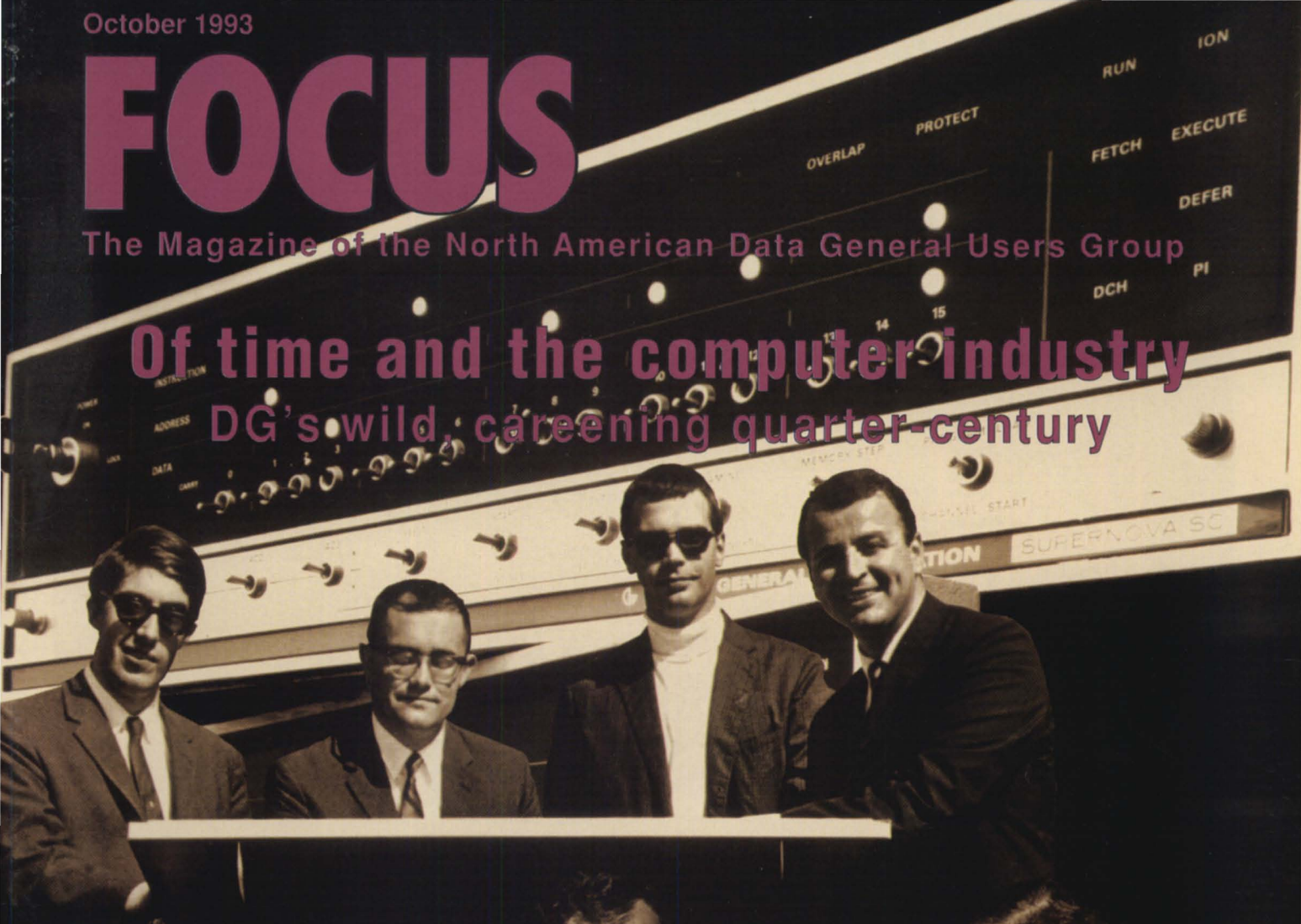
October 1993

FOCUS

The Magazine of the North American Data General Users Group

Of time and the computer industry

DG's wild, careening quarter-century



Plus

A debugger to fall in love with

ICobol: Legacy for the future

Aviion process and network management

Unix Notebook: Nostalgia

Bulk Rate
U.S. Postage
Paid
Permit No. 38
Fulton, MO

Attention all INFOS® users

OPEN SYSTEMS

Become a SuperHero overnight

FLY TO OPEN SYSTEMS WITH U/FOS™

- the first UNIX implementation of INFOS
- smooth, rapid migration of your COBOL, Fortran and PL/I applications
- full INFOS functionality
- automatic data migration
- choice of AViiON and other major UNIXs
- added functionality and better utilities
- API for third party software products

Huge savings on redevelopment and retraining costs.

Proven migration expertise

With seven years' experience, Transoft are experts in software migration and are a leading international supplier of Open Systems tools to the Data General community. Our UBB Universal Business Basic product – nearly 5,000 copies sold to date – has enabled DG users worldwide to migrate to UNIX. And our powerful U/SQL Universal Structured Query Language operates on a variety of COBOL, BASIC and RDBMS file types.

The same – only better

With U/FOS you can achieve a smooth, rapid migration of your INFOS applications to UNIX, and also utilise powerful additional features – including better data compression, better checkpointing and RDBMS-level data integrity and recovery. SQL reporting is also planned, using our proven U/SQL product.

Benefits and savings now

Why wait? Get the financial and operational benefits of Open Systems now, plus U/FOS improved functionality with RDBMS-level transaction security – and make massive financial savings on redevelopment and retraining.

To obtain further details and arrange a personal demonstration contact:
TRANSOFT INC, 1899 Powers Ferry Road, Suite 420, Marietta, GA 30067, USA.
Tel: (404) 933 1965 Fax: (404) 933 3464
TRANSOFT LTD, Nash House, Datchet Road, Slough, SL3 7LR, England.
Tel: 0753 692332 (Int + 44 753 692332)
Fax: 0753 694251 (Int + 44 753 694251)

TRANSOFT®
Portability and Productivity
Software Tools for Your Trade

Circle 48 on reader service card.

FOCUS

The Magazine of the North American Data General Users Group

Cover design by Ann Soto and John Houser

FOCUS, the Magazine of the North American Data General Users Group (ISSN 0883-8194) is the official publication of the North American Data General Users Group (NADGUG) in cooperation with Turnkey Publishing.

Editorial and business offices are at P.O. Box 200549, Austin, TX 78720, phone 512/335-2286. NADGUG headquarters are located at NADGUG, c/o Danieli & O'Keefe Associates, Inc., Chiswick Park, 490 Boston Post Rd., Sudbury, MA 01776, phone 508/443-3330.

FOCUS, the Magazine of the North American Data General Users Group is distributed to members of the North American Data General Users Group. Membership fees are $60 per person ($110 outside North America). A one-year (12 issues) subscription to *FOCUS, the Magazine of the North American Data General Users Group*, costs $48. For memberships and subscriptions outside the U.S., add $50 to defray the cost of mailing.

The cost of single copies is $4. Requests to replace missing issues free of charge are honored only up to six months after date of issue. Send request to *FOCUS, the Magazine of the North American Data General Users Group*, c/o Turnkey Publishing.

NADGUG is an independent association of computer users; it is not affiliated with Data General Corporation, nor does it represent the policies or opinions of Data General Corporation. The views expressed herein are the opinions of the authors, and do not necessarily represent the policies or opinions of NADGUG or of Turnkey Publishing, Inc.

Advertisements in *FOCUS, the Magazine of the North American Data General Users Group* do not constitute an endorsement of the advertised products by NADGUG or Turnkey Publishing, Inc.

Volume 9, Issue 10.

NORTH AMERICAN DATA GENERAL USERS GROUP

DATA GENERAL IS OPENING A LOT OF EYES WITH THE BEST OPEN SYSTEMS SERVER IN THE WORLD!

The AViiON 9500– for performance range, high availability, data integrity, storage management and price/performance. It's simply the best open systems server in the world!

AViiON®–The complete open enterprise solution:
Flexibility–Choose the size server you require today.
Scalability–As your needs expand, the AViiON 9500 grows to 1600 MIPS of power.
High Availability & Data Integrity & Storage Management– The combination of AViiON servers, CLARiiON™ storage products and the DG/UX™ operating system is unmatched.
Price/Performance–AViiON is a recognized industry leader.
AViiON–Impeccable credentials:
A $1 Billion customer base in less than four years. Voted #1 for Customer Satisfaction in Computerworld's "CW Guide to Servers."*
AViiON–The world's best server!
From a company that understands your enterprise computing needs. Call **1-800-DATA GEN**

Data General
Where the World is going!™

Circle 13 on reader service card.

Something for everyone

by Calvin Durden
Special to Focus

"Something for everyone," is what you will find at NADGUG 93.

The theme of this year's conference is "Training, Networking, Technology," and we covered all the bases. From PCs to MVs, from open systems to proprietary systems, from technical to conceptual topics, from general information to future trends, if you use Data General equipment we've got something for you.

Thinking of moving from MVs to open systems?

Then attend our full-day executive seminar on downsizing.

Want to find out about open systems?

Then attend our full-day "Introduction to Open Systems" seminar.

Need to connect PCs to your MV or Aviion?

We've got just the session you need.

Trying to decide which technology on which to bet your future?

Toss that crystal ball and attend one of our "Future Trends" sessions, such as wireless LANs, FDDI and ATM, document imaging, storage, and Windows NT. Plus, we have a wide range of "how-to" sessions for you MV and Aviion users.

Something new this year is the integration of the half-day Data General Education seminars into tracks of the conference, and making them FREE. Class size is limited, on a "first come" basis, so sign up early.

Also new is a special Healthcare Symposium, hosted jointly by NADGUG's Healthcare Special Interest Group and Data General on Wednesday, October 27th. Industry experts will address a wide range of topics of interest to healthcare professionals.

NADGUG 93 will take place in Atlanta, Georgia, October 25-28. It marks a new era for the user group and a new look for our conference.

With the SIG meetings and roundtables moved to Monday, any hot issues, problems, or questions that come up can be presented to the proper persons for answer or resolution over the course of the conference.

This people-to-people networking alone is worth the cost of attending.

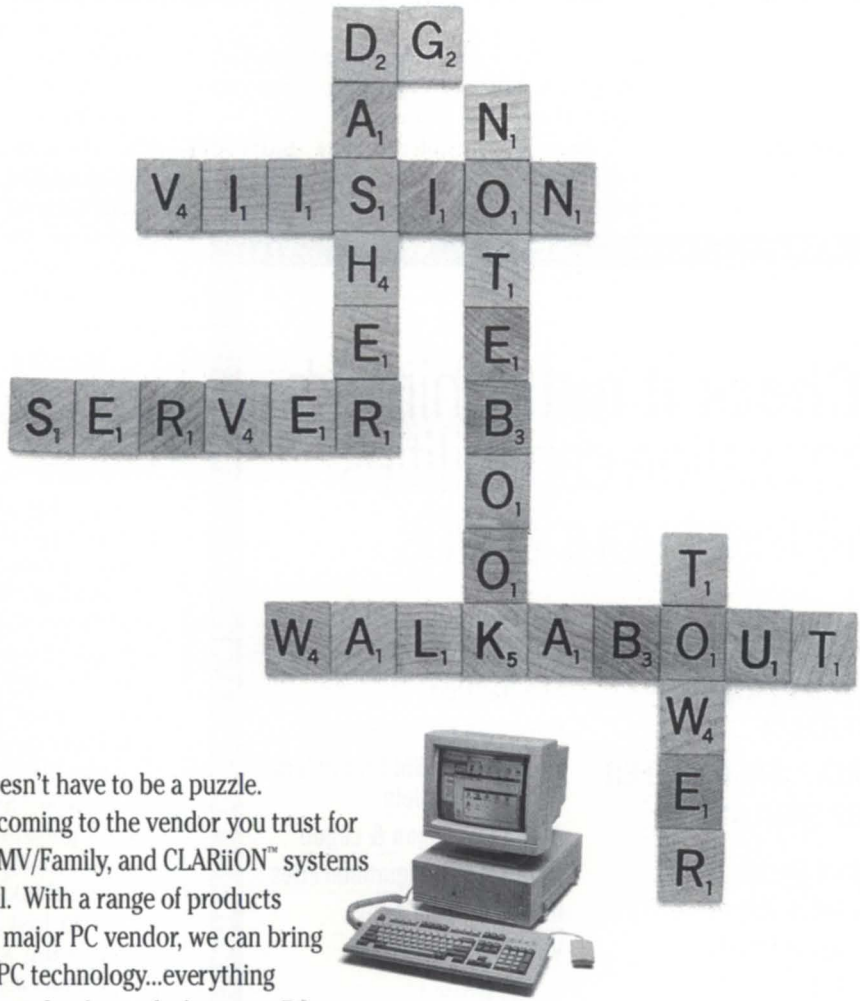
And while you are in Atlanta you can help us celebrate a series of anniversaries: 1993 marks Data General's 25th, NADGUG's 20th, and the SOURCE conference's 15th.

So mark your calendar now to attend. It's going to be *dynamite!* Δ

Calvin Durden is chairman of NADGUG's Audit Committee, and of the Conference Committee.

FOCUS
The Magazine of the North American Data General Users Group

Puzzled by the PC game? DG brings the pieces together.



Buying PCs doesn't have to be a puzzle. It's as easy as coming to the vendor you trust for your AViiON®, MV/Family, and CLARiiON™ systems — Data General. With a range of products that rivals any major PC vendor, we can bring you the latest PC technology...everything from portable notebooks, to desktops, to PC towers.

When you need PCs, score the winning points by building on your strong foundation with Data General. Our full line of PCs will be displayed at NADGUG '93. Stop by and see us!

Just call 1-800-DATA GEN for more information.

intel inside

MICROSOFT® WINDOWS™ READY-TO-RUN

Data General
Where the World is going!

Circle 2 on reader service card.

FARE_WELL:TILL_NEXT_TIME

by Greg Farman
Focus Publisher

A lot of people—myself included—have tried to imitate one aspect or another of the inimitable BJ persona. But at least for me it never quite came out right. The man is an original.

Before you get the idea that I'm writing a eulogy, let me explain. BJ (aka Brian Johnson) is the author of the "System Manager's Log," the longest-running column in *Focus*' 8-year history. Yep, he was there for issue #1 in September 1985, and he never missed an issue, even when the deadline found him consulting with European clients or carousing with biker buddies in Sturgis, South Dakota. He helped solve a lot of problems for MV system managers along the way, and made a lot of friends and even a few enemies in the process.

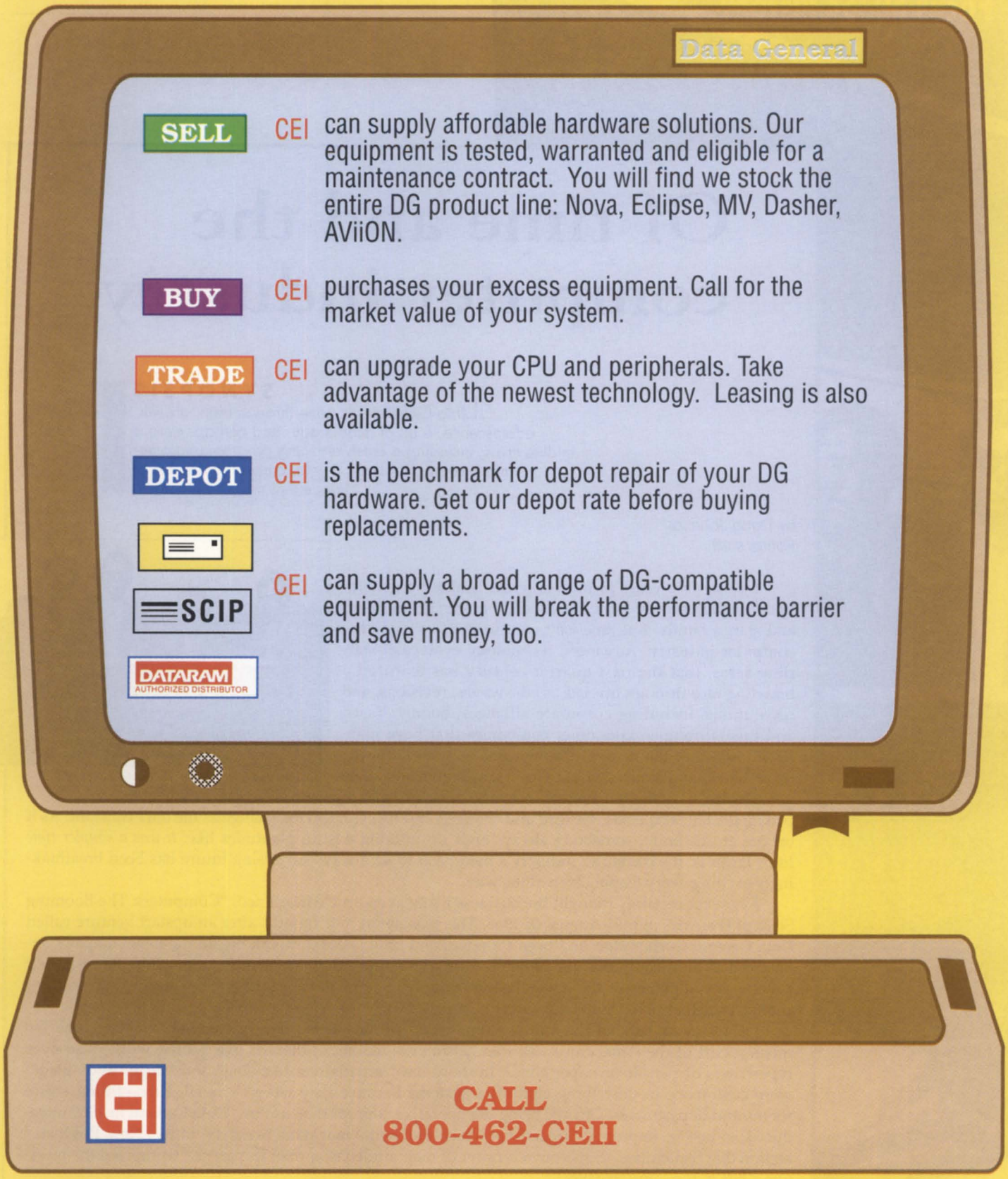
BJ called me a few weeks ago to say he'd been thinking things over, and he just didn't see how he could continue to write a once-a-month column. MVs and AOS/VS have proved remarkably stable. They are getting faster and less expensive with each model introduction, but otherwise there aren't many new developments. Nor are there the burning issues that enlivened many of his columns in the past. The natives just aren't as restless as they used to be, and that left him with less to write about.

I sadly agreed. The thing that makes the "System Manager's Log" such a pleasure to read is its *freshness*. You can always count on BJ to tell you what he thinks. He says he will continue to do just that—but not as often, because the MV world is giving him less to think *about* these days.

So what is BJ thinking about these days? His beloved Harley-Davidsons, of course. And, of all things, performance monitoring. He told me about the prototype he recently built: a real-time display of all the parameters that affect H-D performance. On a recent motorcycle trip into the California foothills he got so many "Where can I get one of those?" questions that he's contemplating going into full-scale production.

It's good to know that a whole new market stands to gain from BJ's advice. And I'm glad to say that MV users can still look forward to his strong opinions—just not as often.

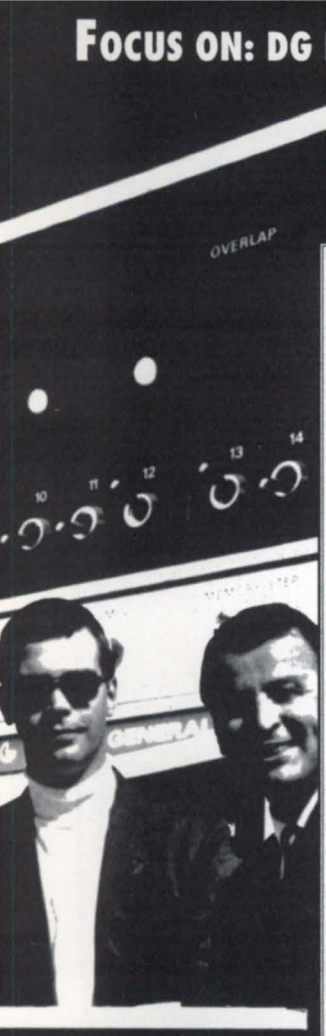
P.S.—If you really want to hear from BJ more often, give him something to think about by calling or leaving a BBS message. Δ



Computer Engineering International

2231 Star Court
P.O. Box 81755
Rochester, MI 48308
MI Phone: 313-853-0770
FAX: 313-853-0775

Circle 8 on reader service card.



Of time and the computer industry

SYNOPSIS

Data General has gone through birth, growth, adolescence, a bit of middle age, and perhaps even a midlife crisis, including a crash reducing program and other traumatic events. It's been a wild, careening ride full of change that never really stops.

by Doug Johnson
Focus staff

DG founders (L-R) Henry Burkhandt, Edson de Castro, Dick Sagge, Herb Richman

Twenty-five years in human terms is about a generation—enough time to grow up, choose a career, get married, and start a family. But time isn't measured that way in the computer industry. Advanced technology evolves at startling rates. Just the past quarter-century has featured a brawling race through myriad trends, waves, revisions, and revolutions, including corporate alliances, booms, busts, and breakthroughs. Like other companies that have managed to endure the turmoil (though not without suffering some blows and bruises), Data General's origins and early years seem like ancient history now. So much has happened; the landscape has changed and changed and *changed*, leaving change as the only constant. As if on cue at this point, somebody always ends up uttering wistful platitudes like: *It was a simpler time then*. Truth is, the computer industry's mad dash to an always surprising future has been breathtakingly exciting every leaping step of the way.

Newsweek certainly thought the business lively in an article headlined, "Computers: The Booming State of the Art," in mid-August of 1968. This was about four months after an upstart venture called Data General Corporation had been founded, and only about a month before the new company announced its first computer, the Nova. But tiny DG didn't rate any mention alongside the respected behemoth companies of the day—IBM, Burroughs, Sperry Rand, Honeywell, Control Data, and, of course, principal rival Digital Equipment Corp. (DEC).

Data General entered the fray at a time when, in *Newsweek*'s words, computers had become "so much a part of the American scene that getting excited about them is like getting worked up over typewriters or pencils or paper clips." In those days superlatives like "big" and "huge" and "giant" were customary in describing computer systems because they were . . . well, *huge*—filling entire rooms and departments. Change being the operative concept then as now, Data General in 1969 introduced its newer, improved Supernova, which *Electronics* magazine reviewed in its "New Products" section that September. "Supernova is order of magnitude faster than its parent," trumpeted the headline. "Fits in the same size box."

An article in *Time* appearing June 19, 1971, observed how completely computers were—all together now—*changing* American life, commented on price wars waged by the big companies, and gushed over breakthroughs of the time: "Last week Honeywell-G.E. introduced its Series 6000 line of fourth-generation models (price: up to $4,500,000), which can execute 1,000,000 instructions a second." For those of you keeping score in 1993, that would be $4.5 million per MIPS.

1,000...2,000...3,000...4,000...5,000...

In March 1971 Data General delivered its 1,000th computer; from there the pace would accelerate. By February 1973 there had been the 2,000th, 3,000th, 4,000th and 5,000th. Late that same year DG would be listed on the New York Stock Exchange.

An upstart company creates its own culture, and that gets noticed sometimes when writers seek to say something insightful. "The Long Hairs vs. The Stuffed Shirts," read the headline on one such article in the Jan. 15, 1976, issue of *Forbes*. The "Stuffed Shirts" were DEC, the "Long Hairs" were DG. These two fiercely competitive minicomputer companies exhibited markedly dissimilar styles.

"A number of Wall Street analysts are sure they know what DEC's trouble is," commented *Forbes*. "They say DEC—itself an 18-year-old minicomputer pioneer—has become a stuffy company top-heavy with management; the vice president of sales even has a glass and wood-paneled office and a built-in stereo system." But just a few miles away at DG's headquarters in Southboro, the contrast was striking:

"There are only two types of office furniture: cheap steel desks and cheap folding tables, the kind its computers are made on. Some of the top brass wear their hair hippie-long and work out of offices jammed with computer printout material."

Data General was seen as a company that had built its success through the ability make machines more cheaply than DEC, and then sell them at a lower price. "Data General keeps its manufacturing costs low by such devices as using plain-gray toggle switches on the face of its machines; DEC uses chrome-plated ones," noted *Forbes*. By the end of its first decade DG had opened manufacturing plants in foreign countries, announced its first Eclipse computer (the micro Nova) and the Dasher, opened its Westboro headquarters in Massachusetts and the Research Triangle Park lab in North Carolina, and shipped its 50,000th computer. It was nearing the $500 million mark in annual revenues as well. The MV/8000 had made its debut, along with the AOS/VS operating system.

"The market for minis soars out of sight," heralded *Business Week*'s issue of April 18, 1977. In third place behind DEC and Hewlett-Packard, Data General was in the middle of enjoying a bonafide boom. The boom included battles and price wars: "Early in March, Digital Equipment made a hurried announcement of a new, lower-priced machine to compete with a popular Data General model. Last week Data General leapfrogged the industry leader and introduced a new model of its own that undercut the price of the just-announced DEC machine by 20 percent or more." And so it went, dizzyingly onward and upward.

"Data General at 10: all's well" read the headline in an *Electronics* article dated Jan. 4, 1979, which praised 10-year-old DG's high profit margins and commented, "The management team is balanced, with a resulting balanced emphasis on engi-

25 Milestones

1968 - 1969

April 15, 1968 • Data General founded.
September 1968 • Nova computer announced.
March 1969 • DG moves to Southboro.
November 1969 • First public offering of common stock.

1970 - 1975

1970 • DG establishes sales operations in Europe.
March 1971 • 1,000th computer delivered.
March 1972 • Field Engineering Depot opened in Manhattan Beach, California.
April 1972 • 2,000th computer delivered.
July 1972 • 3,000th computer delivered.
October 1972 • Sunnyvale, California, plant opened.
December1972 • 4,000th computer delivered.
February 1973 • 5,000th computer delivered.
April 1973 • DG opened core memory assembly plant in Hong Kong.
December 1973 • DG listed on New York Stock Exchange.
April 1974 • 10,000th computer delivered.
1974 • Manufacturing plan opens in Thailand.
October 1974 • First Eclipse computer announced.
April 1975 • Westbrook, Maine, plant opened.

1976 - 1980

March 1976 • First micro Nova computer announced.
June 1976 • 1,000th Eclipse computer delivered.
1976 • Manufacturing plant opens in the Philippines.
July 1976 • Portsmouth, New Hampshire, plant opened.
August 1976 • First Dasher announced.
November 1976 • AOS operating system introduced.
January 1977 • Westboro headquarters opened.
May 1977 • First CS computer announced.
November 1977 • Research Triangle Park (North Carolina) lab opened.
June 1978 • 50,000th computer shipped.
August 1979 • Joint venture agreement reached with Nippon-Data General.
September 1979 • Austin, Texas, plant opened.
November 1979 • Xodiac and AZ-TEXT software announced.
February 1980 • Data Base Management System (DBMS) announced.
April 1980 • Transaction Processing Management System (TPMS) announced.
• Eclipse MV/8000 computer announced.
• AOS/VS operating system announced.

FOCUS ON: DG HISTORY

1981 - 1984

- February 1981 • Apex, North Carolina, plant opened.
- • Dasher G300 graphics terminal announced.
- May 1981 • 100,000th computer shipped.
- July 1981 • DG-Woodstock purchased.
- November 1981 • CEO introduced.
- February 1982 • DG-Milford opened.
- March 1982 • DG acquires majority ownership of Nippon-Data General.
- November 1982 • Eclipse MV/4000 computer announced.
- March 1983 • Eclipse MV/10000 computer, GW/4000
- • Expert Workstation and Dasher G500 graphics terminal announced.
- June 1983 • Ada programming language validated by Department of Defense.
- July 1983 • Desktop Generation announced
- • DG announces a Unix operating system environment for the Eclipse MV Family computers.
- January 1984 • Scientific Advisory Board is formed.
- February 1984 • DG signs agreement with PacTel Communications Systems, the first agreement through which a newly created regional telephone operating company will market a computer firm's integrated office system.
- March 1984 • CFO (Comprehensive Financial Operations) announced.
- May 1984 • DG purchases 132 acres of land in Durham, New Hampshire, for manufacturing and R&D purposes.
- • GW/8000, GW/10000, Graphics Workstations announced.
- June 1984 • DS/4000 and DS/4200 Distributed Systems workstations announced.
- • DG/UX native Unix announced.
- • TCP/IP announced.
- July 1984 • ADA validated for the second consecutive year.
- • Common LISP programming environment announced, allowing DG to enter the artificial intelligence market.
- September 1984 • Data General/One Personal System is announced.
- • DG becomes a $1 billion corporation.

1985 - 1988

- January 1985 • Eclipse MV/10000 SX processor announced.
- February 1985 • AOS/VS Decision link announced.
- • Eclipse MV/4000 DC announced.
- July 1985 • Dasher/One announced, the first in a series of Data General intelligent workstations.
- August 1985 • CEO Document Exchange Architecture (DXA) introduced. CEO DXA enables users to exchange documents, messages and other information over IBM DIStributed Office Support System (DISOSS) networks.
- September 1985 • European Development Lab opened in Cambridge, England.
- October 1985 • Dasher D555 Integrated Voice/Data Workstation, VMC/2 Voice Mail Controller and CEO Voice Software announced. Combination of the VMC/2 voice board and CEO Voice software provides the first voice mail product fully integrated in a business automation environment.

November 1985 • A series of Eclipse MV/20000 systems and an Eclipse MV/2000 DC are introduced, extending the performance of the Eclipse MV Family at both ends of the price/performance scale.
• DG/7700 series of engineering workstations unveiled.
• TEO Technical Electronic Office software announced. First software package to provide a wide variety of engineering and office automation functions while allowing users to communicate with others on the same distributed network.
• AOS/DVS Advanced Operating System/Virtual Storage announced.

December 1985 • DG acquires assets and technology of the Integrated Digital Network (IDN) operation, a subsidiary of United Technologies Corporation.

January 1986 • DG purchases minority interest in Dama Telecommunications.

February 1986 • 40,000-square-foot manufacturing facility in Chihuahua, Mexico, opens.
• XTA/SNA Backbone is announced, integrating Data General Xodiac networks with IBM's Systems Network Architecture.
• DG joins Corporation for Open Systems (COS) as a senior research member.

April 1986 • Office Automation lab established in Spain.

May 1986 • Data General/One Model 2 is introduced.

July 29, 1986 • Eclipse MV/7800 computer introduced.
• DG/DBUS is introduced.

Sept. 30, 1986 • DG/RDOS operating system introduced for the MV/3000 DC.

Dec. 9, 1986 • Eclipse MV/15000 family of computers is introduced.

May 5, 1987 • Data General/One Model 2T is introduced.

May 12, 1987 • Technology assets of EASINET Pty Ltd. of Australia are acquired.

June 16, 1987 • Eclipse MV/1500R (Rugged) and Eclipse MV/15000T (Tempest) series computers are introduced.

Aug. 11, 1987 • Eclipse MV/7800XP and Eclipse MV/7800 DCX computers are introduced.

Oct. 20, 1987 • AOS/VS UNIBOL language is introduced.

Jan. 12, 1988 • CEO Desktop Composer is introduced.
• CEO 3.00 is introduced.

Jan. 25, 1988 • Austin, Texas, facility is sold.

Feb. 2, 1988 • VSPAC is introduced.

April 18, 1988 • DG announces plans to develop a complete line of industry standard computer systems based on Reduced Instruction Set Computing (RISC) technology. The systems will incorporate the 88000 series of microprocessors from Motorola.

Aug. 22, 1988 • DG/UX revision 4.0 begins shipping to customers.

Oct. 3, 1988 • Eclipse MV/40000, Eclipse MV/40000 HA, and AOS/VS II are introduced.

Dec. 20, 1988 • 727 MB Winchester disk drive is introduced.

neering, manufacturing, marketing, and maintenance." The company was charging confidently into its second decade.

No one predicts completely what will happen in the computer industry, but a cursory look backward with sharp hindsight indicates that at times the pundits were seeing pieces of it clearly. "What's paving the way for greater use of computers?" asked *U.S. News & World Report* in its July 19, 1976 issue. "For one thing, the equipment is getting cheaper and easier to use every day. Size is shrinking, too, and that fact alone is opening up all sorts of new applications."

Cheaper processing power could mean an eventual "squeeze" for the minicomputer industry, *Business Week* forecasted in its issue of April 26, 1976. Mainframe manufacturers would have incentive to move downward, announcing their own minicomputers and competing more directly with companies such as DEC, Hewlett-Packard, and DG. Attacking from the low end would be the newer, cheaper, microprocessor-driven personal computers.

The "old days"

By 1980 Data General had been around long enough that there was even such a thing as "the old days."

"Data General—life in the fast lane," mused the headline of a retrospective piece appearing in *Forbes* the week of March 3, 1980. "The old days were only a decade ago. In 1968 [Herbert] Richman, President and CEO Edson de Castro, and three others started the third-largest minicomputer manufacturer in a vacated beauty parlor in a strip shopping center." Pictured with "The First Nova" on display at the Westboro headquarters was a smiling de Castro.

The minicomputer boom was an exciting thing to contemplate. "Welcome to the world of hypergrowth," commented *Forbes*. "It has been like this in the minicomputer industry since it was born in the early 1960s, and the end is not in sight." Rapidly developing technology knocked prices down some 15 to 20 percent each year, "and new uses and users come pouring out of the woodwork. Nobody really knows what the total demand for minicomputers is."

Data General was accelerating along: "a worldwide enterprise—13,700 employees and over $500 million sales—growing 30% to 35% a year." At 30 percent a year, DG would have swelled to $7 billion by 1990. But hypergrowth was not a thing that could continue forever.

In May 1981 Data General shipped its 100,000th computer. The company set out to conquer the office automation market with the announcement of CEO in November 1981. The story of the company's development of the acclaimed "Eagle" system was immortalized in a bestselling book, *The Soul of a New Machine*, by Tracy Kidder. DG's stock peaked at 87 in 1980, but slumped to 27 just two years later. Earnings had dipped, and the company faced "a fork in the road," according to *Forbes* in an October 11, 1982, article. DG's momentum was slowing, "and its hard-selling tactics were increasing-

ly out of step with a changed marketplace."

"Can Data General regain star status," asked *Forbes*, "be a standout company in the different data processing markets of the Eighties?"

Business Week, in its issue of June 29, 1981, announced "Data General's comeback attempt" with the introduction of a new family of desktop computers "that it hopes will help it regain momentum." Called the Enterprise 1000, this first entry into personal computers by Data General would, two years later, be recalled by the same publication (*Business Week*, Aug. 1, 1983) to have "failed badly." New desktop models from DG were needed, *Business Week* asserted, to regain the company's growth track, "particularly at the low-performance end of the market, where competitors with lesser-priced microcomputers were eating away at the company's staple product—the cabinet-size minicomputers selling for $30,000 to $140,000 that generated most of Data General's growth in the 1970s."

In its issue of March 7, 1983, *Business Week* reported on a management-overhauled Data General that was doing bold things. "In November [1982], Data General introduced the MV/4000 superminicomputer, an entry-level system reputed to have about double the performance of comparably priced systems from DEC." And now the company would unveil the MV/10000, "which will offer twice the performance of its current top-end machine, at almost the same price."

By September 1984 Data General the minicomputer company reached $1 billion in annual revenues. But personal computers were becoming the exciting growth market, with distributed networking somewhere out there in the near future, and that heralded *change*, of course. Huge layoffs were unheard of at one time in the computer industry, but those were coming, too, and not just at Data General.

In its Nov. 25, 1985 issue, *Business Week* asked, "Who's Breathing Down Whose Neck Now?", meaning that Data General, with the introduction of a new superminicomputer, would be coming on fast in its lifelong corporate battle with DEC. DG had laid off about 7 percent of its workforce in 1985, but the company was unveiling a supermini code-named "Viking" that would become the MV/20000. *Business Week* noted that the new machine was expected to be "20% faster and 30% less expensive than DEC's VAX 8600."

It's easy to look back now and say, "See there back in June 1984 when Data General announced DG/UX?" Then only about five years later the company would announce its new Aviion line of Unix-based workstations and servers, and Data General would charge away into open systems, and by 1992 Aviion revenues would eclipse the MV . . ." as though it would happen according to some pre-arranged plan.

But there is no such thing as predestination in the computer industry. In the space of about two decades, Data General had gone through birth, growth, adolescence, a bit of middle age, and perhaps even a midlife crisis, including a crash reducing program and other traumatic events. It had been a wild, careening ride full of change that never really stops. Data General the minicomputer company in the late 1980s and beyond would evolve toward becoming Data General the Unix workstation and server company. Call it a career change, a new chapter in a life story. Δ

Editor's Note: *To be continued in November.*

Tim Boyer

A debugger to fall in love with

SYNOPSIS

ICobol debuggers have in the past fallen short of potential, opines our columnist, but not anymore. Get ready for a soon-to-be-introduced tool called RAID from Wild Hare Computer Systems.

I suppose it's a result of rebelling against my early upbringing that I'm so entranced by a good debugger. Consider the way I learned to program. First, I used that IBM 26 or 29 to keypunch my programs. I handed the card deck to the High Priests in the operations room, who put it behind a bunch of other cards waiting to be processed. A half-hour or so later, the cards ran through the reader, generating an error report.

When I got my CS-40, I found that I could do as many as four compiles an hour! I vowed then and there to quit being the machine's precompiler. As a result, my initial programs tended to be error-ridden, with multiple misspellings. But who cared? The compiler flagged the errors much more quickly than I could, then I corrected them and recompiled.

I also became highly dependent on debuggers, something we didn't have with the punchcards. Or maybe we *did* have it, but no one told us. Rather than pouring over code listings, I'd much rather fire up a debugger and step through the code. One of the reasons I tend to write small ICobol programs is that in ICOS the debugger took a lot of overhead. The debug information had to fit into memory with the interpreter. If your programs were too big you couldn't debug them.

The ICobol debugger was a big improvement over ICOS, because the size constraints were gone. The debugger in ICobol 1.60 represented another giant step forward, adding a couple-dozen commands that I'm not sure how I ever did without. I *need* the DUMP command in ICDEBUG—no more guessing whether that data item contains spaces, or LOW-VALUES, or HIGH-VALUES. But the standard to which I hold debuggers is Borland, and ICobol debuggers have fallen woefully short.

Until now. The people at Wild Hare Computer Systems of Boulder, Colorado, have been kind enough to lend me a beta version of their new RAID debugger, and I think I've fallen in love. RAID, which stands for **R**educed **A**ggravation **I**nteractive **D**ebugger, is an integral part of Wild Hare's Axis compiler. Programs are compiled for debugging by typing the normal Axis compile line, along with the "/debug" switch. This causes two additional files to be created, which contain the actual debugging instructions.

Once a program has been cleaned up, there's no need to recompile—simply delete the additional files. Note that this means that you could compile and debug a program on the PC, move the source files to your MV or Aviion, or whatever, and the programs will run without a glitch. And I'll bet your MV doesn't compile at more than 8,000 lines per minute, like my 486/66 just did. But regardless of where you program, the RAID debugger is available on all the Axis platforms, from the PC to the MV, to VAX, to the Aviion, and object code is portable across the platforms.

About the only missing feature is an internal editor. Bruce Ray from Wild Hare has said that the problem is in deciding *which* editor emulation to implement (for reasons I can't fathom, he refuses even to *consider* SED), and stated that the next release of the debugger will allow shelling to DOS. From there, the user could pick his favorite editor. I suggested either minimal Wordstar-Turbo Pascal editor keys (^F moves a word right, ^A moves a word left, ^Y deletes the line, etc.), or a user-definable

Shop Our Million Dollar Data General Inventory And Get "Right-On-Time" Delivery.

• Million dollar inventory of systems, peripherals, parts and equipment • From Nova to MV -- you get "Right-On-Time" delivery • Highly skilled engineers • Valuable assistance, consulting, training and installation • Customer service reps with broad product knowledge • Problem solving advice • All backed by RAVE's 60-day warranty

• Sensible business solutions with significant bottom line value •

SELL • BUY
LEASE • RENT
Data General & Compatibles

We Also Sell Sun Microsystems

Sun Microsystems Computer Corporation
Authorized Refurbishment Reseller

Announcing -

RAVE FINANCIAL SERVICES, INC.

- Complete In-House Financing
- Custom Leasing Packages
- Long or Short Term Programs
- Rentals Available

Rave Computer Association, Inc.

RIDE THE WAVE — RAVE COMPUTER ASSOCIATION, INC. — SAVE WITH RAVE

36960 Metro Court
Sterling Heights, MI 48312
(313) 939-8230 / Fax: (313) 939-7431
International office
(516) 929-5000 / Fax: (516) 929-5007
1-800-966-7283

Circle 36 on reader service card.

editor that one could shell to automatically. But I ended up not needing one after all (see the end of this column).

RAID is invoked when either the initial program contains debugger information, the console-interrupt key is pressed, or an abnormal system condition (i.e., a bus trap) occurs. It starts out with two windows, the program window and the command window. The program window contains the program listing, and follows the execution of your particular program. This remedies what I feel to be the most glaring deficit of the current ICobol debugger: the need to "step and display" your way through a program. You'll normally look at the source code in normal ASCII fashion, but you can also look at the actual opcodes being executed. I haven't found a good use for this yet, but I will.

The command window is a section at the bottom that allows you to enter debugger commands, and provides a history of the most recent commands. Debugger commands can be entered in one of three ways: 1) using a mouse on the pulldown menus; 2) typing the command; or, in some cases, by 3) using function keys. Of the three methods, the mouse is by far the easiest; I'd recommend using this debugger without a mouse just like I'd recommend using Windows without a mouse. The input screen won't accept minimally unique commands (e.g., "s" and "step" will single-step the program, but not "st" or "ste"), and only a half-dozen function keys are active. But if you're used to a mouse, you'll like the menu layout. And the things you can do with it.

Take the "Run" menu, for instance. Like the ICobol debugger, you can set breakpoints and data watchpoints, and run to them (incidentally, Axis does the paragraph name breakpoints correctly—in ICobol, try using a paragraph name of, say, "123" and setting it as a breakpoint). You may also, however, choose to run to the next PERFORM statement, the end of the current PERFORM, the end of the current paragraph, the next CALL, the next I/O statement of any type, or the next I/O error. This eliminates the necessity of constantly setting and deleting breakpoints in order to move from one event to another.

I especially like the "run to I/O error" option. You don't need to set watchpoints on all of the file-status variables, or breakpoints in the DECLARATIVES. Just one mouse click, and you stop the next time a nonzero file status code is returned by any file. Combine this with the FILES window (see below), and it makes mode errors a breeze to diagnose. In the next version, this type of "exception reporting" will be expanded. You'll be able to set independent breakpoints to break on subscript error (an error that my programs run into frequently), size error, math error, I/O error, or EXCEPTION STATUS.

RAID uses the ACCEPT command to set new variable values, and the DISPLAY command to show the value of a

variable. Once again, these commands are enhanced.

You may choose to accept or display variables in decimal, hex, octal, BCD, or IEEE floating-point formats, as well as choose the number of bytes to display. The statement:

```
DISPLAY MY-VARIABLE: 6, H
```

will display your variable as 6 hex bytes.

In addition to these two windows. RAID allows you to pop up another dozen or so windows providing a wealth of information about the program execution. The PERFORMS windows, for instance, contain the iteration count, paragraph name at the start and stop point, and line number executing the perform. The FILES windows display the status, file type, open mode, and open file name of all the files in a program.

Ever write an update program that wasn't updating, and no matter how hard you looked for a bug you couldn't figure out why it wasn't updating, only to discover that you forgot to open the file? Yeah, me too. This window prevents that. The MEMORY window will display user variables, or areas of memory, in ASCII, hex, octal, or decimal. All windows, including the program and command windows, may be moved, opened, closed, or resized as necessary.

The "History" facility may be the most useful of all. These commands will allow you to trace back from a particular event, watching the program arrive to that point. There's an internal facility, which by default records the last 50 statements, a program history file, and a program history window. Using this facility, you'll never have to watch your program roll off into Never-Never Land, while scratching your head and asking yourself, "How the heck did I get *here*?"

To use History, all you need to do is turn the recording mode on and run. When the program stops, for whatever reason, selecting "Replay" from the History menu will open a history window and position you at the last statement executed. From there you can use the up arrow to walk back through the program execution. In the documentation I received, info on the History feature seems to be a bit sparse, but then again, this is a beta test. I hope it will be expanded before the actual product release.

There's only one feature missing from this debugger that I'd like to have, and I've just been told that it will be the next enhancement added: the ability to set a breakpoint depending on program values. I've seen it in the Aviion MXDB debugger, and it's an incredibly powerful tool. MXDB will allow you to write a statement like so:

```
breakpoint 50 if (CLOCK-NUMBER IS EQUAL 233)
```

I keep running into programs that fail only when, say, the customer number

Continued on page 39

David Novy

Nostalgia

SYNOPSIS

Our columnist offers a toast to the memories of bygone times in the computer industry (they weren't all that good, by the way), and sees a promising future for a reorganized Data General.

The theme for this issue of *Focus* is looking back, particularly at Data General and NADGUG. As it happens, Data General celebrates its 25th anniversary during 1993, and NADGUG celebrates its 20th anniversary the same year.

Since I've been a NADGUG member for more than 12 years, I was asked to write about the good old days. After thinking about the subject for a while, the idea which most often came to mind was survival. The good old days weren't really that good. The disk drives were unreliable. My expensive ($250,000), blazing-fast (all of 2 MIPS) minicomputer was obsolete and nearly worthless in less than three years. Backups were painfully slow and not very reliable.

So much for the good times. Here are some more memories:

- Carrying my 18-month-old son two miles on my back during a blinding snowstorm, so that I could go in to reboot a computer at work. Today that can be done easily via modem from home.

- Trying out a new 500 MB backup tape device. I thought I was safe because my collected backups were 9 deep. I made a successful recovery on my 9th tape. Two hundred people almost lost half a year's worth of work. Today there are hot-spare RAID systems and genuinely fast 5 GB tape backups systems, which perform those backups at more than 1 MB per second.

- Having to be at work through most of Thanksgiving and Christmas several years, because that was the only time available for formatting 277 MB removable-media disk packs. Yes, I am still married to the same woman and yes, someday she will be declared a saint for putting up with my work schedule.

- Bloody knuckles received while installing and removing 15-inch double Eurocards. And who needed the thrill of bungee jumping back then when you had the opportunity to install 100-pin connectors on MV backplanes? The new Aviion machines, thankfully, do not have such a thing.

'Forever' has arrived, and sooner than we thought

by David Novy
Special to Focus

During the last two years, the Data General marketing department has used the slogan, "MVs are forever." A recent announcement by Monarch Software has indicated that MVs may be forever, but very few people will care. Monarch is announcing third-party software that emulates the functionality of Infos and CEO.

The Infos replacement is called Open File Manager (OFM), and the CEO replacement is Open Electronic Office (OEO). In addition to OFM and OEO, a third-party, VS Cobol-compatible runtime system is available. Monarch can justify its claims because the project leaders and company officers are former Infos developers. What made the project feasible is that Monarch found a way that was both efficient and practical to break the tight coupling between Infos and the MV ring architecture. The people at Monarch were aided in their work by another third-party software developer, which agreed to develop a VS Cobol emulator. It is apparent that the reason most people still purchase MVs is to support Infos and VS Cobol applications.

The new software should allow users to migrate these applications to the Aviion and take advantage of the Aviion's price and performance benefits compared to the MV architecture. I am sure that there are several people in DG marketing who are not very pleased by the Monarch announcement. However, any marketing person who does his or her job well knows that MVs are not forever. Eventually, all customers will have to make the decision of when and how they are going to migrate their Infos applications to another platform, probably either a Unix or Windows NT-based system. Having a version of Infos that runs on an Aviion helps ensure that former Infos and VS Cobol users remain DG customers. And as loyal Infos users know, what is important is that Infos is forever. There are applications where Infos can compete very effectively against any major relational data base on the market today.

One of the areas where the new Monarch Infos replacement shines is in the area of user connectivity. With most relational data bases on the market today, a user count of 100 is exceptional. Bill Cole of Monarch Software states that the OFM software should be able to support user counts in excess of 1,000 on a powerful Aviion. Best of luck to Moncarch on their endeavors to replace Infos. Δ

On the positive side there was the satisfaction of a job well done, and the people at work who helped me along the way. The computers my colleagues and I supported helped creative people accomplish tasks that only a few years earlier would have been impossible.

I also remember the fun and camraderie over the years at NADGUG conferences. Hundreds of people gathered to share ideas and see the newest DG products. Over time, there's no doubt that if you want to see the finest in new computer hardware and meet interesting and fun people, the NADGUG conference was and is the place to do it.

From 1987 through 1991, there were the painful memories of Data General's corporate reorganization. The company reduced its workforce by 50 percent in order to survive. The phrase, "Someone was RIFed," came into being. RIF stands for **R**eduction **I**n **F**orce. Data General cut back its staff to the point that you might have been left wondering if anyone would be there to answer the phone. And then three months after the big RIF, there was another RIF. Finally, costs and income came into balance and it now appears that the company will not only survive but become reasonably successful.

In the late 1980s Data General's competitors used to claim mockingly that the reasons for DG's problems were poor products and poor marketing. Data General's problems were *not* caused by poor products, but rather by a fundamental change in market demands. The fact that DG was not a market leader in either the minicomputer or mainframe market at the time—and that the company already had a somewhat lean organization—probably was its salvation. Market leaders like IBM and Digital used their corporate reputations to continue selling outmoded equipment for a while, but doing so delayed their development of competitive client-server systems. And because Digital and IBM are so much bigger than DG, the degree to which they must reduce in size now to achieve profitability will prove to be that much more painful. I certainly do not believe that Digital and IBM have seen the last of major layoffs.

So this year at NADGUG 93 I will drink a toast to the memories of bygone times. I will remember that I worked in the computer field during a time when hardware failures evolved to become the exception rather than the rule. Now I look forward to trying to determine if it's possible to get a computer network of PCs, Macintoshes, and Unix workstations to work together. Who needs the thrill of bumgee jumping now when you've got MS-DOS 6.0 with Double Space and Memmaker? Δ

David Novy is a technical computer specialist at 3M in St. Paul, Minnesota. He is past chairman of the AOS/VS special interest group, and current chairman of NADGUG's SIG/UX.

Legacy for the future

SYNOPSIS

ICobol is a mature (read "legacy" or "old"), semi-proprietary Cobol language created to address multi-user, data-entry applications in the 1970s. It performed its initial job admirably in the 70s and 80s, but needs to adapt to future user requirements if it is to remain an effective application language.

by Bruce K. Ray
Special to Focus

ICobol was ahead of its time. Data General is now a quarter-century old, and ICobol is only slightly younger, but it remains the language of choice among DG commercial VARs (value-added resellers) and end users. It was designed in the mid-1970s to address a blossoming commercial market by a computer company previously intent on being an "iron shipper." In the late 60s and early 70s, Data General prided itself as a builder of fast hardware with low prices, and catered to the knowledgeable industrial and scientific user. DG initially targeted business VARs and end users that matched the following profile:

- Small, multi-user business system
- 1 to 16 terminals
- Time-shared central processor
- Wide range of hardware configurations
- Business language
- Focused business, commercial marketing.

Large competitors at this time were selling solutions in the range of $100 K to $2,000 K, since most were based on the dominant mainframe hardware and software designs. However, DG's progenitor and subsequent archrival, Digital Equipment Corporation (DEC), was increasingly successful selling solutions in the $50 K to $100 K range based on its PDP-11 hardware and DIBOL or BASIC language.

This market required a major attitude shift within Data General. Success demanded a solution presented in commercial terms, even if the hardware was sold elsewhere on its technical merits. A business-oriented language would be required for application programming, as would a simple, non-technical approach to system management. Equally important would be a business or commercially oriented sales and marketing strategy, even if only glued

over successful "iron-shipping" strategies. Data General had several internal development projects addressing these goals, with the "Easy" system and Interactive Cobol system being the most politically prominent. Each had its own technical strengths, agenda, and political profile (each providing enough interesting material for separate articles!), but in the end, ICobol was the final winner. Interactive Cobol rev 0 was originally sold in the mid-1970s as a separate language that ran under RDOS rev 5. It encompassed many technical achievements while giving DG a sales solution for the commercial market. Design considerations included:

- A commercially acceptable, Cobol-based programming language
- Interactive, terminal-oriented language extensions for technical and marketing advantages
- Pseudo-code program implementation
- Use of standard operating system for file handling
- Simple, independent ISAM file handler.

The ANSI '74 Cobol low-level standard was used as the language base, although ANSI compliance was not provided. But the real user appeal was the new "SCREEN SECTION" extensions created specifically to make programming interactive, terminal-oriented applications easier than on other systems.

Then-current hardware technology dictated a compromise that accidentally gave users benefits decades later. The major technical decision to convert Cobol statements to an intermediate "pseudo-code" was based primarily on memory size restrictions that are imposed by the computer's hardware, especially if the hardware was not designed to handle commercial, character-oriented, low-level data. However, if the Cobol compiler produces object code designed to run on an "ideal" computer with character-oriented, machine-level instructions, the number of machine-level instructions needed to execute a Cobol statement is reduced.

For example, a Cobol statement

may require 20 to 100 Nova machine-level instructions to execute, but may require only 2 or 3 machine-level instructions of a cleverly designed "ideal" machine. Since the Nova is a word-oriented rather than byte-oriented machine, this translates to 40 to 200 bytes for a typical Cobol statement. An "ideal" machine may be byte-oriented, and may require only 6 to 10 bytes to perform the same operation.

This led to the design decision to implement the system as a pseudo-code interpreter using a combined compiler/runtime. The compiler performs the task of converting Cobol source code to an efficient pseudo-code, which the runtime system then "interprets" when the resulting application is run. This also led to an unexpected long-term benefit: application portability

ICobol percolated through 4 major releases from the mid 1970s to the early 80s. DG marketing transformed RDOS into a "commercially oriented" operating system (ICOS) with only cosmetic changes. The "new" ICOS and the ICobol compiler/runtime were packaged in specific hardware configurations and sold as complete solutions. To simplify things for DG or the customer, specific configurations were given specific marketing numbers (i.e., CS/40, CS/50, CS/100, CS/200, etc.).

ICOS releases 4.2 through 4.5 were workhorse systems, and were available for both Nova and Eclipse hardware. A marketing decision was made in the early 1980s to "consolidate" the expanding range of hardware/software options to only the Eclipse systems, effectively making Nova ICobol systems obsolete. The "new" Eclipse ICobol releases were renumbered starting at revision 1.0, and were the start of the 1.x series that continues through to this day.

The early 1980s saw only minor incremental modifications to the product, mostly designed to maintain compatibility with Data General operating systems and marketing positions, or to enhance performance. No real language changes or enhancements were added from ICOS rev 4 through ICobol 1.0, 1.1, and 1.2. The next user benefits came with Cobol subroutine support and variable position "DISPLAY" in the "SCREEN SECTION" years later in revision 1.3. ICobol 1.5 added the ca-

pability to handle large ISAM data files that previous versions would choke on, but no other major enhancements.

The latest official DG ICobol release, 1.71, has added new user benefits in the form of the "STRING", "UNSTRING", and "SEARCH" verbs, enhancements to the "START" and "PERFORM" verbs, and "END-xxx" constructs. However, it was initially delivered first on Data General's newer Aviion platform, much later for AOS/VS, and is not available for other DG platforms or operating systems.

But user demands for improved development tools and capabilities are expanding more rapidly than the DG ICobol standard. Relatively minor enhancements added over the years still do not correct many current deficiencies or design limitations, including:

- Minisam single-thread file handler bottlenecks, especially in large systems.
- Compromised or non-existent file access and recording locking in client-server and peer-to-peer environments. This means that networks cannot be effectively supported.
- Expanded program and data size requirements above and beyond the 64 KB/64 KB standard.
- SQL use and RDBMS access from Cobol programs.

Third-party solutions have therefore been required to address these, and many other, immediate customer needs.

Current ICobol extensions

ICobol in its official DG form is showing its age. Even though Data General has added parts of ANSI '85 Cobol language constructs to its 1.7 ICobol release, more fundamental issues still remain ignored by the official language. Therefore, independent ICobol suppliers have added items to the DG standard to meet user requests:

Program and data size. The original 64 KB program and 64 KB data size have become limiting factors as applications become more sophisticated and the amounts of data manipulated increase dramatically. We now provide 2 GB support rather than the original 64 KB, accommodating even the largest program data requirement while maintaining compatibility with standard DG ICobol systems.

EXTERNAL items. Customers have wanted the capability to declare data files and program data as EXTERNAL, so that any program within a program unit may access common data. This greatly simplifies program design, provides compatibility with non-ICobol systems, and increases performance in many applications.

ISAM capabilities. Increasingly sophisticated data requirements and faster hardware have resulted in the number of keys per record increasing to 16 from the DG standard of 5. Any key may contain duplicates, including the primary key if so desired. All keys may be up to 255 bytes in length.

Windows. VARs and end users alike are expecting even mature applications to have the look and feel of the latest PC program. This adds new

life to old applications, decreases development and maintenance costs, and generally leverages past program investment.

Hotkeys. Like windows, the general concept of "hotkeys" is becoming expected in today's applications. Also like windows, these hotkey extensions to ICobol may be used without touching a line of code. Many hotkey options have been designed for Cobol optimization, and may be used with even old ICOS programs without modification.

Language extensions. Source language extensions have been added only if they directly improve the application developer's capabilities or provide immediate user benefits. We have added ANSI '85-compliant features such as "SORT", "SEARCH", "INSPECT", "STRING", and "UNSTRING" at the source level since they dramatically increase developer capabilities.

4GL access. Compatibility with third-party DBMS and 4GL (fourth-generation language) products can dramatically expand current application capabilities. For example, Oracle data base access through SQL may now be accomplished directly from within an ICobol program using Wild Hare's *Axis* and *Choice!* products.

Networking. More and more environments involve sharing data across diverse systems and distances. Our file handler has been designed around peer-to-peer technology from the very beginning, allowing us to work with client-server and peer-to-peer systems without modification. We also designed all of our systems to work with homogenous or heterogeneous networks at the highest level, eliminating dependencies on a specific hardware or software technology.

Client/server. As data repositories become more popular with enterprisewide applications, users' demands for flexibility and support have increased. All Wild Hare file handling is done with system-independent, peer-to-peer, multi-threaded technology.

Debugger. Since testing and maintaining application programs consumes increased resources, better tools bring direct results to everyone's bottom line. The new "RAID" ICobol debugger combines menu and window technology with a simple, flexible interface.

ICobol: Caterpillar to butterfly

ICobol has successfully provided commercial solutions for more than one and a half decades so far. But can it adapt to the increasing demands expected of future applications? Users are demanding more capabilities from applications while wanting vendor, operating system, language, and data independence. We can see the major areas need to be enhanced if ICobol is to meet these future needs:

User interface. Wild Hare has brought windowing and hotkey technology to ICobol programs using object-based techniques. Future products will build upon this base to provide easier user interface prototyping

Continued on page 39

Attention NADGUG Members
Notice of By-Laws Amendment

AMENDMENT I

Whereas the Board of Governors wishes to increase the representation and participation of the membership in the governance of NADGUG, it is hereby resolved that the By-Laws for the North American Data General Users Group, Inc. (NADGUG), ARTICLE IV, be amended as follows:

- Move Section B, sub-section 1 after sub-section 2 and renumber sub-section 1 and 2.
- Insert "the Governors-at-Large," after "the Board of Directors," in Section B, sub-section 1.

Insert a new sub-section, before Section B, sub-section 3 and renumber sub-section 3 as sub-section 4.

- "3. There shall be six (6) Governors-at-Large elected for two-year terms." and renumber.

Insert a new sub-section at the end of Section C.

- "g. Governors-at-Large — The Governors-at-Large will be responsible for representing the interests of all members of NADGUG."
- Insert "and for the election of Governors-at-Large" after "NADGUG Board of Directors" in Section E, sub-section 1.
- Insert "and Governors-at-Large" after "The newly elected officers" in Section E, sub-section 3.

Insert a new sub-section at the end of Section G.

- "3. The Governors-at-Large shall be elected in overlapping terms."

It is further resolved that the 1993 elections include the election of:

- Three (3) Governors-at-Large for a one year term; and
- Three (3) Governors-at-Large for a two-year term.

AMENDMENT II

Whereas the Board of Governors wishes to improve the efficiency of the operation of NADGUG, it is hereby resolved that ARTICLE VII, Section E, sub-section 2 be amended by substituting "one (1) month" for "three (3) months."

If there are any questions regarding these amendments to the By-Laws, contact Brad Friedlander, By-Laws Committee Chairman, or Jan Grossman, President.

Notice of Withdrawal of Recognition of Regional and Special Interest Groups

The Board of Governors has elected to withdraw recognition of the following Regional and Special Interest Groups for violation of ARTICLE VII, Section C of the NADGUG By-Laws:

RIGs

• Alabama • Northern California Frequent Users of Data General Equipment (No Cal FUDGE) • Los Angeles End-Users of Data General Equipment (LA EDGE) • San Diego Data General Users Group • Connecticut • Southeastern New England • Central New York Data General Users Group • Metropolitan New York Data General Users Group • Ohio • Smoky Mountain Users of Data General Equipment (SMUDGE) • Southeast Area (SEARIG) • Pittsburgh Area Users Group • Tri-State Area Data General Users Group • Dallas Area Users Group • Houston Area Data General Users Group (HADGUG) • Heart of Texas Data General Users Group • Potomac Users of Data General Equipment (PUDGE) • Quebec Area Data General Users Group

SIGs

• Federal SIG • Music & Recording Industry SIG • SMBasic

This withdrawal of recognition will take effect at the close of the General Business Meeting unless it is overruled by the membership because the RIG or SIG has presented proof of conformity with ARTICLE VII, Section C of the NADGUG By-Laws.

General Business Meeting Announcement

The next General Business Meeting of the North American Data General Users Group will be held in conjunction with the NADGUG annual conference in Atlanta, October 26, 1993 at 8:15 a.m. All members in good standing are eligible to vote on items brought before the meeting, which will include the above amendments to the By-Laws and the election of officers.

In accordance with NADGUG's By-Laws, any member in good standing may, by written proxy, authorize any other member to vote in their behalf. If you do not plan to attend, the Board of Directors urges you to exercise your right to vote by contacting the Recording Secretary no later than October 12, 1993, at the following address: Thomas Bounds, McCarty Farms, Inc., P.O. Box 366, Magee, MS 39111.

Process and network management

SYNOPSIS

The author continues a series of articles explaining the performance monitor utilities available for solving some of the most common bottlenecks that affect DG/UX process and network management performance in a multi-user server environment.

by Thomas E. Soukup
Special to Focus

In the August and September issues of *Focus* we looked at Aviion performance tuning using the Unix system activity reporter (*sar*). In this issue Unix process and network management will be discussed. The first section of this article explains Unix process management. Both UX/RPM and AV SysScope include process management statistics screens.

Sar does not report any process statistics; however, you can use the Unix utility *ps*. For additional information on Unix process management utilities, refer to the online manual pages on *ps*, *acctcom*, and *who*. The second section of this article explains Unix network management. Both UX/RPM and AV SysScope include network management statistics screens.

Sar does not report any network statistics, either; however, you can use the Unix utility *nfsstat*. For additional information on Unix network management utilities, refer to the online manual pages on *nfsstat*, *nfsd*, *statd*, *rpcinfo*, and *netstat*. The following sections will describe the significant data items displayed by *ps*, the Unix process status monitoring utility, and *nfsstat*, the Unix network file systems monitoring utility. For each option the following columns and descriptions are included: the significant data item that *ps* or *nfsstat* reports; the significant data item that the real-time performance monitor (UX/RPM) displays on its screen; the significant data item that the AV SysScope performance monitor displays on its screen; a brief explanation of each data item; some general performance guidelines; and additional features included in UX/RPM and AV SysScope not reported by *ps* or *nfsstat* or other Unix utilities.

Process management

A large percentage of CPU time spent in user code "%usr" (*sar -u*) can indicate "runaway" applications, excessive looping, extremely long (but correct) code paths, or it may just be how the code operates. In a common multi-user server environment, *%usr* should range from 40 to 80 percent, *%sys* from 10 to 40 percent, *%idle* from 0 to 40 percent of the total CPU time. Many times, ported and new applications need to be profiled to ensure they work efficiently. For example, if an application is written in C, replacing the "getchar()" C call with the "getc()" C call can affect overall system performance.

Record locking may be the number-one potential performance problem that affects the performance of applications. In a multi-user server environment, a rule of thumb is to use locks only when necessary. If you must use a lock, lock only that record being updated or written for as short a period of time as possible.

Other potential performance problems may be uncovered by closely monitoring your overall system usage patterns. Prime-time usage should be given to those jobs that are most important. By default, DG/UX attempts to schedule jobs giving online terminal users more system resources than to non-interactive batch and compute bound jobs. Long-running, resource-intensive batch jobs should be started during off-hours. If batch jobs need to be run during prime time, you should start them with a lower priority. For example, use the Unix command "nice -4 Process_ID" to start the job at four priority levels below interactive jobs. To lower the priority of a batch job already running, use the Unix command "renice 4 Process_ID" to lower its priority by four levels. For more information on changing a process' priority, refer to the online manual pages on *nice* and *renice*.

The Unix command *ps*, process status, reports information about processes, giving a snapshot of what is going on in the system. The *ps* command can be used to identify those processes that are using excessive resources. To identify those processes, use the following set of Unix commands outline in Figure 1 (page 30).

The output of the *diff* command will print what processes have accumulated CPU time. If a single process accounts for most of the CPU time, it may be a runaway process. You should check the owner of the process, and if it is a runaway process, as superuser use the Unix command "kill -9 Process_ID" command to halt the process. For relational data base management system users, you may first need to disconnect the user from the data base before using the Unix kill command. Failure to disconnect the user from the data base

may cause the data base to retain record locks for that user.

Process statistics

Process information is a bit different from other performance statistics. The main difference is that processes are extremely dynamic. Processes are constantly being created and destroyed under normal system usage. The number of processes in existence is variable with time.

The Unix command *ps*, report process status, prints certain information about active processes. The output of the command is controlled by the arguments specified. UX/RPM provides users with two process screens that differ only in the information displayed. Unlike the Unix command *ps*, UX/RPM continuously examines the processes and displays information on a tunable cycle.

The AV SysScope product contains the *sscope-ps* command to monitor process performance. This process performance monitor displays a window of performance statistics similar to the Unix command *ps*.

Guidelines for process statistics

Monitoring process statistics will give you an understanding of those processes that are consuming large amounts of CPU. Many times, a custom application consuming large amounts of CPU is actually a runaway process caught in an infinite loop. You must use the Unix kill command to stop that process.

You can also use process statistics to determine if processes are waiting on a specific resource. You may need to increase a kernel parameter or load balance disks to minimize the time a process spends waiting for a specific resource.

Features of UX/RPM and AV *sscope-ps* for disk activity

Both UX/RPM and *sscope-ps* UX/RPM continuously examine processes and display information on a tunable cycle. AV *sscope-ps* reports the total CPU time, but in addition reports "SYS_TIME," amount of time that the kernel was running on behalf of the process and "USER_TIME," amount of time the process was running in user

Figure 1: Identifying resource-intensive or "runaway" processes

Unix Command	Function
ps -ef > /tmp/ps.1	Takes a snapshot and saves it to the file "/tmp/ps.1."
sleep 20	Wait about 20 seconds.
ps -ef > /tmp/ps.2	Takes another snapshot and saves it to the file "/tmp/ps.2."
diff /tmp/ps.1 /tmp/ps.2	List the differences between the two snapshots.

space. Another feature of AV *sscope-ps* is the reporting of the value "SWAP, the amount of anonymous memory (swap space) reserved for use by the process, whether actually used or not.

Network management

The Unix utility *netstat* shows the status of all network parameters. The *netstat* command symbolically displays the contents of various network-related data structures: the state of all sockets; the state of interfaces that have been auto-configured; network addresses as numbers; per-protocol statistics; the routing tables; the local and remote addresses; send and receive queue sizes (in bytes), protocol, and (optionally) the internal state of the protocol for active sockets. For additional information on the *netstat* Unix utility, refer to the online manual pages for *netstat*, route, hosts, networks, protocols, and services.

You can use *netstat* to investigate network bottlenecks and other performance issues associated with the network. Record locking over the NFS network is extremely resource-intensive. Most relational data base management systems perform network locking for you. However, if you plan to write applications that use record locking over an NFS network, you need to develop and test locking algorithms that do not adversely affect overall network performance.

We will discuss the Unix utility *nfsstat* at length. *Nsstat* displays statistical information about the NFS (network file system) and RPC (remote procedure call) interfaces to the kernel. The *nfsstat* command symbolically displays the contents of various NFS and RPC data structures: the client NFS and RPC information and the server NFS and RPC information. For additional information on the *nfsstat* utility, refer to the online manual pages for *nfsd*, *statd*, and *rpcinfo*.

Remote procedure calls statistics

Remote procedure calls (RPC) is a name given to a layer of services that distribute computing resources across a network. RPC consists of clients and servers. An RPC client intercepts a ser-

Continued on page 32

Figure 2: Process statistics

ps (l) item	UX/RPM item	scope-ps item
F	f	F

The process flags associated with the process (octal and additive): (1) Process is being traced, (2) Process if bound to a virtual processor, or (4) Process is not bound to a virtual processor.

ps (l) item	UX/RPM item	sscope-ps item
S	s	S

The state of the process: (-) Non-existent; (S) Sleeping; (W) Waiting, (R) Running; (I) Intermediate; (Z) Terminated; and (T) Stopped.

ps (f,l) item	UX/RPM item	sscope-ps item
UID	user	NAME

The user name for the user ID of the process, if available from the password file. Otherwise, the user ID.

ps (all) item	UX/RPM item	sscope-ps item
PID	pid	PID

The process ID.

ps (f,l) item	UX/RPM item	sscope-ps item
PPID	None	PPID

The process ID of the parent process.

ps (all) item	UX/RPM item	sscope-ps item
PGID	None	PGID

The group ID of the process.

ps (all) item	UX/RPM item	sscope-ps item
SID	None	None

The session ID of the process.

ps (f,l) item	UX/RPM item	sscope-ps item
C	i	C

The processor utilization, represented by an integer from 0 to 7. The number reflects a process' relative interactivity. A process with 7 is highly interactive. A process with 0 is not considered interactive, but uses mostly CPU resources. You cannot control this value; it is produced dynamically by the kernel.

ps (f,l) item	UX/RPM item	sscope-ps item
CLS	None	CLS

Scheduling class. Classes may be TS (time sharing), FF (FIFO), or (RR) round robin.

ps (l) item	UX/RPM item	sscope-ps item
PRI	pri	PRI

The priority of the process; higher numbers mean lower priority.

ps (l) item	UX/RPM item	sscope-ps item
NI	nice	NI

Nice value; used in priority computation.

ps item	UX/RPM item	sscope-ps item
ADDR	None	None

The memory address of the process.

ps (l) item	UX/RPM item	sscope-ps item
SZ	size	SIZE

The size in pages of the resident memory image of the process including shared and unshared segments. A page is 4096 bytes.

ps (l) item	UX/RPM item	sscope-ps item
WCHAN	None	None

The event for which the process is waiting or sleeping; if blank, the process is running.

ps (f) item	UX/RPM item	sscope-ps item
STIME	None	START_DATE/TIME

Start time of the process in hours:minutes:seconds.

ps (all) item	UX/RPM item	sscope-ps item
TTY	tty	TTY

The controlling terminal for the process.

ps (all) item	UX/RPM item	sscope-ps item
TIME	cpu	TTIME

The cumulative execution time for the process, in minutes:seconds.

ps (all) item	UX/RPM item	sscope-ps item
CMD	cmd	COMMAND

The command name; and the full command name and its arguments (with ps -f).

ps (all) item	UX/RPM item	sscope-ps item
TTY	PID	TTY

The controlling terminal for the process.

Figure 3: RPC Statistics for NFS

nfsstat -r item	UX/RPM item	SysScope item
calls	server calls	server calls/sec

Calls is the number of RPC calls received by the NFS RPC server per second.

nfsstat -r item	UX/RPM item	SysScope item
badcalls	server bad calls	server bad calls/sec

Bad calls is the number of RPC calls rejected by the NFS RPC server per second.

nfsstat -r item	UX/RPM item	SysScope item
nullrecv	server null received	server null received/sec

Null received is the number of null RPC packets received by the NFS RPC server per second.

nfsstat -r item	UX/RPM item	SysScope item
badlen	server bad lengths	server bad lengths/sec

Bad lengths is the number of times that an RPC packet with an invalid length was received by the NFS RPC server per second.

nfsstat -r item	UX/RPM item	SysScope item
xdrcall	server bad headers	server bad Xdr calls/sec

Bad headers is the number of times that an RPC packet with a malformed header was received by the NFS RPC server per second.

nfsstat -r item	UX/RPM item	SysScope item
calls	client calls	client calls/sec

Calls is the number of RPC calls made by the NFS client per second.

nfsstat -r item	UX/RPM item	SysScope item
badcalls	client bad calls	client bad calls/sec

Bad calls is the number of RPC calls made by NFS clients that were rejected by the NFS RPC server per second.

nfsstat -r item	UX/RPM item	SysScope item
retrans	Client Retransmissions	Client Retransmissions /Sec

Retransmissions is the number of times that an NFS RPC call had to be retransmitted by the NFS RPC client per second.

nfsstat -r item	UX/RPM item	SysScope item
badxid	client bad XID	client bad XID/sec

Bad XID is the number of times that a reply to the client NFS RPC call did not match what the client sent per second.

nfsstat -r item	UX/RPM item	SysScope item
timeouts	client timeouts	client timeouts/sec

Timeouts is the number of times a reply to a call is not received within a required time limit per second.

nfsstat -r item	UX/RPM item	SysScope item
wait	client waits	client waits/sec

Waits is the number of times that a reply to a call was not received within the period allowed by the NFS RPC time-out value per second.

nfsstat -r item	UX/RPM item	SysScope item
newcred	client new cred	client new cred/sec

New cred is the number of times that the NFS client had to refetch authentication per second.

Figure 4: NFS Server Statistics

nfsstat -n item	UX/RPM item	SysScope item
calls	server calls	server calls/sec

Calls is the number of NFS calls received by the NFS server per second.

nfsstat -n item	UX/RPM item	SysScope item
badcalls	server bad calls	server bad calls/sec

Bad calls is the number of NFS calls rejected by the NFS server per second.

nfsstat -n item	UX/RPM item	SysScope item
create	none	server create calls/sec

Create is the number of NFS_CREATE element calls.

nfsstat -n item	UX/RPM item	SysScope item
getattr	none	server get attr calls/sec

Get Attr is the number of NFS_GET_ATTRIBUTE element calls.

nfsstat -n item	UX/RPM item	SysScope item
link	none	server link calls/sec

Link is the number of NFS_LINK element calls.

nfsstat -n item	UX/RPM item	SysScope item
lookup	none	server lookup calls/sec

Lookup is the number of NFS_LOOKUP element calls.

nfsstat -n item	UX/RPM item	SysScope item
mkdir	none	server mkdir calls/sec

Mkdir is the number of NFS_MKDIR element calls.

nfsstat -n item	UX/RPM item	SysScope item
read	none	server read calls/sec

Read is the number of NFS_READ element calls.

nfsstat -n item	UX/RPM item	SysScope item
readdir	none	server readdir calls/sec

Readdir is the number of NFS_READDIR element calls.

nfsstat -n item	UX/RPM item	SysScope item
readlink	none	server readlinkcalls/sec

Readlink is the number of NFS_READLINK element calls.

nfsstat -n item	UX/RPM item	SysScope item
remove	none	server remove calls/sec

Remove is the number of NFS_REMOVE element calls.

nfsstat -n item	UX/RPM item	SysScope item
rename	none	server rename calls/sec

Rename is the number of NFS_RENAME element calls.

nfsstat -n item	UX/RPM item	SysScope item
rmdir	none	server rmdir calls/sec

Rmdir is the number of NFS_RMDIR element calls.

nfsstat -n item	UX/RPM item	SysScope item
root	none	server root calls/sec

Root is the number of NFS_ROOT element calls.

nfsstat -n item	UX/RPM item	SysScope item
setattr	none	server set attr calls/sec

Set attr is the number of NFS_SET_ATTRIBUTES element calls.

nfsstat -n item	UX/RPM item	SysScope item
statfs	none	server statfs calls/sec

Statfs is the number of NFS_STATFS element calls.

nfsstat -n item	UX/RPM item	SysScope item
symlink	none	server symlink calls/sec

Symlink is the number of NFS_SYMLINK element calls.

nfsstat -n item	UX/RPM item	SysScope item
write	none	server write calls/sec

Write is the number of NFS_WRITE element calls.

nfsstat -n item	UX/RPM item	SysScope item
wrcache	none	server write cache/sec

Write cache is the number of NFS_WRITECACHE element calls. This is an obsolete call and not reported by DG/UX.

nfsstat -n item	UX/RPM item	SysScope item
none	server procedure calls	none

Procedure calls is the total number of remote procedure calls generated by NFS per second.

Figure 5: NFS Client Statistics

nfsstat -n item	UX/RPM item	SysScope item
nclseeps	client Sleeps	client sleeps/sec

Sleeps is the number of times that a client NFS call had to be deferred because resources were not available per second.

nfsstat -n item	UX/RPM item	SysScope item
nclget	client gets	client gets/sec

Gets is the number of times a new client handle had to be created per second.

nfsstat -n item	UX/RPM item	SysScope item
none	client procedure calls	none

The number of Remote Procedure Calls generated by NFS.

Continued from page 29

vice request on one system in a network, packages the request, and transmits it over the network to an RPC server. The RPC server unpackages the request on another system and executes the request. The results are then passed back to the RPC client, which passes them to the original requester.

Guidelines for RPC-related NFS activity

You need to monitor the number of calls to ensure the server NFS is able to handle the activity. If your application uses NFS-mounted file systems, monitoring the server NFS calls received (calls) and rejected (badcalls), and the counts and percentages for the various calls that are made will help you determine any network or application bottlenecks.

The client remote procedure call (RPC) activity shows the number of calls sent and rejected, as well as, the number of times a CLIENT handle was received (*nclget*), the number of times a call had to sleep while awaiting a handle (*nclsleep*), and a count of the various calls and their respective percentages. If your application uses NFS-mounted file systems, monitoring the client NFS calls and type of calls will help you determine any network or application bottlenecks.

NFS statistics

The network file system (NFS) is a distributed file system product developed and licensed by Sun Microsystems. NFS gives users the ability to access files over a communication link as if they were resident on the user's local machine. NFS is typically found running over Ethernet, Token Ring, FDDI (fiber optic) LANs (local area networks). NFS uses RPCs and, like RPC, has both servers and clients.

The *netstat -n* option displays the number of calls and percentage of NFS_XX procedure calls by element of the total number of remote procedure calls generated by NFS per second.

NFS client statistics include all the items reported in the NFS server statistics. Figure 5 (page 31) lists additional NFS client statistics.

Guidelines for NFS activity

The Unix commands *mount* and *exportfs* control how remotely mounted file systems will interact with the NFS server. The following options may be used on the server: *ro* | *rw* allows read-only or read-write access; *nosuid* setuid execution disallowed; *bg* | *fg* if the first attempt fails, retry in the background, or, in the foreground; *retry=n* the number of times to retry the mount operation, *rsize=n* sets the read buffer size to *n* bytes; *wsize=n* sets the write buffer size to *n* bytes, *timeo=n* sets the initial NFS time-out to *n* tenths of a second; *retrans=n* the number of NFS retransmissions for soft mounts only; *port=n* the server IP port number; *soft* | *hard* returns an error if the server does not respond, or continue the retry request until the server responds; *intr* allows keyboard interrupts to kill (or signal) a process that is hung waiting for a response from a remote server; *acregmin=n* holds cached attributes for at least *n* seconds after file modification; *acregmax=n* holds cached attributes for no more than *n* seconds after file modification; *acdirmin=n* holds cached attributes for at least *n* seconds after directory update; *acdirmax=n* holds cached attributes for no more than *n* seconds after directory update; *actimeo=n* sets minimum and maximum times for regular files and directories to *n* seconds.

The defaults in DG/UX 5.4 are: *fg*, *retry*=10000, *timeo*=7, *retrans*=3, *port*=NFS_PORT, *hard*, *acregmin*=3, *acregmax*=60, *acdirmin*=30, and *acdirmax*=60. The defaults for *rsize* and *wsize* are set internally by the system kernel which is normally 8 K.

Normally, the defaults are sufficient; however, by monitoring your application's usage NFS statistics, you may need to increase some of the defaults. For example, setting the *rsize*=4096 (4 K versus 8 K) and *wsize*=4096 (4 K versus 8 K) will force more NFS operations per second but decrease overall NFS throughput.

You will need to monitor overall network activity with the *netstat* Unix command. When invoked with an interval argument, *netstat* continuously displays a running count of statistics related to network interfaces. This display shows two columns: one for all interfaces, and one for the first interface on the interface list. The first line of each screen of information contains a summary of activity since the system was last rebooted. The *netstat* command pauses the number of seconds indicated by interval before refreshing the screen. Subsequent lines of output show values accumulated over the preceding interval.

If a socket's address specifies a network but no specific host address, address formats are displayed in the form host-port or network-port. When the host and network addresses are specified, they are displayed symboli-

cally according to the data bases "/etc/hosts" and "/etc/networks", respectively. If a symbolic name for an address is unknown, or if the *-n* option is specified, the address is printed in the Internet dot format. Unspecified or wildcard addresses and ports appear as "*.*".

The interface display provides a table of cumulative statistics on packets transferred, errors, and collisions. The network address (currently Internet-specific) of the interface and the maximum transmission unit (mtu) are also displayed.

The routing table display indicates the available routes and their status. Each route consists of a destination host or network and a gateway to use in forwarding packets. The flags field shows the state of the route (U if up), whether the route is to a gateway (G), or whether the route is to a particular host (H). (Routes with an H flag appear as the result of an ICMP redirect or someone using the route command with the host parameter.) Direct routes are created for each interface attached to the local host. The *refcnt* field gives the current number of active uses of the route. Connection-oriented protocols normally hold on to a single route during a connection; protocols without connections obtain a route, then discard it. The use field provides a count of the number of packets sent using that route. The interface entry indicates the network interface used for the route.

Guidelines for background vs. foreground

File systems mounted with the *bg* option indicate that mount is to retry in the background if the server's mount daemon (*mountd*) does not respond. The Unix *mount* command retries the request up to the count specified in the *retry=n* option. Once the file system is mounted, each NFS request made in the kernel waits *timeo=n* tenths of a second for a response. If no response arrives, the timeout is dynamically determined based upon the server's network responses for that particular call type. When the number of retransmissions has reached the number specified in the *retrans=n* option, a file system mounted with the soft option returns an error on the request; one mounted with the hard option prints a warning message and continues to retry the request.

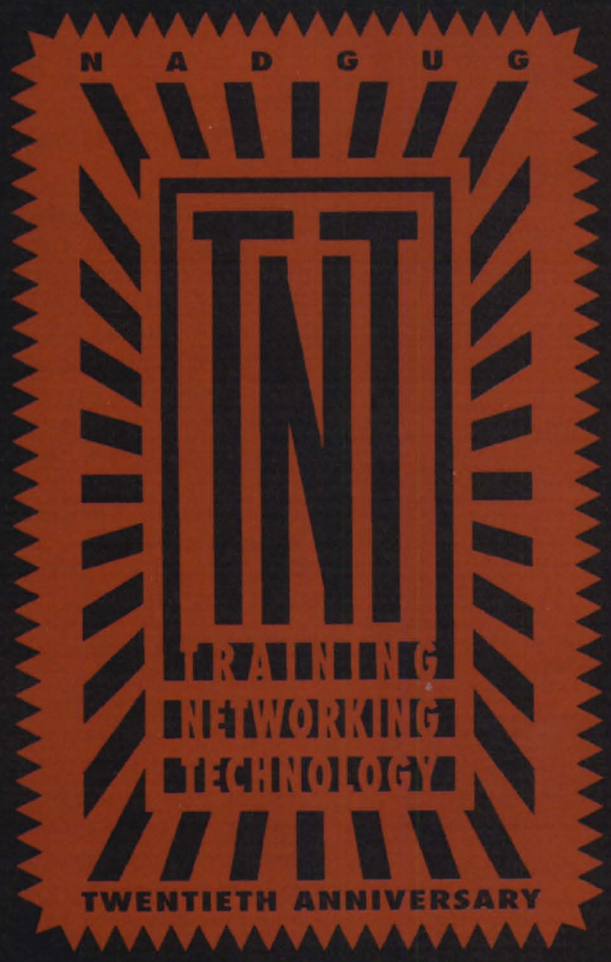
Guidelines for soft versus hard mounting

In most cases, file systems should be mounted with the *hard* and *intr* options. To prevent application failures due to application paging, even an apparently "read-only" directory like an executable or manual page directory should be mounted with the *hard* and *intr* options.

Guidelines for read-write versus read-only

NFS file systems that are mounted *rw* (read-write) should use the *hard* option to prevent possible loss of data;

Continued on page 36



New for 1993

Executive Seminars including:

- Migrating to an Open Systems Environment
- Downsizing and your Enterprise

Training Seminars including:

- Network Strategies for the 90's
- How to Select the Right AViiON Database Management System
- Recent Developments in AOS/VS II

An Extensive Exhibit of the Latest Products & Services

Unlimited Networking Opportunities

NADGUG 20th Anniversary Celebrations

OCTOBER 25-28, 1993
THE INFORUM
ATLANTA, GEORGIA

NADGUG 93

will blow you away...

Don't miss this opportunity to be part of the largest Data General gathering in the world!
Watch your mailbox for further details or for more information call 1-800-932-6663 ext. 1234

Circle 29 on reader service card.

Continued from page 34

and the *intr* option to enable keyboard interrupts.

Guidelines for setting file attributes

The attribute cache retains file attributes on the client. Attributes for a file are assigned a time to be flushed. If the file is modified before the flush time, then the flush time is extended by the time since the last modification (under the assumption that files that changed recently are likely to change soon). There is a minimum and maximum flush time extension for regular files and for directories. *Actimeo=n* sets the flush time to n seconds for both regular files and directories.

Guidelines for setting the number of *biod* and *nfsd* daemons

NFS servers that support many clients will need many *biod* and *nfsd* daemons. By default, 8 *biod* and *nfsd* daemons are started. However, if they all receive process time (refer to the *ps* command), then you need to start more *biod* or *nfsd* daemons. A heavily used NFS server may need 16 or more *nfsd* daemons. For NFS clients with their own bootable disk, there is no need to start any *nfsd* daemons and only a few *biod* daemons.

Summary

Data General Corporation has a dedicated group of performance experts, the Systems Evaluation and Performance Analysis Center (SEPAC). This group of highly trained individuals develop performance-monitoring tools and provide analysis services to DG/UX Aviion system customers and value added resellers (VARs). Specifically, SEPAC offers the following tools and services: UX/RPM, Performance Statistics Package, Performance Statistics Plus On-Site Service, and Capacity Planning. Δ

Thomas E. Soukup is a member of the Technical Services Group at Data General Corporation in Atlanta, Georgia. Currently serving as the worldwide benchmark coordinator, he has been active in benchmark performance at Data General Corporation for the past five years. He has also been involved with assisting system engineers worldwide with DG/UX performance tuning and monitoring.

'Watching' DG/UX performance

Collingham, West Yorkshire, England—As the Aviion becomes the enterprise server of choice, there is a growing requirement for well-presented, real-time performance monitoring. The Watcher is among the first of such packages to appear for the Aviion line.

Flexible logging and easy-to-use reporting are major strengths of the package. Pop-down menus control all system functions. All performance metrics are described, and context-sensitive help is available throughout the system.

In addition, the optional "Watch'n Go" service provides the professional approach to remote analysis, leading to more accurate systems optimization and capacity management.

Workstation version is $750; server version (AV 5xxx and above) is $1,500. Dealer inquiries welcome.

Synergistic Software Co. Ltd., 2 Hastings Court, Collingham, West Yorkshire LS22 5AW, England; 011-44-937-573-446.

Storage, storage, storage

Nashua, NH—Storage Computer expanded the disk drive capacity of its RAID 7 integrated rackmount units to 27 drives from 18. The company's rackmount series includes four different chassis models that can expand from 1 to 104 GB of storage capacity, support parity and hot standby drives, and 2 to 4 host interfaces.

Also available are desktop units that provide capacity expansion from 1 to 44 GB, support parity and hot standby drives, and 1 to 2 host interfaces. RAID 7 systems plug and play on a variety of systems, including Data General MV and Aviion servers.

Storage Computer Corp., 11 Riverside Street, Nashua, NH 03062; 603/880-3005. Δ

AD INDEX

Company	PG#	RS#
Ames Sciences, Inc.	24	1
Asset Remarketing	10	3
Async Systems, Inc.	23	4
CDS	11	5
Claflin & Clayton, Inc.	13	6
Claflin & Clayton, Inc.	32	7
Compuplan	39	-
Computer Engineering International, Inc.	7	8
Computer Wholesalers, Inc.	31	9
Data Bank Associates, Inc.	18	11
Data Bank Associates, Inc.	11	12
Data General Corporation	5	2
Data General Corporation	3	13
Data General Hot Site	24	10
Data General Professional Services	36	14
Data Investors Corporation	29	15
DataLynx	19	16
DataLynx	33	27
Digital Computer Consulting	14	17
Digital Data Systems	24	18
Digital Dynamics, Inc.	37	19
Eagle Software, Inc.	34	20
Egan Systems	1	21
Gerry Manning & Associates	28	22
Hanson Data Systems, Inc.	23	23
Intelligent Information Systems, Inc.	37	25
International Computing Systems, Inc.	22	24
Mac Data, Inc.	32	26
Minitab Statistical Software	39	-
Monarch Software, Inc.	C4	28
NADGUG	35	29
NADGUG	9	30
NADGUG	26	-
National Computer Dynamics	11	31
NSTS	12	32
Park Place International	21	33
PereLine Data Systems	39	-
Productivity Systems Development Corporation	25	38
R&D Computers	14	35
RAVE Computer Association	17	36
Rhintek, Inc.	6	37
SCIP	28	39
Scott McIntyre & Associates	C3	40
Security Computer Sales, Inc.	12	41
Strobe Data, Inc.	15	42
Sysgen Data	36	43
Threshold, Inc.	14	47
Transoft Ltd.	C2	48
Transoft Ltd.	15	53

PRODUCTS AND SERVICES

Company	PG#
Synergistic Software Co. Ltd.	37
Storage Computer Corporation	37

ON-LINE HELP

***Who to call for answers about* NADGUG *and* FOCUS**

NADGUG address:
c/o Danieli & O'Keefe Associates, Inc.
Chiswick Park, 490 Boston Post Rd.
Sudbury, MA 01776
FAX: 508/443-4715

FOCUS Magazine address:
c/o Turnkey Publishing, Inc.
P.O. Box 200549
Austin, TX 78720
FAX: 512/335-3083

NADGUG

Membership, RIGs, SIGs
NADGUG staff **800/253-3902**
(Outside the U.S and Canada.) **508/443-3330**

Electronic bulletin boards
(300,1200, 2400, or 9600 baud modem)
Rational Data Systems **415/499-7628**
DASH bulletin board (2400 bps modem) **800/DASH-CSC**
Data General Customer Support Center **800/344-3577**

FOCUS Magazine

512/335-2286

Editorial comments, article suggestions.....Doug Johnson
(please send product announcements to the address listed above)

Information about advertising.....Susan de Blanc

FOCUS back issues .....Turnkey Publishing staff

Continued from page 19

equals "10410". Currently, the only way to trap at that point is to set a breakpoint and keep typing RUN until the customer comes up, or add an artificial line to the program, like so:

```
IF CUSTOMER-NUMBER = "10410"
        PERFORM P1.
```

and then trap P1. Wouldn't it be much nicer to be able to type in:

```
BREAKPOINT 50 IF (CUSTOMER-
NUMBER IS EQUAL "10410")
```

and then just run the program? A feature like this will considerably ease my programming burden. To quote Mr. Ray, "A general purpose 'break-on-expression-true' or 'expression-false' capability, combined with 'test-expression-every-instruction' or 'test-every-time-break-location-is-executed' gives the capability to do practically everything. Most importantly, a shortcut way of specifying 'break-if-value-changes' or 'don't-break-if-value-changes' tremendously reduced the time spent on storage items whose values [don't] change as you expect."

Now that I'm wrapping up this review, I've decided that an editor isn't necessary after all. Just a PC, MS Windows, and a lot of memory. Here's how I ended up doing compiles/edits/and debug runs. First, I shot a copy of the program on down to the PC. Fired up Windows with an editor in one window, RAID in another, MS-DOS in a third, and an emulator hooked to the MV. This way I could edit the program while running through it with RAID, recompile the changes in the DOS window, and finally, copy any changes using the clipboard over to the MV. Editor? We don't need no stinking editor!

The RAID debugger is darn near perfect, and it's only a beta version. I can't wait to get my hands on the real version and start fixing some programs. In fact, I think I'll go write some buggy programs now, just so I'll have something to work with.

Odds And Ends Dept.

An interesting bug has cropped up in ICobol 1.71. It seems that sometimes, seemingly at random, the console-interrupt keys will hang the runtime system. Type "^C" and you're stuck. The normal remedies for a frozen terminal (^Q, CLEARD/RX) don't work, and the only available option is to terminate the PID. So far I've been lucky, and haven't seriously munged up any files.

My advice? Wait until ICobol 2 before upgrading that 1.60 system. Δ

Tim Boyer, 1993 NADGUG Vice President, is EDP Manager at Denman Tire Corporation. He may be reached at 400 Diehl South Road, Leavittsburg, OH 44430; 216/675-4249, faxed at 216/675-4232, on the NADGUG bulletin board at 415/924-3652, or on the CSC bulletin board at 800/DASH-CSC.

Continued from page 26

and maintenance, independent of program logic and data handling.

Data handling. Data of the future will be accessed through gateways, DBMS, and enterprisewide technology. Programs of the past operate on statically structured records and files. Future products will need to bridge this gap in a program-transparent fashion.

Language. Fewer VARs and users are concerned about new Cobol language standards than they are concerned about realizing the promises of 4GLs. ICobol currently defines a power ful base to launch new applications and technology while preserving past software investments. Third-party enhancements already have been made to keep up with increased user demands and expectations. This same technique will be used in the future to ensure user access to advanced software technology from within a familiar and proven development environment. Δ

Bruce K. Ray is president of Wild Hare Computer Systems, Inc. 6595 O'Dell Place, Suite M, Boulder, CO, 80301; phone 303/530-2221 or fax -9637.

DASH Items...

SYNOPSIS

Notices and queries found posted on the DASH (Direct Access to Support Help) bulletin board.

Category: DG/UX

Author: **Dave Cagigal**
Subject: ***Modems and backups***

I was wondering if anyone had any suggestions on setting up a modem on our Aviion 5225+. We are currently running 5.4.2 and cannot seem to implement a modem to dial out to another site using a system name. I do know about the "/etc/uucp/Devices", "/etc/uucp/Systems", etc., but cannot figure it out, and the manuals are not helpful. The idea that I had in mind is that our user would call a script that contained: "cu -systemname". And the modem would dial automatically. I would appreciate any suggestions that you might have.

Also, I am having a problem backing up our system. We are using the *tar* commands currently, but we cannot seem to append to existing backup tapes. I keep getting "Unable to backup blocked tapes." I just read about "/dev/rmt/ln" (no rewind device) in the *Focus* magazine July 1993, but that did not seem to work, either.

Reply by: **Home Beneficial Life Ins. Co.**

Your "/etc/uucp/Devices" file should have entries similar to: "ACU tty13,M - 19200 hayes". You can have entries for as many modems as you have attached to ports on your Aviion; just change the "tty??" above to the proper number. Also, the 19200 above is the baud rate that you are operating at (yours may be different). Your "/etc/uucp/Systems" file should have entries similar to: "dash Any ACU 19200 918003274272". "dash" could be any system name that you want (you make it up, and it is only meaningful to you). The Aviion will then look at all "ACU" instances that you have defined as 19200, and will find one not in use and use that one to dial out. The phone number is the last field. After you have these defined properly, you could dial out with the statement: "cu dash".

"dash" is the system name you defined in "/etc/uucp/Systems".

Reply by: **Ephraim Nussbaum**

Just one note. By now, I'm sure you know the command should be "cu systemname". No dash before the systemname.

Reply by: **David Dennis**

As for using *tar* for backups . . . many implementations (including DG, I think) do not support the *r* option to *tar* when used on 1/4-inch tapes (has something to do with the capabilities of the QIC format). You SHOULD be able to "mt eom" to go to the end of the tape, and then create another *tar* file there.

If you're trying to do incremental backups, I would HIGHLY recommend using "dump2" instead. It offers the additional flexibility of interactive file selection when using "restore".

Author: **Jamie Royer**
Subject: ***DG terminals on Aviion***

We are using DG D410 series terminals running in DG native mode on an Aviion 6240 running 5.4.2. Is it possible to stop the cursor from going to the top-left corner of the screen when you press DEL? It happens in sysadm, SQL*PLUS, and the Korn shell. We set up the editread for our user's menu using Korn shell, but the Korn shell is not usable for system people.

Reply by: **Steve Mason**

We are able to use D410 tubes on an Aviion with the Bourne shell and ".editreadrc" as long as the D410 is set to ANSI mode. If you need to run an application with the tube in DG mode, just echo the following to the tube upon entry into the application . . . "\033[<31" . . . when exiting the application, echo the following to the tube to set it to DG mode . . . "\036F@" . . . Believe it or not, it works! We use a script to echo the mode set, fire up the application as a child process, and then set the mode back to ANSI on exit from the child process.

Reply by: **Ephraim Nussbaum**

I've been doing the same thing with D216 screens. A script turns on DG mode or ANSI mode as we enter or leave certain applications.

Category: Hardware

Author: **Robert R. Abel**
Subject: ***Reading IBM 3480 cartridges on MV under AOS/VS***

Does anyone know if a 1/2-inch 3480 cartridge from an IBM system will fit into a 6426 130 MB cartridge drive, and if AOS/VS can read this tape through VS/Cobol or Sort/Merge or DUMP_II? A vendor of ours used to send us mag tape, but wants to start sending 3480 cartridges instead. Any ideas?

Reply by: **Ken Takemura**

DG Special Systems has a 3480 cartridge drive available for the MVs. The model number that we have used is 5583ST and is actually a Storage Technology product packaged by DG SS. It is daisy-chained to the 6299/6300 controller along with any other 6299/6300 9-track tape drives, and appears to our operating system (MIIS) as any other 6299/6300. I believe that a SCSI version is available (5583STK?). Two of our clients have this tape drive installed, and use it to back up their system instead of their 6299. It takes only 1/3 the time that the 6299s took. Δ

DASH runs on an Aviion 5200 server located at the Data General Customer Support Center in Norcross, GA. The bulletin board is available 24 hours per day, 7 days per week, free of charge. ***Access to DASH is now possible via Internet address 128.222.159.141,*** *or via modem at 1-800-DASH-CSC.*

Plug And Play And Rock All Day!

***NOW PLAYING from* Monarch Software** ♪♪
Here's what you've been waiting for! The first indisputably easy migration path to DG/UX® for INFOS® and CEO® users.

OPEN ELECTRONIC OFFICE ™ (OEO ™)

Open Electronic Office is the only easy migration path for **CEO** users moving to DG/UX. **OEO** is keystroke-for-keystroke compatible with **CEO** and provides a familiar setting that ensures users will maintain, and even improve, their productivity. **OEO** also provides multiple choices in PC integration and MV interoperability.

OPEN FILE MANAGER ™ (OFM ™)

Open File Manager is the only high performance, high reliability migration path to DG/UX for **INFOS** applications. **OFM** uses the **INFOS** API so your programs won't have to change to take advantage of the improved performance and greater recoverability features designed into **OFM**. We also offer a VS COBOL-compatible run-time system and a sort/merge, all in one cost-effective bundle.

For More Information about easy migration to open systems call: 1-800-MIGRATE

Monarch Software™

4904 Waters Edge Dr. • Raleigh, NC 27606
Phone: (919) 851-5408
Fax: (919) 851-9508
E-Mail: info @ monarch.com

Circle 28 on reader service card.