sensitive to any kind of system slow down and require a great deal of resources.

Let's talk hardware. Will these new machines give NeXT the market share that they and developers have been waiting for?
I think the biggest problem NeXT has right now is production. I think that they could sell more machines and do more advertising if they had a higher production rate. I don't know what the current production situation is; I've heard reports that there have been NeXT employees who gave up their Christmas vacations to go to the factory and help build machines to try to catch up with the backlog of orders. NeXT's biggest problem is ironic. After a year and a half of having a problem of demand, they suddenly have a problem of supply. That's a better problem, because it's more easily solved, but it's still a problem. I think what is going to make it on these new machines is price/performance. NeXT has got to do a better job of getting the word out, but they're in a catch-22 situation. They really don't want to get the word out until they have the capacity. It could be very devastating to create demand and not be able to fill that. I think there have been a few other problems with NeXT, especially one which I've harped on to NeXT for close to two years now: I think that they have done a very poor job of dealing with the press. It is always interesting for me to talk to other computer journalists and to note the degree of hostility and, to put it bluntly, the ignorance, in terms of not really knowing what's being offered by NeXT, what the products are, what the prices are, and what's being bundled. I think if NeXT really wants to survive short-term or long-term, it is going to have to do a much better job of addressing this issue.

"I think the biggest problem NeXT has right now is production. I think that they could sell more machines and do more advertising if they had a higher production rate"

What is the significance of NeXTdimension, and what types of applications would best exploit its capabilities?
It has been very interesting. I've been doing "The State of the Mac" column for Macworld, and there have been a number of columns dealing with video and the Macintosh: static and real time image compression, the JPEG algorithm, mixing video and computer generated graphics, having TV and live video appear on the screen, and graphics acceleration. Suddenly, NeXT has come up with a board that embodies most of what people have been trying to do on the Macintosh and, at a price that is substantially less. To get the same functionality of the NeXTdimension on the Macintosh, you would need three to five different boards, and the total cost

and proof of concept. What happened was that along the way people became excited about the NeXT version, and we've continued to pursue it. We still have plans to develop for other platforms, but our initial target is the NeXT.

"Custom Software...may be very viable. More tools need to be out there especially more object libraries for custom development to be really effective"

What type of software category would you advise a present or prospective NeXT developer to target?
I think custom software, which NeXT is promoting, may be a viable market. However, more tools--especially more object libraries--need to be out there for custom development to be really effective. There also needs to be better books and better documentation on how to program the NeXT. This lack of information is frustrating, since the NeXT is the best development environment that I've worked on. As a consequence, when I want to do something that I haven't done before, it is often very difficult to figure out exactly how to do it. It is not immediately clear that, yes, this is the approach that I'm supposed to take, and this is how it works.

You mentioned object libraries. When will "software IC's", as envisioned by the founders of Stepstone Corporation, finally become a reality?
I have no idea what Stepstone is doing, especially in terms of their contribution to the entire NeXT effort. I do know that Stepstone has been pushing ahead in the MS-DOS and UNIX markets, and, at one point some months ago, they were looking for someone to port their compiler to the Macintosh. I don't know if they're going to become a great contributor to object libraries on the NeXT; I think that's going to come from other people. What I would very much like to see is all the NeXT development tools--the Objective C compiler, gdb, and all that--available on other platforms. If someone could port the gcc and gdb utilities over to the Macintosh and PC, I'd be thrilled. In fact at a previous job, I purchased Stepstone's Objective C Compiler for MS-DOSl, which turned out to be a preprocessor for Microsoft C and not a native code compiler. There isn't much in terms of debugging support for developing with this compiler on the PC. If I had the same Objective C tools running on a PC--with a good debugger, like gdb, a good compiler, and a common class library--I could build my own Application Kit equivalent on the PC and on the Macintosh, which would make my code more portable. I don't know if anyone is doing anything along those lines.

Would it be in NeXT's best interest to have that level of portability?
Ultimately, I think so, though they run the risk of encouraging their developers to quickly port things over to other systems. Still, that situation would encourage developers who wouldn't have otherwise developed for the NeXT to say, "Yeah, I can develop for the NeXT and also get the other platform." It's an issue we face: we are doing our development in Objective C on the NeXT, but we are not sure what will have to be done when we start moving to other platforms. My basic feeling is that we will have to transfer over on an object level rather than on a source-code level, rewriting the program in C++. Then, at some point down the line, we'd back-translate it to C++ on the NeXT.

Do you have any idea about what we can expect to see in System 3.0?
I can tell you what I'd like to see. I'm not sure we'll see it. This is something which I've rattled a few bars about, but the response I've gotten hasn't been encouraging. I'd like to see multiprocessor support, a situation where Mach's multiprocessing capabilities are fully implemented. You have a NeXTcube, you stick in a second board with several processors on it, and the operating system just starts distributing the load. In a question-and-answer period last fall, I brought up this issue with Steve Jobs. He could have been trying to throw me off the trail or just stating their position, but my impression was that what they focusing on a single high-speed CPU, such as a very fast '040 or possibly a RISC

and, in the next three months, did a complete draft of the book, all written on the NeXT using WriteNow, version 0.85. After finishing the first draft, I had to port it over to the Macintosh so that Addison-Wesley could do something with it. When System Software 0.9 came out, Addison-Wesley was extremely anxious to ship the book before someone else had a chance to bring one out. They wanted to print the 0.8 version. I didn't want to do this, and NeXT basically told them that it would not be acceptable. I revised the manuscript for System 0.9, and Addison-Wesley had tremendous turn-around: I turned in my last changes and screen dumps in June and got a printed copy of the book back five weeks later in July. That's why it was pretty much dragged out . I'm really just now starting the 2.0 revision; I got the Release 2.0 software a few weeks ago.

Will version 2.0 of your book be available for Digital Librarian?
It's still going to be a paper medium, but we've had some discussion with NeXT and, also, one third party vendor about making the whole book available in electronic form as well.

As a developer, what has NeXT done to facilitate your process, and if you had a few requests for additional services, what would they be?
Actually, as a developer, I have not yet made too many demands on NeXT. In terms of marketing, our company has been receiving a fair amount of support and attention from NeXT with regards to our product. Our company is in the process of obtaining our own technical representative - someone inside of NeXT with whom we will be working.

"In terms of marketing, our company has been receiving a fair amount of support and attention from NeXT"

Tell me what you are developing at Pages.
We are developing, to use a somewhat misleading phrase, a page-layout expert system. It is not designed to compete with full-blown page-layout systems. Instead, it's intended to allow the average end-user to quickly and easily lay out a document and have the assurance that it will be well-designed. It has a built in design model, that is, a set of specifications for twenty-one graphic design elements and a set of styles for each of those elements. There are also built-in rules determining when those elements can be used, how they can be combined, and where they can be placed on a page. All this works together to make it, we hope, very easy and fast to layout a document and basically make it look good. As anyone who has used conventional desktop publishing software knows, it is all too easy to create a poorly-designed document.

Should we expect to see more products in the future incorporating expert systems and eventually certain aspects of artificial intelligence?
I hesitate to use the term "expert system" because that implies more intelligence than is there. It is not something that's sitting there and advising you per se; the expertise is in the design model. You take an expert graphic designer, who creates a design model embodying his or her expertise, then embed that design model in the software. That allows the user to create documents that are well designed and are done in the style of that graphic designer. We have our own default design model that comes with the package. The package will also be able to accept models done by others, so that third-party firms and corporate designers can release their own design models. And people can go with a different look or feel for their documents based on the model that they're using.

How did your firm decide to develop for NeXT?
Initially, our decision wasn't aimed specifically at the NeXT. The people who developed the concept for our product have been working on it for well over a decade and had considered developing it on a number of platforms but never had much success. I came on board to oversee development and, at first, used the NeXT primarily for prototyping, designing the user interface, testing out several algorithms,

```
 Unknown) F=$L/Unknown;;
 Webster) F=$N/Webster;;
 sketchdemo) F=$DA/SketchDemo;;
 *) echo open: Unable to open $1; exit;;
esac
echo $F
shift
/usr/bin/open $F $*
```

Interface
Pat Rick

An interview with Bruce F. Webster,
Author of the NeXT Book.

With last September's release of the newest generation of machines, numbers of questions have arisen concerning the viability and objectives of this small band of Northern California innovators. In business, as inwar, discretion is the better part of valor -- or the dollar. Consequently, many of the answers to one's question just aren't forthcoming from NeXT. Hey, they're busy changing the world. We'll do sushi when they're finished. For the time being, if you want to get a real feel for where this firm of the the1990's and beyond is heading, you might talk to one of the best NeXT advocates (Advocate: a sacred word within NeXT which has the same effect as evangelist does on another silicon valley company) found anywhere outside of NeXT itself.

This person is Bruce F. Webster, author of the now famous ***The NeXT Book*** *(Addison-Wesley, 1989.) Bruce Webster has written numerous articles for a variety of computer magazines and is currently a contributing editor to both NeXTworld and Macworld. In addition to his career as journalist, Bruce Webster has also been an instructor of Computer Science at the prestigious Brigham Young University and a software developer for a number of firms including Pages, Inc. where he is presently vice president of R&D.*

What follows is a condensed version of an interview held in early January between Pat Rick (Contributing Editor, NUJ) and Bruce Webster. Certain passages have been edited for clarity and conciseness, but the editors have worked to maintain the original tone and tenor of the dialogue. We believe Bruce Webster has many interesting insights, and we hope that the information provided will be of use to our readers. To anyone at NeXT who reads this article, please consider some of the suggestions made below because they just might help you become that billion dollar company sooner rather than later.

How did you first make contact with NeXT?
In December 1987, I was contacted by Steve Stansel of Addison-Wesley. He told me that I was under consideration to write a book on a very innovative new computer product, but that he couldn't give any details. I indicated to him that I was interested. The following month, in January, while I was at the Macworld Exposition, I met again with Steve and Carol McClendon and discussed particulars. At this time, they did acknowledge that this book was to be about the NeXT computer and two weeks after that I made my first visit to NeXT. It was a rather lengthy process from the time that I first visited NeXT to when the book first came out: roughly 18 months.

Will version 2.0 of your book demand as much time as the original?
I hope not. The first book took awhile for a number of reasons. At the very start, I spent a fair amount of time at NeXT. I attended a very small developer camp consisting of ten companies and myself in March 1988. Thus, I received some good background on the basics of the machine and was actually doing some programming with their development environment which was then running on Sun workstations. Also, I saw prototypes of the machine and had some private discussions with Steve Jobs and others about what the features of the machine were going to be. After that, I had to sit back and wait a number of months before I was able to get any more information from NeXT. Not until the machine's announcement in October did I receive any real information. I attended the developer seminar the day after the introduction and then finally in December of '88 actually got a NeXT system to start working with. That same week, I sat down and started working on the book

boy Mike!)

Erica,
How’s it going? I just got back from COMDEX. It’s been a long, exhausting week. Incredible response level from the floor.
I’ve been slack about writing Tao, due to my killing scheduling. Probably I won’t have any more issues until Feb 1991, when I finish everything that’s on my plates now.
Robert Lin

Robert : February, she is acomin’ and we’ll be awaitin’ to see what TAO is going to do next. Take care! Erica

Some minor items for buzzings and such:
1.The article on installing 4MBytes simms in the latest Buzzings was interesting. However, it’s no longer necessary to go to all that trouble. It is now possible to purchase 4 MByte Simms that are no larger than the 1 MByte simms that come with the NeXT and simply plug in without difficulty. I have 8 such simms in the machine I am using now, along with 8 1MByte simms, for a total of 40 Mbytesk.
The vendor I used is
South Coast Electronics
10920 Wilshire Bl. Suite 1100
Los Angeles, CA 90024
213-208-3260, 800-289-8801, Fax: 213-208-3282
Ask for SC84000-08 300mm chips)
2.The NeXT implementation of "tip" is non-standard and won’t work if you don’t have a DIRECTORY with name /usr/spool/uucp/LCK writeable by the effective user of tip.
3.I have found the appended very simple (abbreviated here) shell script very useful. It enables me open application from within unix, without hunting for the mouse, typing long names etc. I call it "open" and have it in a directory in my path ahead of /usr/bin. So, to open the Chess applicaton, for example, I simply type "open Chess". If I wanted to alias "Chess" to "chess", that would be easy. There are certainly better ways of doing this, but not many simpler ones. The time spent searching a shell script of this length is negligible compared to the rest of the time required for opening an application.
dgc

David G. Cantor
Department of Mathematics
University of California
Los Angeles, CA 90024-1555
Internet: dgc@math.ucla.edu

The "open" shell script:

```
#!/bin/sh
D=/NextDeveloper/Demos
DA=/NextDeveloper/Apps
L=/LocalApps
N=/NextApps
case $1 in
 3-d_Rotation) F=$L/3-d_Rotation;;
 3-d F=$L/3-d_Rotation;;
 Billiards) F=$D/Billiards.app;;
 Blancer) F=$D/Blancer.app;;
 Chess) F=$D/Chess;;
 DataViz) F=$D/DataViz;;
 Draw) F=$D/Draw;;
 Edit) F=$N/Edit;;
 Emacs) F=$L/Emacs;;
 FrameMaker) F=$D/FrameMaker.app;;
 Henon) F=$L/Henon;;
 HitchHiker) F=$L/HitchHiker1.0;;
 Icon) F=$D/Icon;;
 InterfaceBuilder) F=$N/InterfaceBuilder;;
 RealTimeAlgebra F=$D/RealTimeAlgebra;;
 Saturn) F=$D/Saturn.app;;
 Shell) F=$N/Shell;;
 Stealth) F=$D/Stealth.app;;
 TeXview) F=$L/TeXview;;
 TopologyLab) F=$D/TopologyLab.app;;
 Twin) F=$L/Twin;;
```

Dick Jensen

Dick : Sorry about the problems with the printing. No one else has reported similar problems. Since the issue is in Writenow format, it should be loadable and printable from a Macintosh, if the postscript doesn't work. Since last issue, I have learned to fix my mouse (article next issue) but my printer and disk drive continue to be problematic. My SE should be coming to take a look this week. My fingers are crossed. -- Erica

Erica :

Actually, NeXT extended the three month warranty to one year, so your warranty might be expiring right about now ...

My machine was delivered in September of 1989, on the same day our campus received the 1.0 systems OD's. (I think they were holding my system, waiting for 1.0, which in retrospect I am grateful for.)

I leave my machine on essentially all the time. I only turn it off whenever I reboot, which is maybe once every couple of weeks when the swapfile gets too big. I have never had to reboot because of a system crash. When my machine is not in use, I use Bryce's LockScreen 2.1 to blank the screen.

My 330M Maxtor died about two months after the original three-month warranty ran out, which was fortunately one week after I received a letter from NeXT saying my warranty was extended to one year. (Good timing there.) They replaced the drive and the new one has had no problems the nine months since. That was the only time I ever rebuilt my hard disk.

My OD has exhibited no problems whatsoever. I use it once a day to back up my hard disk. It has never been in a smoking environment.

My screen is almost always dimmer than it can be, even though the brightness (key) is set all the way up. I'm used to this, but I know it can be brighter since sometimes it will intermittently flicker into a bright state. It does this more just after cycling the power for a reboot.

My mouse seems just fine, but on the public NeXTs here, the mice are prone to multiple clicks from a single click (very annoying in an environment where a double click can launch an application, which is a slow and laborious process).

Also the OD on one of the three public NeXTs has died. The screens on the three have not shown any degradation, even though only the default screen blanking is used on them. One of them, however, has been fuzzier than the other two since the beginning.

News flash: just today, the public cubes will not allow OD's to even be inserted! They just put in new OD filters, so that may be related, but I'm checking into that ...

On another note, I much prefer the WriteNow format for the Journal, especially since I can print it in 2Up format. (I don't have Frame, and don't intend to get it.)

Thanks for your excellent work on the NeXT User's Journal.

Mark Adler

madler@piglet.caltech.edu

Mark : No such luck with my warranty. It is LONG since expired. However, Mike Popinski, my SE will be taking a look this week. (Atta

dimmer and seem to be holding up well. The mice, use without pads, are also doing fine--although someone did steal one of the balls. I'll let you know if trouble develops.

By the way, what problems have you had with Framemaker? I was thinking of getting it for some DTP. -Charlie Fletcher

*Charlie : See comment s below. Everyone does seem to love that screen dimmer utility. Now that I have been using Gefilte for 24 hrs/day, I better download it and start using it! Framemaker now works again under full 2.0 release with the patches for Frame 2.0**d**. If you don't have this patch level yet, contact your SE or school service representative. Personally, I adore the package and am still very grateful to Frame Inc for donating a copy to BuzzNUG. I do amazing quantitites of work in Frame and am very sorry that the freebie-demo was not included in the 2.0 release -- thus making us revert to Writenow. Sigh.*

I look forward to your effective Journal efforts and am sorry that you are having problems with your machine. I'm having trouble with printing it. The machine I had worked great, but did have some failures.

I received the NeXT I worked with in February 1988 and had an OD and later added a 330mb hard disk. The OD had problems but it was promptly replaced by NeXT. (ca. two days) The Hard Disk started sounding as if the bearing(s) were going out and before they did, we replaced it (again NeXT sent a new one in one day and the UC Santa Barbara support staff replaced it immediately, reformatted it and reloaded the stuff we'd stored from the OD. (I was at UC Santa Barbara until recently.)

We left the machine running all the time, turned it off very seldom, but ran the 'lock' product developed by Bryce at Oregon State as a screen saver [a very wise $3 investment.]

The machine was used by me as a workstation, as a UNIX server from remote terminals, and as a route from ethernet to a serial outport via tip. It may have crashed twice or three times during that 20 month period. The printer would mysteriously stop working but that appears to have been primarily due to the loss of the daemon (can't tell you why we lost it, but restarting it would get it up again.)

In short, the mouse, the keyboard, the screen and the connections all worked fine and continue to function well. The HD and OD problems were troublesome, but taken care of without charge.

I just wish I had one now rather than running this PS/2 50Z as a dumb terminal to a UNIX server which I have to depend on for PostScript formatting.

Alas, since I am not using a NeXT right now, and am having trouble printing the NUJ11. My printer works from the last to the front with pages 42 to 20 having produced pages without any eps (maybe there weren't any) and then there were problems. I separated pages 1-4 and printed those, but 5-20 are creating problems. Perhaps others have problems?

Best wishes for a continuation of a wonderful effort. The WriteNow format leaves a lot to be desired compared to what you had before, but beggars...

Thanks.

additional features that are very useful for all NeXT users:

1. The ability to function either with a separately available external SCSI floppy drive OR the new NeXT internal drive (which means new NeXT owners get all of the features of FloppyWorks without the purchase of a floppy drive).

2. The ability to support Macintosh 800K disks (with the purchase of a drive that can access the 800K format).

3. The ability to perform character translations to or from NeXT, DOS and Macintosh floppies.

4. The ability to make exact duplicates of floppy disks. This feature is very handy for duplicating a floppy for distribution.

5. A panic button that allows you to reset the internal or external floppy drives if a floppy gets stuck. The only other alternative is the guru-intensive use of the disk -e command, or the forceful application of a paper clip and a machine reboot.

6. The ability to launch itself automatically when a Macintosh disk is inserted into the external or internal floppy drive. This feature makes FloppyWorks very convenient to use.

Considering the range of capabilities provided by FloppyWorks that are not offered by the current NeXT software and hardware, it is clear that there is still plenty of need for FloppyWorks, especially for those users who need access to Macintosh floppies on the NeXT. More information about FloppyWorks and the available floppy drive options is available from Digital Instrumentation Technology, Inc., at 505-662-1459 (ask for Liz Shrum).

I hope this article has been helpful for those of you who want to use the floppy capabilities offered by the new NeXT software and hardware, and for those of you who need the additional capabilities offered by FloppyWorks. We welcome any comments or suggestions you may have.

David Lewis
Software Development Manager
Digital Instrumentation
Technology, Inc.
72317.3304@compuserve.com

David : again, thank you for your prompt response, your time and effort in talking to our readership and for your courtesy and helpfulness when we talked. We look forward to hearing more from you and the others at DIT. -- Erica

Erica-- After ordering thousands of dollars in NeXT equipment I find out it will only last a couple of years, say that is good news!! (I ordered the ed special 030 -> 040 deal.)

I thought I would answer your 'Welcome Column' in the latest NUJ. Although I have resently ordered a cube, I have been using the cubes in our library computer center since they came in (shortly after they were introduced.) There are six cubes there. The library is dusty so (of course) all six eventually had failed OD's (they have been replaced or cleaned, I don't know which, and work fine now.) I have noticed no other problems. The machines are on 24 hr. a day, seven days a week. The monitors use the standard NeXT

FLOPPY WAS INSERTED. DIT uses an OEM supplier for both the original floppy drive and the new floppy drive. It was the manufacturer's decision to initiate the "motor-on" function when the drive is polled. There are distinct advantages to this system (notably, an assurance that the drive and software are functioning properly), but it would be nice if the drive didn't "click". In any event, the new drive does not click.

3. THE BROWSER WAS TERRIBLE. We admit that this interface is not as smooth as it could be. In the first release of the operating system Application Kit, NeXT chose not to include certain very useful interface objects (such as the Browser). As we did not have access to network archive sites (where some third-party implementations of the NeXT browser could be found), we made an effort to provide a browser that was as useful as the NeXT browser and integrated nicely with our interface. We chose not to take the time to re-invent the NeXT browser because we knew it was going to change for the V2.0 release,and that a browser object was going to be offered. While we believe that our current browser is adequate, the NeXT browser is a feature that we plan to offer in a future release of FloppyWorks.

4. THE BROWSER WAS SLOW. The Browser is now quite fast, especially on the 68040 systems.

5. THE PROGRAM DISPLAYED A PROGRESS WINDOW THAT WOULDN'T GO AWAY. We agree that this can be annoying, but it is consistent with the NeXT interface to provide a "modal window" informing the user of what is going on (the NeXT 1.0 Browser did this). As far as I know, modal windows are always placed at the front of the Workspace Manager's window list (the Print Panel is a common example of this),and there is no easy way around it. We are, however, concerned about interference with other applications, and are we are considering other ways to provide status information to the user (like the "Processes" panel that the NeXT 2.0 File Viewer provides)

6. TRANSLATOR OPTIONS WERE NOT SAVED. FloppyWorks now saves translation filter options automatically on a most-recently-used basis. Filter settings are saved separately for each UNIX login account.

7. UNIX FLOPPIES WERE NOT SUPPORTED. FloppyWorks supports UNIX floppies. Since UNIX floppies are now supported by the NeXT operating system, the main use for this capability is to apply translation filters.

8. MACINTOSH DISKS WERE NOT SUPPORTED. Macintosh disks are supported by FloppyWorks (including 800K disks, which require the purchase of a special floppy drive because of the way the Macintosh writes these disks). This includes access to the resource fork, where many fine treasures are stored. As noted earlier, FloppyWorks also automatically generates the file type and creator when writing a file to a Macintosh floppy.

In addition to answering the concerns expressed in the earlier review, FloppyWorks offers some

by the same program on another machine. Yet another difficulty for users is that NeXT decided not to support the Macintosh filesystem in the 2.0 version of the operating system. This feature may not appear for quite some time, since the Macintosh filesystem presents several complications that make it non-trivial to support as an NFS-mounted filesystem.

This is where FloppyWorks comes in. The CubeFloppy program reviewed in Volume 11 of NUJ was a bundled hardware/software package, intended to provide the floppy hardware and software features that the NeXT1.0 operating system and the original Cube lacked. In light of the new products being offered by NeXT, the product has been significantly enhanced for use with both the original Cube and the new NeXT hardware offerings, and was renamed FloppyWorks. The floppy drive is now optional, and the FloppyWorks software can be used with new NeXT internal floppy drive, which means that an owner of the new NeXTcube or NeXTstation needs only the FloppyWorks program to access Macintosh floppies, and to apply a full range of character translation filters (such as CR-LF to LF, Strip CTRL-Z, LF to CR, etc.) when transferring to or from NeXT, DOS, or Macintosh floppies. To provide additional flexibility when accessing Macintosh disks, FloppyWorks allows you to read or write either the data or resource forks of a file on a Macintosh floppy, and automatically generates the proper file type and creator for most NeXT file extensions when transferring from the NeXT to a Macintosh floppy. For the real NeXT guru, FloppyWorks allows you to add new extension/file-type/creator combinations using NeXT's dwrite facility for writing to the defaults database.

For those who have the old Cube or need an extra drive, DIT's floppy drive has been upgraded to support the new 2.88Mb floppy format. As well, a special floppy drive is now available from DIT that is able to read the 800K Mac format (and all 720K and 1.44Mb disk formats) for those who need access to floppies produced by the older Macintoshes (as most Macintosh users have discovered, the 800K disks are written with a unique disk format that requires the use of a special floppy drive).

The review of CubeFloppy mentioned above pointed not so much to a lack of functionality, as to a lack of performance and features. Below are listed the problems with CubeFloppy pointed to in this review, and how they have been addressed by its new incarnation as FloppyWorks:

1. THE DOS TRANSFER SOFTWARE WAS SLOW. The original review of CubeFloppy covered an early version of the software. The later versions of the DOS software for the 1.0 operating system performed quite well. In fact, the newer DOS software in many cases performed faster than that in the NeXT 2.0 release. However, because of the architecture of the NeXT 2.0 file system software, FloppyWorks must use NeXT's DOS code for accessing DOS floppies.

2. THE DRIVE CLICKED WHEN NO

transferring his floppy-based files onto the NeXT, the product overall was not as convenient and useful as he would have liked.

As the manufacturer of CubeFloppy (now known as FloppyWorks), we found it helpful to have feedback on our product, and we have greatly improved the product since the aforementioned review was published. We thought it would be useful to the readers of the NeXT User's Journal to bring them up-to-date on our current product offering. I will also take this opportunity to communicate some technical issues related to the use of any floppy drive with the NeXT computer.

First, because the review of CubeFloppy that appeared in NUJ was written prior to the NeXT product announcement last September, I will take a moment to bring you up-to-date on the current level of support for floppies on the NeXT computer. As all of you know, every NeXT now comes with an internal floppy drive supporting the 720K DOS and NeXT, the 1.44Mb DOS and NeXT, and the 2.88Mb NeXT floppy formats. NeXT is currently the only vendor I know of shipping a product that uses the 2.88Mb format, and there are no current standards for 2.88 DOS or Mac formats. The 2.0 version of the NeXT operating system provides automatic NFS mounting for DOS filesystems on 720K and 1.44Mb floppy disks. This means that if you stick a DOS floppy in your new NeXTcube or NeXTstation, the DOS filesystem is mounted just like a UNIX filesystem, and a floppy icon pops up in your File Viewer (NFS is Network File System, a licensed Sun technology that provides a filesystem-independent way to ship files and filesystems transparently over a network). Most standard NeXT operating system calls (and thus most user application features) operate on an NFS-mounted DOS filesystem, although DOS filesystem restrictions apply. For instance, you cannot name a file on an NFS-mounted DOS filesystem with a filename longer than 8 characters or an extension longer than 3 characters. Programmers will want to know that the scandir(3) function will not work on DOS filesystems in the first 2.0 release,and they should use the directory(3) functions if they will be accessing DOS filesystems.

Although NeXT now provides an impressive level of support for DOS-based floppy filesystems, some difficulties remain. The most troublesome problem for users is the issue of file formats. Although the NFS-mounted DOS filesystem gives users access to DOS filesystems, it is strictly binary access. The program accessing the DOS filesystem is responsible for performing any character conversions and format conversions that may be necessary. This means if you bring a DOS C-source file over from your PC, you still have to take the additional step of converting CR-LF to LF and stripping CTRL-Z's, since the C-compiler on the NeXT is not forgiving to the PC end-of-line and end-of-file conventions. Other programs, such as Wingz, WriteNow, and FrameMaker, are very forgiving if you are using a file produced

EDITORIAL GUIDELINES

The NeXT Users' Journal is looking for incisive, intelligent articles focussing on how-to, review and about NeXT technical issues. Our readership primarily consists of developers (academic, commercial and government) who are sophisticated programmers. Code is always cheerfully welcomed and highly desired.

Please copy-edit your piece as much as possible and post to *erica@kong.gatech.edu*.

You may also mail or phone
Erica Liebman
1150 Collier Road NW L-12
Atlanta, GA 30318
(404) 352-5551

(Calls between 10AM and 10PM only, please indicate if you can be called back collect).

If you wish to have your software/book/etc reviewed, please contact me at the above address or phone number. We will make every effort to get your review into the next issue.

If you have software we at NUJ could make use of and would be kind enough to donate a copy, again send to the above address. Your contribution will be duly noted as loudly as possible in as many issues as possible.

Please see the NeXT Users' Journal Manifesto article about advertising, author rewards, subscriptions by mail and so forth.

Submission Guide for Articles

1. Start with Title, Subtitle (if included) and Author each on a separate line. Author contact information including phone, address and e-mail should be included on separate lines below author name and may be repeated in greater detail with an author bio at the end of the article.
2. Include carriage returns only at the end of paragraphs.
3. Please submit articles in WriteNow, Rich-Text or ASCII format for now.
4. Avoid passive voice. Please spellcheck. Please copy-proof before submitting.
5. Send articles to erica@kong.gatech.edu.

If you do not get an "Acknowledgement" or "ACK" back within seven days, please call or resubmit.

FEEDBACK from the TRENCHES

Letters, Hints and Other Fun

I just realized that you are doing this editing stuff on a volunteer basis! I guess I have a billion things going on at the same time too, but this must be quite a time sink for you!

I really appreciate your work, and hope for guten Fortschritt on your Studentenarbeit! -Avery Wang

Danke sehr. Ich still am getting die fier punkt nulls im Klass, so meine Studentenarbeit ist gut aber seit Ich so viel arbeit zum tun, diese NewsBrief ist sehr spat. Shade! - deine Erica

Hi; Has anyone in your Next group acquired and used a DIT CubeFloppy on the Next?

If so, have they had frequent system crashes since they installed the Cube Floppy for no apparent reason?

I suspect these DIT boxes somehow induce the Next computer to crash. Do you have any information or news on this? I'd appreciate your help. J. Rial

Take a look at the letter below from the DIT people. I suspect you are running a very old copy of the software.

SUBJECT: Floppies and you on the NeXT computer (Macintosh, too!)
Dear NeXT User's Journal readers:

We recently came across a review of our CubeFloppy disk drive and software product in the NeXT User's Journal, Number 11. The author indicated that although he found CubeFloppy to be useful for

NeXT Users' Journal

Issue 12 - December/January 1990-1

CONTENTS

Contributers

Erica Liebman, Pat Rick, David Cantor, David Lewis, Ed Hill, Roger Rosner, Steven Staton, William Barker, Philippe Provoste, Steve Sieczko, Ron Weissman, Mike Ferris, Pat McGee, Conrad Geiger, Chuck Herrik

Welcome

Erica J. Liebman
erica@kong.gatech.edu

Despite all rumors to the contrary, I am neither dead nor living dead. Lots of stuff has happened since November. I passed my core exams. I got bronchitis. I fnished my PhD breadth requirements. I got the flu. School started again. In short, I have been one sick and busy puppy. Sorry to all I have dropped out of touch with. Thank you to all those who called and comforted me in my hours of coughing, sniffling and phlegm.

Due to the prolonged time between the last issue and this, this one will be especially rich and varied. I have a piece on Image Processing in here myself and have more set for the next few issues including one next month on NeXT Security.

Lots of things have happened over the last two months, but few of them are, well, newsbreaking. The emphasis is on follow-through and production. In the case of Lotus and Insignia Solutions it has been delivery as promised of final release software. On the part of NeXT it has been cranking up production and getting those machines rolling down the assembly lines to fill all the advance and new orders for the slabs.

NeXT-World premiered in a glorious splash. The magazine is sharp, professional and elegant -- and highly recommended. Buzz-buddy Dan Lavin is one of the strong voices behind NeXT World and we Buzzies wish him and the N-W crew the best.

This issue again is in Writenow format due to the general positive feedback from our readers in that 1. they could print it out and 2. they could read it, alter it and save it without owning a license to Frame and 3. it could be read, etc under 2.0. We are also using 12 point fonts as the basic reading font so that you can still read the text even on a "fit-to-page" Writenow screen. If you agree with this change and like it or if you prefer otherwise or even have other ideas for deliver media, please (as always) write.

On a very serious matter, we need every one of you to pitch in and write articles. Rewards for doing so are described in the NeXT Users' Journal Manifesto later this issue. Changes and goals are detailed in this article. These include Mail Subscriptions, Pleges of Support, Advertising, Columns and such. Unless we start getting some money into to the Journal, we will be having some deep problems. Please read carefully.

Thanks again to everyone for their support of the Journal!