

PLEXUS SOFTWARE SUPPORT NEWSLETTER

Fall 1987

PLEXUS SOFTWARE SUPPORT NEWSLETTER

Fall 1987

Editor:

Harry K. Wong

Associate Editor:

Lynn Macey

Editorial Review Board:

Ron Green

Ric Urrutia

Contributing Authors:

Linda Allread

Alan Henley

Jeff Huntington

John Maddox

Mickey Mestel

Stephen Palmer

Sheryl Truby

Diana Yee

CDB is a trademark of Third Eye Software, Inc.

DEC, VT220 and VT100 are trademarks of Digital Equipment Corporation.

IBM and AT are trademarks of International Business Machines Corporation.

INFORMIX is a trademark of Informix Software, Inc.

MC68000, MC68010, MC68020 and MC68881 are trademarks of Motorola, Inc.

Multiplex is a trademark of Network Innovations Corporation.

Plexus, XDP, XDP System and XDE are trademarks of Plexus Computers.

RM/COBOL is a trademark of Ryan-McFarland Corporation.

UNIX is a trademark of AT&T Bell Laboratories.

PLEXUS SOFTWARE SUPPORT NEWSLETTER Fall 1987

DEPARTMENTS

Manual Error Correction Notice	1
New Product Announcements	2
Customer Satisfaction	3
Support Services Available	7
Plexus Customer Education News	10
Technical Publications Update	11
Feature Terminfo: The Wyse 350	13
Public Domain Software Spotlight	15
Tips and Techniques for Your Plexus System and Software	17

FEATURES

Customizing Your VI Editor Session	20
Debugging Programs Which Use Dynamic Memory Allocation	25

APPENDIX

Software Limitations in UNIX S/M/R 1.5	32
Third Party Software Limitations	35
Software Patches and Updates	40
Current Software Releases	42
Current Documentation Releases	51

A Note from the Editor

Welcome to the Fall 1987 edition of Plexus' Software Support Newsletter! Our features this month include a discussion of the *vi* text editor by Alan Henley of our Software Support Team. Read on as Alan highlights and explains some of the powerful features in the *vi* text editor that often remain a mystery to many of us. In addition, Stephen Palmer has written a highly useful article on debugging programs that use dynamic memory allocation.

One section not to miss is the appendix, which lists the latest software and documentation available for your Plexus computer. Please check in the appendix to see if you are fully up to date on the latest levels of Plexus software and documentation. For those customers who are still running on UNIX release S/M/R 1.5, please be sure to check the section on Software Patches and Updates.

This publication is designed to help you take full advantage of the capabilities of your Plexus computer. Each issue provides information on the latest software available on Plexus computers as well as handy information on bug fixes and current updates on our latest software.

We hope that you will find this Plexus Software Support Newsletter to be helpful. The Plexus Software Support Newsletter is a supplement to Plexus Documentation and assumes that the reader has a working knowledge of system administration on UNIX machines. If you have any comments, helpful hints or articles that may be useful to Plexus users, please let us know by sending your comments via email to lynn@FE from the Plexus Software Support Bulletin Board.

Subscribing to the Plexus Software Support Newsletter

The Software Support Newsletter is published four times a year by the Software Services Group at Plexus Computers. You may obtain this newsletter by purchasing an annual Software Support Contract from Plexus Computers through your local Plexus Sales Office.

ERRATA

A pagination error was discovered in the **Sys5 UNIX User's Reference Manual** after it was printed. The manual pages for commands MMT(1) through OD(1) are not in alphabetical order. The page order as printed is: MMT, **NROFF**, **OCW**, MORE, NEWFORM, NEWGRP, NEWS, NICE, NL, NM, NOHUP, and OD.

To correct this error, move the **NROFF** and **OCW** manual pages to just after the NOHUP manual page.

The order of the manual pages, after you follow these instructions, will be: MMT, **MORE**, NEWFORM, NEWGRP, NEWS, NICE, NL, NM, NOHUP, **NROFF**, **OCW**, and OD.

PLEXUS INTRODUCES APPLICATIONS DEVELOPMENT ENVIRONMENT FOR ITS EXTENDED DATA PROCESSING (XDP) SYSTEM

Plexus Computers introduces a fully integrated software development environment which allows users and programmers to create sophisticated database applications 10 to 20 times faster than traditional development tools and languages. The Plexus XDP Applications Designer simplifies and accelerates the design of custom application software for the company's Extended Data Processing System. Running under Microsoft Windows, the Applications Designer is the first development environment to provide easy-to-use tools for building a "what you see is what you get" user interface which is fully integrated with a standard fourth generation language and relational database management system.

These tools have been extended to include features for entering, displaying, manipulating, storing and retrieving both standard structured alphanumeric data and large amounts of unstructured data in image or text format. "These development tools allow users and programmers to produce complex data processing applications in weeks rather than the months normally required to produce the same application using traditional development tools and languages," said Tom Frederick, vice president of marketing. "In conjunction with the XDP System, Plexus offers the most powerful and flexible application development platform for managing diverse data types in the data processing industry," he said.

The Applications Designer does not require a detailed program-level understanding of Microsoft Windows, MS-DOS, UNIX, Informix, C or imaging hardware because it automatically generates the correct Microsoft Windows and C code for these operating systems, environments and languages.

The Plexus XDP System is the first comprehensive commercial computer system that integrates a relational database and departmental computer with personal computer workstations, document scanners, omnifont optical character recognition devices, laser printers and optical disks to enable users to merge and manage diverse forms of information, including image, text, alphanumeric data and voice.

The Applications Designer includes a high-level procedural language designed for data base oriented applications. Database management is handled through SQL statements. Plexus extensions support the management of Microsoft Windows menus, forms and input/display, as well as report writing that accommodates a mix of text and images. The output of these tools is a set of standard C, SQL and Microsoft Windows statements, making the application fully compatible with industry standard environments.

When using the Applications Designer, the programmer starts by building the user interface for a particular application including pull-down menus and forms in a graphically oriented "what you see is what you get" format. Basic window and user presentation characteristics (entry fields, buttons, scroll bars, titles, field attributes) are defined via a graphical dialog within the system. This feature allows programmers to quickly develop a prototype of the application which can be visually verified by the end user. Once the application has been verified, the system will then generate Microsoft Windows code. These tools are currently available for MS-DOS 3.x and Microsoft Windows. Plexus has also announced its intent to migrate these tools into the OS/2 and Presentation Manager environments when they become available.

Customer Satisfaction

by Lynn Macey, Manager of Software Support

In our last newsletter, I announced that we were changing how we handle calls. We implemented the "Response Center" in August for the morning hours. I am pleased to announce that the concept has been successful, enough so that we now run it all day long.

Now, the response statistics for the last quarter. We are changing the way they are reported in a couple of ways. In the past, exceptions were apparently not included in the averages. The numbers were provided by the Administration group and had been all summarized. The statistics are now being done so that all calls are included in the averages.

<u>Month</u>	<u>Average Call Response Time</u>	<u>Response Exceptions</u>	<u>Average Call Closure Time</u>	<u>Closure Exceptions</u>
Jul A	94 Minutes	5	13.9 Hours	7
B	296 Minutes	22	23.9 Hours	16
C	950 Minutes	39	46.2 Hours	39
D	476 Minutes	9	15.1 Hours	9
Aug A	29 Minutes	1	7.0 Hours	4
B	108 Minutes	21	20.2 Hours	10
C	128 Minutes	14	36.8 Hours	23
D	204 Minutes	5	40.3 Hours	5
Sep A	12 Minutes	0	4.2 Hours	3
B	32 Minutes	5	21.8 Hours	16
C	111 Minutes	13	38.2 Hours	15
D	93 Minutes	1	20.1 Hours	2
Oct A	5 Minutes	0	16.9 Hours	5
B	10 Minutes	1	26.6 Hours	15
C	74 Minutes	6	27.9 Hours	18
D	56 Minutes	0	15.6 Hours	1

This is the first time that we have published statistics for "C" or "D" calls or for response time exceptions. Remember that our published goals are: Respond to "A" and "B" calls within 2 hours and "C" and "D" calls in 8 hours; Close "A" calls within 2.5 working days, "B" calls with in 5 working days and "C" and "D" calls within 10 working days.

The major emphasis since August has been on the response time. When a customer calls in for assistance, we want to be able to address the problem as quickly as possible. We believe that we have been very successful in our improvement effort. The average number of calls that we handle each month is currently around 350.

In late November, Field Engineering installed an Automatic Call Distributor, or ACD, to our phone system. You will be receiving more information about how it works in a separate mailing. From the Software Support Perspective, it will give

us a tool to more easily handle the Response Center. We hope the you will not be put off by the system because the benefits include not hearing the "Plexus, please hold..." response to your phone call.

There have also been some questions about Software Support that we would like to address.

What is Software Support?

Software Support is really an information source. There are two basic things that Software Support does. We provide updates to software products and we provide consulting services to customers on the use and maintenance of Plexus computers and software.

What is supported?

A software support contract covers specific software on a specific machine or machines as the contract is written. The type of contract determines the extent of coverage and services available.

Is support part of the purchase price of the system or software?

No, the systems and software are covered by a warranty but services provided beyond the scope of the warranty are chargeable against a contract or purchase order.

What, exactly, does the warranty cover?

Plexus warrants software products to the original buyer for 90 days from the date of shipment. During that period, the software will load and execute on the machine, and configuration, for which it was delivered in accordance with the appropriate installation documentation.

During the warranty period, Plexus will:

1. Replace software products that are not loadable due to media defects.
2. Provide assistance with, or new instructions for, installation.

Plexus does not warrant that the software is free from programming errors nor that operation of the CPU or software will be uninterrupted.

Are different levels of support available?

Yes, Plexus offers two basic levels. The first is a Software Update Service or SUS agreement. It promises to deliver updates to covered software automatically when those products are released from manufacturing. These updates will bring your software and documentation to the same level as if you purchased a new copy.

The second level is the Comprehensive Software Support or CSS agreement. Like SUS it provides updates to covered software but also gives you access to Software Support analysts by phone for consulting purposes.

There is also a Basic System Support or BSS that gives phone support for obsolete products like UNIX System III.

Is that all that is included?

Well, no. Both levels include a quarterly newsletter, access to the Bulletin Board, reduced rates for other services and a few other things.

Which level is best for me?

That depends. All systems should be covered by an SUS agreement but whether you need the CSS coverage or not depends upon how you use the system and your level of experience. Customers who are well versed in UNIX as an operating system and require only occasional support may do better with an SUS agreement and go with Time and Materials for the calls that they make. Customers who are not very familiar with the system should have a Comprehensive Support agreement.

I bought my system from a VAR rather than direct from Plexus. How do I get support?

Unless you have entered a Software Support agreement with Plexus, you must call your VAR for support or pay T&M billings for the call to us. It may be true that your VAR has an agreement with us but that only allows the VAR's contacts to call for support. If you have any questions about coverage call Contract's Administration at (408) 943-2275.

I have several machines, are they all covered by one contract?

No. As a general case, a contract is written to cover a single machine by serial number. For customers with several machines, the contract must be written and priced for multiple systems. In practice, multiple machines of the same type in one location are generally indistinguishable if there is only a single contract. This is, however, not unlike several people going thru the salad bar on one plate; possible but not conducive to lasting good will.

This probably causes the most confusion and, yes, frustration for Software Support and customers alike. There was a time in the past when agreements were sold that way, however the practice had to be stopped or the price of a service agreement raised dramatically. We decided, instead, to reduce the price slightly but require that all calls be for machines listed on the contract. Exceptions require a T&M billing.

The Customer Service Representatives are instructed to to establish a billing arrangement for calls on machines that are not on contract. It's their job to do so. If you believe that our database is in error and that you are entitled to support under contract, please ask to discuss it with a manager.

I have a CSS agreement. What happens when I call for support?

When you call the toll-free hotline, you will reach a customer service representative who will ask some basic questions about who you are, and the machine that you need service for. After verifying your support status, you will be asked for a short description of the problem. Don't get technical yet, just some give a brief description of what you need assistance with.

Your call will be placed into a queue for a response analyst who will discuss you problem or question with you. The goal at this point is to solve your problem or to collect enough information to do the research necessary to close you call when an analyst calls you back.

I don't have a CSS agreement. What happens when I call for support?

The only difference is that you will have to agree to pay T&M rates and provide a purchase order.

How much will that cost?

The hourly rate for non-contract customers is \$100/hr. SUS customers receive service at \$80/hr. The minimum billing is 30 minutes.

What hours is Software Support available?

The Software Support hotline is available from 7 AM to 6 PM Monday thru Friday Pacific timezone. Support is not available on legal holidays.

What is included in an update release?

An update release will include a new version of the software, a release notice describing the installation procedure and a manual update or new manual as appropriate.

Do I have to update to new releases?

Software Support is available for the current release and previous release. The previous release is supported for 180 days (six months) from the general release of an update. After that time, support for the previous release is on a T&M basis.

Since the main purpose for an update is to correct problems encountered in earlier releases, it is strongly recommended that the update be installed.

This is the last issue of 1987, what about 1988?

1987 was marked by progress but I didn't get everything done that I had hoped for the department. I think that the service that we are providing now is better and more consistent than a year ago. But, we still have much to do. Along with the ACD, we will be installing a new database system to support our efforts. That should happen in December. I am not happy with the present Bulletin Board System and am currently looking at alternative software. I want to be able to provide quarterly reports to customers that covers usage of services and tracking of reported bugs. Plexus is also bringing out a new and exciting product based on developing database technology. It is called XDP for eXtended Data Processing and brings with it new hardware and software. We will broaden our coverage to support these new products.

As always, feel free to comment on the newsletter, our services or send us an article to publish.

1987 Support Services Available

Plexus Software Services provides several levels of software support as outlined below:

C.S.S. Comprehensive Software Support - based on system classification. Includes:

- (800) telephone hotline (limited to 2 hrs. per month); two authorized individuals per CPU. Phone services provided in excess of 2 hrs. per month will be billed at the rate of \$80/hour.
- Access to the Electronic Bulletin Board, with the ability to read information and offload programs/files
- Automatic distribution of software product and documentation updates
- Automatic distribution of quarterly newsletters
- Ability to report software bugs (verbally or written)
- A six-hour phone allowance for installation support services (for new systems only, if CSS is purchased at time of system order)
- Support for Plexus-approved peripherals
- Reduced consulting rates

S.U.S. Software Update Service - based on system classification. Includes:

- Access to the Electronic Bulletin Board, with the ability to read information only
- Automatic distribution of software product updates and documentation updates
- Automatic distribution of quarterly newsletters
- Ability to report software bugs (written only)
- Reduced consulting rates

B.S.S. Basic Software Support - intended for out-of-production products only, based on volume of CPU's purchased. Includes:

- (800) telephone hotline (limited to 2 hrs. per month); two authorized individuals per CPU. Phone services provided in excess of 2 hrs. per month will be billed at the rate of \$80/hour.
- Automatic distribution of quarterly newsletters

- Workaround support for Sys3 software bugs

Block Support - a choice of two quantities of telephone consultation time by a dedicated analyst:

A = 10 calls or 5 hours at \$375, whichever is reached first

B = 20 calls or 10 hours at \$650, whichever is reached first

Software Installation Service - Customers who purchase a CSS contract at the time of the system order will get 6 hours of free installation service. Otherwise, a choice of two programs are available (new system sales only):

- Telephone Service - \$480 for up to 6 hours

- On Site Service - \$500/day, plus expenses

Time & Materials - On-site consulting services, authorized by purchase order or American Express card

- for existing CSS, SUS, or BSS customers - \$80/hour

- for non-contract customers - \$100/hour

Installation Service for UNIX-based products

Customers who purchase a CSS contract at the time of the computer system order will receive 6 hours of free installation service. This service is intended to provide customers with telephone support while installing software and connecting peripherals (terminals, printers, modems) to work with the software.

All other customers may still obtain Installation Services on their computer systems during the first 90 days following shipment from the factory. There are two programs available:

Telephone Service - \$480 for up to 6 hours

On Site Service - \$500/day, plus travel & lodging expenses

Software Duplication

Software duplication services are available to produce 9-track or cartridge tapes from customer-supplied originals with prior Plexus management approval. The cost per duplicate is \$50 plus shipping. Contact Plexus Field Engineering to discuss duplication suitability.

Blank Media

Blank 9-track or cartridge media is available for \$35 each. Tapes used in the software duplication service will be billed to the customer if blank media is not supplied with the software original.

Priority Shipping

Plexus normally ships all software and documentation by UPS or US Mail, with an average delivery time of 5-8 days (domestic) or 15-21 days (international). At the customer's request, Plexus will ship via Federal Express (next day delivery, including Saturdays) or UPS Blue Label (48 hour delivery), but the cost of expediting shipments will be billed to the customer. Currently, Federal Express shipments average around \$25, and UPS Blue Label shipments average around \$15. If you require immediate receipt of software and/or documentation, inform the Plexus software support analyst you're talking to when the need for shipment is determined.

Plexus Customer Education News

The new Training Calendars for January through March, 1988 will be mailed in early December. Listed in the brochure are several new classes which include:

XDP System Administration - This course focuses on both the software and hardware components of the Plexus XDP system. Students will learn to effectively perform the duties of an XDP System Administrator. The skills learned in this class include managing the UNIX file system, configuring the disk, setting up and managing large data objects, performing data backup and recovery, configuring printers and PC's, maintaining user environments and administering the network.

XDP Application Development - This course provides the skills necessary to develop, implement and enhance XDP applications.

P/90 System Familiarization - Centered around the P/90 system, this course is designed to provide the skills necessary to identify, interpret and repair system malfunctions with the aid of Technical Support Specialists at Plexus.

If you presently do not receive our catalog, and wish to, please contact the Customer Education group at 408/943-2237.

Our schedule over the next few months is as follows:

Dates	Course	Location
Jan 7 - 8	Introduction to UNIX	San Jose, CA
Jan 11 - 15	UNIX System Administration	San Jose, CA
Jan 25 - 29	P/55-60-75 Maintenance	San Jose, CA
Feb 1 - 5	P/90-95 Maintenance	San Jose, CA
Feb 18 - 19	Introduction to UNIX	San Jose, CA
Feb 22 - 26	UNIX System Administration	San Jose, CA
Feb 22 - 23	P/55- 60 System Familiarization	San Jose, CA
Feb 29 - Mar 1	P/95 System Familiarization	San Jose, CA
Mar 2 - 3	P/90 System Familiarization	San Jose, CA
Mar 7 - 11	XDP System Administration	San Jose, CA
Mar 14 - 25	XDP Application Development	San Jose, CA
Mar 21 - 25	UNIX System Administration	Washington, DC

Remember, classes fill up fast, so register early. If there are any other courses you would like to see offered, please let us know. We would appreciate hearing from you.

Technical Publications Update

Plexus is continually upgrading our user documentation to better satisfy the needs of our customers. Customers supported under our software subscription service receive software manual updates as part of their support package; new customers receive the newest manuals with their systems. All customers have the option of purchasing newer manuals as they became available.

In an effort to get our improved documentation into the hands of more of our customers, we periodically publish a list of new and updated manuals. These manuals can be purchased through normal Plexus Computers ordering channels.

System Administrator's Handbook (98-05133.1 Ver. A)

This is the original Handbook written in direct response to customers' requests for more detailed system administration information. It contains all of the Plexus specific system administration information except the release specific information contained in release notices. All Sys5 systems (up to, and including the P/75 - except the P/15 - P/20) are described in this book, making it particularly useful for users administrating more than one Plexus system. We believe this book will be very helpful to anyone who performs system administration on a Plexus system, or wants to learn more about system administration on a UNIX operating system. Therefore, we strongly recommend that you obtain and use this Handbook.

System Administrator's Handbook (98-05133.1 Ver. B) - Adds P/95 information to the original handbook.

System Administrator's Handbook (98-05133.2 Ver. A) - Adds P/90 information to handbook.

Sys5 UNIX Manual Updates

The following is a list of Publication Update Packages (PUPs) that are available to update the UNIX manual set. Except where noted, these PUPs are cumulative; you must have the previous version of the manual before you can properly install the next revision. Not all Volumes of the UNIX manual set have been updated; each manual is treated separately for update purposes. The UNIX manuals not included in the following list have not been updated. The first released version of the Sys5 UNIX manual set is dated July 1, 1985. If you have an earlier set of manuals, the PUPs will not update your manuals; contact your Plexus Sales Office for instructions on how to obtain an updated set of UNIX manuals.

BOOK No.	TITLE	Version From - To	PUP No.
98-05076.1	Sys5 UNIX Administrators Guide	A to B	98-05076.11
		B to C	98-40105.11
		C to D	98-40105.12
98-05080.1	Sys5 UNIX Programmer's Guide	A to B	98-40109.11
98-05081.1	Sys5 UNIX Support Tools Guide	A to B	98-40110.11
		B to C	98-40110.12
98-05083.1	Sys5 UNIX User's Guide	A to B	98-40112.11
98-05084.1	Sys5 UNIX Admin. Reference Manual	A to B	98-05084.11
		B to C	98-40113.12
		C to D	98-40113.13
		D to E	98-40113.14
		E to F	98-40113.15
98-05085.1	Sys5 UNIX Prog. Ref. Manual	A to C	98-40114.12
		B to C	98-40114.12
		C to D	98-40114.13
		D to E	98-40114.14
		E to F	98-40114.15
98-05086.1	Sys5 UNIX User's Ref. Manual	A to B	98-05086.11
		B to C	98-40115.12
		C to D	98-40115.13
		D to E	98-40115.14
		E to F	98-40115.15
98-05077.2	Sys5 UNIX Error Messages Manual	.1 to .2	98-40106.11
98-05133.2	Sys5 Administrator's Handbook	A to B	98-40219.21

Feature Terminfo: The Wyse 350

by Dianna Yee

This month's feature terminfo entry is for the Wyse 350 terminal. Compilation of this terminfo source file will result in creation of the terminfo files *wy350* and *wyse350*. They may then be used as your TERM variable to permit operation of visual programs such as vi and programs relying on the curses features. For a full description of the meaning of each of the capabilities listed below, see **terminfo(4)** in the **Sys5 UNIX Programmer's Reference Manual**.

You can download this file from the Software Support Center bulletin board in the directory */SSC-INFO/tools/terminfo*. The file name is *wy350.ti*.

To compile the entry, log in as root and issue the command: **tic filename** where **filename** is the name of the terminfo source file. This will create the new entry *wyse350* and *wy350* in the directory */usr/lib/terminfo/w*.

```
# Wyse350 color terminal
# Terminfo source file
#
wyse350|Wyse technology model WY350,
    am, hs, xon,
    cols#80, lines#24, xmc#1,
    bel=^G, clear=^E+, cr=^r, cub1=^b, cud1=^n, cuf1=^f,
    cup=^E=%p1%'\s'+%c%p2%'\s'+%c, cuu1=^K, cvvis=^E%?,
    dch1=^EW, dl1=^ER, el=^ET, home=^E{, ht=^t, il1=^EE,
    ind=^n, is2=^E%?, kbs=^?, kcub1=^b, kcuu1=^n,
    kcuu1=^K, kf0=^A@\r, kf1=^AA\r, kf2=^AB\r,
    kf3=^AC\r, kf4=^AD\r, kf5=^AE\r, kf6=^AF\r, kf7=^AG\r,
    kf8=^AH\r, kf9=^AI\r, khome=^~, lf0=F1, lf1=F2,
    lf2=F3, lf3=F4, lf4=F5, lf5=F6, lf6=F7, lf7=F8,
    lf8=F9, lf9=F9, rmir=^Er, rmso=^EG0, rmul=^EG0,
    smir=^Eq, smso=^EG4, smul=^EGt, tsl=^Ef,
    :ZB:ZG#1:ZA=%?%p1%{0}%=%t%'0'%e%p1%{1}%=%t%'<'
%e%p1%{2}%=%t%'t'%e%p1%{3}%=%t%'4'%e%p1%{4}%=%t%'x'
%e%p1%{5}%=%t%'8'%e%p1%{6}%=%t%'p'%e%p1%{7}%=%t%'1'%;
%?%p2%t%?%p1%{0}%=%t%'Sz%'4'%;%;%?%p3%t%{2}%+%%;\EG%c

# ZA=
#           %?%p1%{0}%=%t%'0'                # encode color number
#           %e%p1%{1}%=%t%'<'
#           %e%p1%{2}%=%t%'t'
#           %e%p1%{3}%=%t%'4'
#           %e%p1%{4}%=%t%'x'
#           %e%p1%{5}%=%t%'8'
#           %e%p1%{6}%=%t%'p'
#           %e%p1%{7}%=%t%'1'
#           %;
#           %?%p2%t%?%p1%{0}%=%t%'Sz%'4'%;%;
#           %?%p3%t%{2}%+%%;
#           \EG%c
#
```

Below are the termcap entries for those applications that require them. To install the termcap entries, simply use your favorite text editor to add them onto your existing /etc/termcap file. A good idea is to put these entries near the top of the /etc/termcap file so that applications will find the wy350 termcap quicker.

```
# Wyse350 color terminal
# termcap entry
#
ZZ|wyse350|Wyse technology model WY350:\
:is=\E%?:vs=\E%?:\
:co#80:li#24:bs:am:\
:al=\EE:dl=\ER:do=^J:nd=^L:up=^K:\
:cr=^M:nl=^J:do=^J:bl=^G:ta=^I:\
:im=\Eq:ei=\Er:dc=\EW:ho=\E{:cl=\E+:\
:ce=\ET:\
:us=\EGt:ue=\EG0:ug#1:\
:so=\EG4:se=\EG0:sg#1:\
:kh=^:ku=^K:kd=^J:kr=^L:kl=^H:kb=\177:\
:cm=\E=%+ %+ :\
:hs:ts=\Ef:\fs=^M:\
:k0=^A@^M:k1=^AA^M:k2=^AB^M:k3=^AC^M:k4=^AD^M:\
:k5=^AE^M:k6=^AF^M:k7=^AG^M:k8=^AH^M:k9=^AI^M:\
:l0=F1:l1=F2:l2=F3:l3=F4:l4=F5:\
:l5=F6:l6=F7:l7=F8:l8=F9:l9=F9:\
```

Public Domain Software Spotlight

by Harry Wong

In this issue, the Public Domain Software Spotlight continues onward with its second and final installment on working with dates in UNIX. This time, we provide a Clock Confirmation and Reset software package that verifies the date and time every time the system is booted. The package provides the following functionality:

1. Presents the date and time for verification every time the system is booted. Operators can NOT exit from this software to a root shell.
2. Automatically changes the date and continues booting to multiuser mode if CPU switches are set to require an automatic autoboot to multiuser mode.
3. Identifies and corrects the clock if it deviates more than week from the last date the system was operated in multiuser mode.
4. Automatically checks the clock every hour to identify clock or system time deviations (not expected). The program also updates a file containing the current date.

Implementation of this package changes the normal booting and multiuser sequence in the following manner:

1. Verification of the date and time occurs at boot time rather than during the initiation of multiuser operation.
2. The software will initiate a reset if the date is changed. This modification is necessary to update the time the system is booted.
3. On P/15 and P/20's the package requires the system manager to reset the system after the date is changed. The system manager will also need to execute "fsck" to reconnect a unreferenced file created when going into or out of autoboot mode.

Contents of the Clock Confirmation and Reset Package

The files included in this package are:

Clock-RN	A Read-Me Information and Instructions file.
set_date	The program which checks the clock time against the last date the system was operated in multiuser mode (retained in a file which is updated once an hour by chkclk). If the date differs from the date saved when the system was last executed in multiuser mode set_date will set the date to the date saved last by chkclk. The program will ask the user to verify the date in case the system has been powered down more than a day (impossible to detect).
set_date.sh	A script which runs "set_date" and sets up autoboot flags in /etc/inittab if necessary.
set_date.sh.20	The P/15 & P/20 version of set_date.sh

chkclk	The program which checks the system date against the clock. It's primary function is to update a file containing the current date. The program will also correct the clock if the clock chip date differs from the system date. The program is to be executed once an hour by cron.
install.clock	This script installs the software and modifies all necessary profile and parameter files for all systems except the P/15, P/20, P/90, and P/95.
install.P1520	The P/15 & P/20 installation script.
install.P9095	The P/95 & P/90 installation script.
ed.itab.stdin	Commands for modifying /etc/inittab.
ed.itab.20.stdin	Commands for modifying /etc/inittab for P/15 & P/20's.
ed.itab.90.stdin	Commands for modifying /etc/inittab for P/95 & P/90's.
ed.bchk.stdin	Commands for modifying /etc/bcheckrc.
ed.prof.stdin	Commands for modifying /.profile
ed.conf.robin	Commands for modifying /usr/src/uts/m68/cf/conf.c
root.cron.mod	A line to be added to the cron entry for root

Format Of the Tape

All software, scripts, and this document are provided on cartridge or 9-track tape in cpio format.

Availability

The Clock Confirmation and Reset Package is available for the following systems:

Sys5 Revisions 1.5 & 1.7 for P/35, P/55, P/60, and P/75

Sys5 Revisions 1.6, 1.65 & 1.7 for P/90 and P/95

Sys5 Revision 1.5 for P/15 and P/20

Tips and Techniques for Your Plexus System & Software

compiled by the Staff at Plexus Software Support

1. *I'm executing uucp requests via cron. How can I find out when the transfers occurred and if they were successful?*

There are two ways to do this, one easy and one not-so-easy (of course). Both involve adding the `-m` option to the uucp request. The easy way is to use `-m` followed immediately by a file name. The entry in the crontabs file would look like this ...

```
* * * * * uucp -mlogfile myfile nodename/pathname
```

This will append a message to logfile when the transfer of myfile is completed. Just remember to clean out logfile occasionally.

The not-so-easy way involves a couple of extra steps. Using `-m` without a file name will cause mail to be sent to the requestor when the job has been completed. If you're doing this as a cron job, the mail would normally go back to root since it started cron and is therefore considered the requestor. The way around this is to set up a user crontab file in `/usr/spool/cron/crontabs` owned by the user to be notified and set the uid bit like this..

```
chmod 4744 mycrontabs<cr>
```

which will cause cron to execute requests in mycrontabs as if the owner of mycrontabs typed in the commands directly. Be careful of possible security problems with this method.

Just remember to make the appropriate entries in the crontab file and in `/usr/lib/cron/cron.allow`.

2. *How do I reference arguments from previous command lines in the C shell?*

Arguments from previous commands can be references through the history feature of the C shell individually or in groups. The command

```
% echo This is a command line from the C Shell
```

could be repeated by typing `!!<cr>`. To extract an individual argument, you would type `!!:n` where `n` is the argument number (0 based) that you want. For example, if you wanted to extract the argument "command" from the previous line, you would enter `$ echo !!:4`.

The characters `^` and `$` can be used to reference the first and last arguments respectively. You may also reference a range of arguments as `!!:3-7` or `!!:3-$`. The characters "!" mean reference the preceding command. "!#" means reference the current command line. "!n" means reference event number `n` (which you can determine by using the history command).

See **cs(1)** in the **UNIX Sys5 User's Reference Manual** (History Substitutions) for more information.

3. *How do I let UNIX know that a shell script is a C shell script or a Bourne shell script?*

Putting a "#" as the first character on the first line of a shell script indicates C shell. A ":" as the first character indicates that this is the first line of a Bourne shell script.

4. *What does /usr/plx/script allow you to do?*

Script allows you to do a tee of the standard output to a file so that you can keep a log of all activity on a terminal. To start the script, simply key in

```
script file1
```

where *file1* is the name of the ASCII log file which will log all interaction at the terminal. The user will still see standard output on the terminal. To terminate logging, simply key in:

```
exit
```

This will terminate the logging and close the file. Running `//usr/plx/more` of *file1* will show all terminal interaction since script began.

This is particularly useful when you are debugging a problem on the remote system and you want to capture all output to bring back to the local system for further investigation.

Note: Certain commands, such as *vi(1)* will not work when script is running.

5. *The uustat (for uucp status inquiry and job control) command displays status information of UUCP requests by user, system, age, and job number.*

The `-u` option and `-s` options give status information for a specific user and system, respectively. For example:

```
uustat -sralph
```

will display the status for all uucp transmissions to a system named "ralph".

The `-k(job)#` option allows the user to terminate a queued job.

The `-c(hour)` option allows the user to remove entries older than the given hour. For example:

```
uustat -c2
```

will remove all uucp entries which have been queued for more than 2 hours.

6. *How can I use find(1) and cpio(1) to do incremental backups?*

The `-newer` option to *find(1)* provides an easy way to do incremental backups. This option finds any file that has a modification time later than the modification time of the file given as an argument. The general scheme is that every time you do

full backups, you **touch(1)** a designated file on the system. Then when you run incremental backups each night, only those files that have been modified since the modification time of that file will be **cpio**'d to tape. This greatly reduces the amount of tape needed for backups and still assures that you will have as up to date backups as possible in case the need arises. The command might look something like this:

```
find /user -newer /usr/local/backupfile -print | cpio -oBm > /dev/rrm/0m
```

to back up the */user* file system. This is equivalent to doing full backups with **dump 0**, and then doing incrementals with **dump 9**, and so on. Other options can be added to the **find** statement if needed. See **find(1)** and **cpio(1)** in the **Sys5 UNIX User's Reference Manual**.

7. How can I shorten my **cd** commands to directories with long names ?

The **CDPATH** variable aids in quick **cd**'s to directories that you spend a good deal of time in. It works much like the **PATH** variable in that when you do a **cd** and **CDPATH** is set, those pathnames are scanned to see if the specified directory lies anywhere in that set of pathnames. The current working directory, '.', should be the first path in the list, then **\$HOME**, and then any other directories you want. It is recommended that you alias (or set to a function in the bourne shell) **cd** to print out the directory that you have just **cd**'d to each time you invoke **cd**. The problem is that it could be easy to lose track of which directory you are really in when you don't have to specify a full pathname. Doing a **cd bin**, then assuming that you are in a directory called *bin* in your home directory, and then doing an **rm -rf ***, can be disastrous if you happen to really be in */usr/bin*!

To set this variable in the C shell use:

```
set cdpth = ( . $HOME /usr/spool /usr/lib )
```

and in the Bourne shell:

```
CDPATH=:$HOME:/usr/spool:/usr/lib
```


Customizing Your Vi Editor Session

by Alan Henley

Overview

Vi is a powerful screen oriented editing facility which, while loaded with features, can be cryptic and cumbersome at times. This is usually because there are at least six different ways to perform the same task and no two people use vi the same way. In this article, we're going to add some fuel to the fire and pass along some tips that may help make vi friendlier to use.

One quick note for newer users. Where you see the character "" followed immediately by another character, this means press and hold the Ctrl key while typing that second character. The only exception to this is where you see "[", which means press (but don't hold) the ESC key.

We're not attempting to tell you "everything you always wanted to know about vi but were afraid to ask". Life (and our deadline) is too short.

Macros

The `map` and `map!` commands allow you to assign a sequence of commands/keystrokes to a single key or function key. This is very useful for commands that you use often but require multiple keystrokes or cumbersome hand movements. For example, to position yourself at the beginning of a file, you would enter "1G" which requires three keystrokes (1, shift, and G). To perform that command in one keystroke, let's map it to the q key by entering

```
:map q 1G<cr>
```

Now when you hit your q key during command mode, you will be repositioned to the top of the file.

You can also map a sequence of commands to a single keystroke. Suppose you wanted to move to the top of the file and open up a line just below the first line. The map command for that would be

```
:map q 1Go<cr> or :map q 1GO<cr>
```

(to open up a line above the first line in the file).

The mapped commands are evaluated from left to right which means that almost any series of commands that you can think of can be mapped to any key. This includes escapes, control characters, and previously mapped keys. To embed an escape or control character into a map command, you must "escape" them by entering `^`. Let's say you wanted to map `^f` (page forward) to your q key. The keystrokes you would enter are

```
:map q ^f<cr>
```

but what you would see would be

```
:map q ^f<cr>
```

Mapping to Function Keys

Instead of mapping `^f` to the `q` key, it might be desirable to map it to a function key. Let's use function key `#1`.

```
:map #1 ^f<cr> (remember to escape the ^f with ^v)
```

You can use `#0 - #9` to map to `F1 - F10` (where `#0 = F10`). If you're using or emulating a `vt100` you will only be able to map `F1 - F4`.

The command `:map` by itself will show a listing of all currently mapped keys. To remove the mapping from a key, enter

```
:unmap q<cr>
```

which removes the mapping to the `q` key we did earlier. To unmap a function key, you must unmap the escape sequence of the function key itself. On my Wyse 75 unmapping `F1` goes like this...

```
:unmap ^[OP<cr>
```

You can find out the escape sequences for function keys on your terminal by using `:map` by itself.

Using `:map!`

The `map` command only affects keys during command mode. But if you're like me (impatient) there are some commands you want done right away without the bother of pressing the escape key to switch modes. You can accomplish this by using the `map!` command. For example, I have mapped `F1` to get a shell from `vi`, and then resume input mode upon termination of the shell. The command to do that looks like this...

```
:map! #1 ^[:shel^Mi<cr>
```

which replaces

1. press escape key to change to command mode.
2. press/hold shift key
3. hit `:'` key to get command prompt
4. type in `shel<cr>`
5. resume input mode

Entering `:map!` by itself will show you all the keys that have commands mapped to them during input mode. Just like `:unmap`, there is an `:unmap!` to remove a `map!` to a key. You may use the same key for different `map` and `map!` commands.

Abbreviations

Abbreviations allow you to map long strings of text into a shorter series of keystrokes. This is useful if you create documents where you are continually typing the same string of characters. The command

```
:ab ple Plexus Computers Incorporated<cr>
```

maps the string "Plexus Computers Incorporated" to the keystrokes " ple ". Notice the embedded blanks before and after ple. This is because the abbreviation begins with the first character following :ab. This allows you to type "apple" and not have it substituted with "apPlexus Computers Incorporated". One useful application might be to assign nroff and troff control commands to abbreviated keystrokes.

It is possible to map a very short string to a very long abbreviation. If you enter

```
:ab lotsofrats rats<cr>
```

what you get when you enter "lotsofrats" will be "rats frats" which will look funny until you type over the rest of the abbreviation or switch to command mode.

Once you have assigned an abbreviation, you must unassign it before you can reassign it to something else. To unassign an abbreviation, enter

```
:unabb lotsofrats<cr>
```

so that you can reassign lotsofrats to "frogs".

The .exrc File

By now you're probably thinking of all kinds of things that you'd like to map and abbreviate. If you were to enter all the maps and abbreviations each time you began a vi session, you'd spend a lot of your time setting up your environment before you began to do any editing. Instead, you can place all of your setup commands into a file called .exrc in your home directory.

Vi looks for the .exrc file each time you begin a session and tailors the vi environment according to the maps, abbreviations, and sets that you put there. Here is a copy of my own .exrc file.

```
map #1 lG
map #2 G
map #3 ^F
map #4 ^B
map + :shel^M
map = :e#^M
ab ple Plexus Computers, Inc.
set redraw sm shell=/bin/sh
```

To ensure that a particular .exrc file is always used for setup each time you startup vi, set the environment variable EXINIT to something like this ...

```
setenv EXINIT 'source $HOME/.exrc'
```

or for you Bourne shell users

```
EXINIT='source $HOME/.exrc'
export $EXINIT
```

As a matter of fact, you can "source" any file. Be aware though, that when vi starts up, it always looks for a .exrc file in the current directory which will override any conflicting setups between it and the source file you specify in EXINIT.

You could have many different specialized source files to cover different editing requirements. Suppose you had a source file dedicated to nroff edit sessions. After starting up vi you could

```
:source $HOME/.nroff<cr>
```

to add new maps/abbreviations or replace existing ones. If you specify a map to a key that is already being used, the key will be remapped to the new command. The same is true for abbreviations. As a matter of fact, you could map source commands that would allow you to bounce back and forth between setups (i.e. "map #1 source \$HOME/.nroff" in your .exrc file and "map #1 source \$HOME/.exrc" in your .nroff file.)

One last note about the source command. In order to use \$HOME in the pathname, your vi shell should be "/bin/sh". The C shell will give you a bad time about sourcing to \$HOME/.exrc. If you want to use the C shell for your vi shell, you should use either the absolute or relative pathnames to the source file.

Some Other Neat Things to Know

Marking Text

Markers provide a very useful method of accomplishing several different tasks. Placing a marker allows you to refer back to an area of the file quickly without searching for a key string of text or remembering line numbers. To place a marker, type "mn" while in command mode where n is the single lower case character that identifies the marker.

At anytime during your edit session, giving the command "n" will reposition you to the beginning of the line on which you placed the n marker. Typing "n" will reposition you directly to the n marker itself. Other uses for markers include yanking or deleting blocks of text. For example, if you were to page through your document several screenfulls beyond the n marker and suddenly decide to delete everything from your current position back to the marker, you would give the command "d'n". If you wanted to yank instead, you would type "y'n".

The term "placing a mark" is somewhat misleading since the markers only exist for the duration of your edit session. A valid marker is any lower case character from a - z.

Switching Between Files

Vi allows you to bounce back and forth between two files, editing each as long as you save or discard changes between switching. After editing FILE-A, you decide to make similar changes to FILE-B. You do this by entering

```
:e FILE-B<cr>
```

which makes FILE-B your current file and FILE-A your alternate file. To switch back to FILE-A, enter

```
:e #<cr>
```

which means "make the alternate file current, and the current file the alternate". Entering `~` while in command mode will also do this. (Or do what I do and map `:"e #"` to a single keystroke.)

You can only bounce between two files at a time. If while editing FILE-B, you decide to

```
:e FILE-C<cr>
```

you will no longer be able to bounce over to FILE-A without specifically saying you want to edit that file.

Modifying scrolling commands

By default, `^f` and `^d` will page you forward by one or one half screenfulls respectively. Likewise `^b` and `^u` will page you backwards by one or one half screenfulls. By prefixing numbers to these commands, you can set the number of lines to be scrolled in either direction.

Giving the command `"80^d"`, will cause `^d` to scroll forward 80 lines. Typing `"5^u"` will cause `^u` to scroll backwards 5 lines. These changes will stay in effect until you reset them with another value, or you end your session. Of course, if you still wanted `^d` to behave normally, but also had a need to scroll forward 80 lines, you could always ...

```
:map ^i 80j
```

which would cause `^i` to scroll forward 80 lines. (Am I repeating myself?)

Still Want More?

You might want to try *An Introduction to Display Editing with Vi*, and *Ex Reference Manual* in the *Sys5 UNIX Documentation Workbench*. Most people I know have learned vi from other users somewhat like passing on an old recipe from generation to generation. If you know something about vi that you think others might like to know, drop us a line. If we get enough tips we'll pass them along in a future newsletter.

Debugging Programs Which Use Dynamic Memory Allocation

by Stephen Palmer

A constant source of aggravation to a programmer is having to debug a program which requests memory from `malloc`, `calloc` and `realloc`. These programs are difficult to debug due to the dynamic contents of the buffers thus allocated as well as the ease with which the allocation scheme may become undetectably corrupted. The most common mistake made by programmers is the failure to allocate sufficient memory for a given buffer, thus wiping out any information in memory following that buffer (generally the rest of the `malloc` pool). This section describes a scheme for easily checking the consistency of the buffers in use.

The scheme below describes *magic* versions of `malloc`, `realloc`, `calloc`, and `free`, which we will distinguish from their standard counterparts by preceding their names with `M_`. The programmer is cautioned that pointers requested from one of the magic routines must never be passed to the standard functions, and vice versa. Other than that, the pointers may be used for any (legal) purpose.

WARNING -- Never pass a pointer received from one of the *magic* routines described below to `calloc()`, `free()` or `realloc()`. Havoc is thus guaranteed.

The basic idea with this scheme is to remember the size of the buffer the user requested, and use that size to check periodically whether the buffer has been overrun by the user. Our design consists of two files: a header file, `safemem.h`, and the actual allocation routines, `safemem.c`. All programs wishing to take advantage of the safety features of this scheme must include `safemem.h`, and compile with `safemem.c` or `safemem.o`.

The routine `M_checkmagic` is provided to check the consistency of the buffers you are using. Pass `M_checkmagic` a pointer returned from one of the other safe functions. If the pointer fails the consistency test provided by `M_checkmagic`, the program will abort, producing a core dump which the programmer should examine.

NOTE -- It is the programmer's responsibility to periodically call `M_checkmagic` for each of the pointers received from one of the other safe routines. Note that `M_checkmagic` will only detect when you write off the edges of your buffer (either by calculating a negative or slightly too large address index); it will not detect your randomly writing to memory.

To turn off checking by `M_checkmagic` but leave the debugging statements in place for future use, you may place the following definitions at the top of each module which is no longer under consideration for debugging:

```
#define          M_malloc    malloc
#define          M_free      free
#define          M_realloc   realloc
#define          M_calloc    calloc
```

```
#define M_chekcmagic(x)
```

The routines in `safemem.c` can be used in place of the standard library routines for memory allocation. All take the same number of arguments as their standard counterparts.

The routines whose source follows were debugged and tested on a Plexus P/75 running SysV Release 1.5, and a P/95 running SysV Release 1.7. The test program used is also included.

```
/* safemem.c */
#include <signal.h>

#define MEMMAGIC 0x7f3e9815 /* Random pattern */

char *M_malloc(),
      *M_calloc(),
      *M_realloc();

char *malloc(),
      *calloc(),
      *realloc();

void free();

/*
 * If the program fails, look in the following buffer with the
 * debugger to find out why. This prevents us from having to
 * pull the whole standard i/o cruft into programs which wouldn't
 * otherwise have it... Also, keeps us from having hard-coded i/o.
 */
char M_reason[100];

/*
 * M_malloc
 *
 * Request the desired size from malloc(). Request enough space to
 * also store the size requested and a magic tag at the beginning and
 * end of the buffer (3 integers). We round the desired size up to
 * an even number of integers to make addressing easier.
 */
char *
M_malloc(num_ints)
register unsigned num_ints;
{
register int *ptr;

/*
 * How many integers? We allocate memory on an integer basis
 * because it makes the storing of values in the array much easier.
 * Since M_malloc takes a number of bytes, we do the division
 * here, and carry it through all of the other safe routines.
 */

```

```

*/
num_ints = (num_ints/sizeof(int)) + (num_ints % sizeof(int) ? 1 : 0);

ptr = (int *)malloc((num_ints + 3)*sizeof(int));
if (!ptr)
{
    strcpy(M_reason, "No memory in M_malloc\n");
    signal(SIGIOT, SIG_DFL);
    abort();
}

ptr[0] = MEMMAGIC;
ptr[1] = num_ints;
ptr += 2;
ptr[ptr[-1]] = MEMMAGIC;
return((char *)ptr);
}

/*
 * M_free
 *
 * Free a buffer requested from M_malloc.
 */
M_free(ptr)
register int *ptr;
{
    M_checkmagic(ptr);
    free(&ptr[-2]);
}

/*
 * M_checkmagic
 *
 * Check the magic numbers hidden in the buffer. If the user has trashed
 * them, abort.
 */
M_checkmagic(ptr)
register int *ptr;
{
    if (ptr[ptr[-1]] != MEMMAGIC || ptr[-2] != MEMMAGIC)
    {
        sprintf(M_reason, "Magic check failed on buffer at 0x%lx0,
            ptr);
        (void)signal(SIGIOT, SIG_DFL);
        abort();
    }
}

/*
 * M_calloc
 *
 * Just like M_malloc, but the arguments are different. The actual
 * work for M_malloc is done here.
 */

```



```

char
M_calloc(nelem, elsize)
unsigned nelem, elsize;
{
register unsigned size = (nelem * elsize);
register int *ptr;
static char *lastptr;

/*
 * Round up... Once again, our pointer will look like
 * an array of integers, making the rest of the code that
 * much simpler.
 */
size = (size / sizeof(int)) + (size % sizeof(int) ? 1 : 0);

/*
 * Get the memory
 */
ptr = (int *)calloc(size + 3, sizeof(int));

/*
 * Ooops!
 */
if (!ptr)
{
    strcpy(M_reason, "No memory available in M_realloc\n");
    signal(SIGIOT, SIG_DFL);
    abort();
}

/*
 * Set up for consistency checks.
 */
ptr[0] = MEMMAGIC;
ptr[1] = size;
ptr += 2;
ptr[ptr[-1]] = MEMMAGIC;
return ((char *)ptr);
}

char
M_realloc(ptr, size)
int *ptr;
unsigned size;
{
/*
 * Adjust the size to be an integer number of ints.
 */
size = (size / sizeof(int)) + (size % sizeof(int) ? 1 : 0);

/*
 * Realloc our old array
 */
ptr = realloc(&ptr[-2], (size+3)*sizeof(int));

```

```
/*
 * Oops!
 */
if (!ptr)
{
    strcpy(M_reason, "No memory available in M_calloc\n");
    signal(SIGIOT, SIG_DFL);
    abort();
}

ptr[0] = MEMMAGIC;
ptr[1] = size;
ptr += 2;
ptr[ptr[-1]] = MEMMAGIC;
return((char *)ptr);
}
```

TEST PROGRAM for Magic Memory Allocation Routines

```

/*
 * nastymem.c
 *
 * Some moderately nasty pointer arithmetic to demonstrate just how safe
 * the safemem routines for memory allocation are.
 *
 * Note that it is the programmer's responsibility to call M_checkmagic
 * periodically to make sure that an operation hasn't overrun either
 * end of the dynamic buffer.
 */

char                *ptr[100];
main()
{
  int                i;
  int                j;
  char               *M_malloc();
  char               *M_realloc();

  while (1)
  {
    /*
     * Allocate an array and clear it out.
     */
    printf("Allocating\n");
    for (i = 0; i < 90; i++)
    {
      ptr[i] = M_malloc(5*i+50);
      for (j = 0; j < (5*i+50); j++)
        ptr[i][j] = ' ';
      for (j = 0; j < i; j++)
        M_checkmagic(ptr[j]);
    }

    /*
     * Stretch the array a bit.
     */
    printf("Growing...\n");
    for (i = 0; i < 90; i++)
    {
      ptr[i] = M_realloc(ptr[i], 50*i+100);
      for (j = 0; j < (50*i+100); j++)
        ptr[i][j] = ' ';
      for (j = 0; j < i; j++)
        M_checkmagic(ptr[j]);
    }

    /*
     * Now shrink it...
     */
    printf("Shrinking...\n");
  }
}

```

```
for (i = 0; i < 90; i++)
{
    ptr[i] = M_realloc(ptr[i], 5*i + 10);
    for (j = 0; j < 5*i+10; j++)
        ptr[i][j] = ' ';
    for (j = 0; j < i; j++)
        M_checkmagic(ptr[j]);
}

/*
 * And now get rid of it.
 */
printf("Freeing....0");
for (i = 0; i < 90; i++)
{
    M_free(ptr[i]);
    for (j = i+1; j < 90; j++)
        M_checkmagic(ptr[j]);
}
}
```

Software Limitations in UNIX S/M/R 1.5

The following is a list of known problems in Release 1.5 of the UNIX Sys5 operating system:

- | | |
|----------------------------|---|
| /bin/adb | Adb does not treat numbers with a leading zero as octal. It does not currently disassemble the "dbcc" instruction, and it does not print out negative relative addresses correctly. |
| /bin/ld | The -m option lists included modules only, rather than modules and files. |
| /bin/make | The -k option (abandon work on the current entry but continue on other branches not depending on that entry) does not work. |
| /dev/rft | The cartridge streaming driver does not work properly. Use the rpt driver instead. |
| /etc/mkfs | Mkfs will not print an error message if it is invoked with a block count larger than the disk configuration allows. Fsck will catch the problem, so it is essential to run fsck after every mkfs, and check the values reported by it. |
| /usr/bin/lint | Specifying a register storage class on an argument causes lint to incorrectly report errors. |
| /usr/lib/uucp/uuxqt | Uuxqt will not correctly restart communications pending for the uux requestor, leaving the X. file in the spool directory. |
| /usr/plx/ex | If you attempt to use ex or vi on a text file which has a pattern ei:<string>: or ex:<string>: in the first five or last five lines of the file, vi will attempt to interpret the string as an editor command. Thus, a file with the string ex:q can never be edited with vi. Use ed instead. |
| /usr/plx/ff | This utility does not work. Use find instead. |
| /usr/plx/more | More sets echo on, and does not restore the previous value of echo on exit. |
| error logging | Device error and activity logging is not implemented. |
| passwd aging | Login names of more than 7 characters cause password aging to fail. Login names of other than alphanumeric characters may cause problems with system accounting. |

- regex()** The regex metacharacter "\$" used to indicate end-of-string does not work.
- /usr/lib/uucp/uucico** Uucico is affected by the longjmp bug (see below) in an adverse way. Order patches "E" and "F" from Plexus Software Support to overcome this problem.
- C Library: longjmp** longjmp does not guarantee a nonzero return value to **setjmp(3C)** as stated in the manual. If longjmp is called with a second argument of 0, or with no second argument, setjmp will behave incorrectly. The workaround is to guarantee that longjmp return a nonzero value to setjmp by passing a nonzero value as the second argument to longjmp.
- C Library: plock** The **plock(2)** system call always returns -1 and sets errno to EINVAL. This is due to a problem in the kernel system call interface. There is a simple workaround. Contact Plexus Software Support for details.

Problems in UNIX M1.5 Affecting only P/15 AND P/20

The following are problems in UNIX S/M1.5 which only affect the P/15 and P/20 machines:

- /etc/shutdown** Using shutdown from a terminal other than the system console results in some messages being sent to the console and some being sent to the terminal used for shutdown when you next enter multi-user state.
- /stand/help** This program is not complete. The User's Manual has complete documentation.
- /usr/bin/tar** The tar command will not allow you to redirect standard error.
- ioctl()** The ioctl command TCSETAF does not work as documented. It has the same effect as TCSETA.
- line protocol** XON/XOFF protocol does not work correctly for terminals using 7 data bits and even parity.

Problems in S/M1.5 affecting ACP's

ioctl() XON/XOFF protocol does not work with terminals using 7 data bits and even parity.

Sys5 System Call Caveats

C Library: access(2) The **access(2)** system call determines file accessibility with regard to the caller's REAL user id. Very often, this is not what the caller had in mind. To determine accessibility relative to the callers EFFECTIVE user or group id, use the **stat(2)** system call.

C Library: creat(2) Programs should not depend on **creat()** when a lockfile scheme is to be implemented, as the **creat()** system call will ALWAYS succeed when the caller has a user id of 0 (root) -- use

`open(filename,
O_WRONLY|O_CREAT|O_EXCL)` instead.

Third Party Software Limitations

The following sections contain information about limitations/bugs found in third party software.

EXCELAN TCP/IP

1. **Issue:** The `close` system call, when applied to socket file descriptors, returns immediately, with the socket in the process of being closed. A `socket` call requesting the same address while the `close` is in progress will fail, often causing the EXOS board to hang.

Workaround: Active (client) connections need never specify an IP port number for a socket. Prior to calling `socket`, set the port number to 0, to indicate that the system should assign a port address from the group of available ports. The socket can then be used to connect to any address on the server machine. To avoid conflicts where the server and client are on the same machine, always have the server `accept` connections on a port number less than the value of `IPPORT_RESERVED`.

PHILON FAST BASIC

1. Unable to use ADDS-3+ terminal types with this product.
2. If LPRINT command creates a line printer file, a subsequent 'rm' command to clear its contents will also remove the ability to create another file the next time the LPRINT command is executed.

UNIFY

1. ALL UNIFY EXECUTABLES: All Unify executables must be started with standard input, standard output, and standard error already open! Starting a Unify executable with any of these files not open may result in catastrophic corruption of the database or data dictionary. When starting programs using Unify HLI features from `/etc/inittab` or `crontab` files, always redirect standard input, standard output and standard error to ensure that they really are open!
2. DBLOAD: Long path names on command line (e.g. 5-directory path) may result in a core dump.
3. DBLOAD: In version 3.2-850727.1, (check the first line of your UNIFY executable, this is usually `/usr/local/unify`), default mode does not update existing records when a duplicate key is found in the data file. This has been fixed in subsequent releases.
4. DBLOAD: In version 3.2-850727.1, AMOUNT fields which are part of a COMB will not be inserted properly in all cases. This has (presumably) been fixed.
5. AFA/DBLOAD: Key fields with a specified 'unique' afa value that are also indexed with a no-dups btree will receive the appropriate default value, but will not be entered into the hash table correctly (so direct access to them is not possible). There is no workaround. This has been fixed in the upcoming release.

6. RECONF: If there is not enough space on the default file system for the reconfigure to complete, major corruption of the database may result (depending on what phase of reconstruction the program is in at the time of the failure). ALWAYS make a BUDB backup of the database before attempting to reconfigure.
7. HLI Functions: Sys5 output buffering (see release notice).
8. HLI Functions: Attempting to make a record current by calling `adrec` with an existing key will succeed but will result in `btree` problems on that record. Always test using `ackey` to determine if a record with that key already exists, then call `adrec` only if appropriate. This has been fixed in a later release.
9. HLI Functions: Numeric screen fields of length 1-4 not associated with a database field will be output incorrectly by `ptube` (and consequently, `outbuf()`). The solution is to use a LONG variable, or to create a dummy database field to associate this screen field with. Thus, all NUMERIC screen fields not associated with the database are of type LONG, regardless of the display length of the field. This is only of interest if you call `outbuf` or `ptube` to a NUMERIC 1-4 screen field not associated with database field.
10. HLI Executables: If you neglect to include a `main()` in a C executable, and you don't include the source for the `menucall` table, (page 1-21 in the Programmers' Manual), the program will fail with no error messages, and in certain conditions will just print `FOO` on the screen.
11. AFA: AFA processing is written to the current directory (not the `DBPATH` directory). The system administrator should always be in the `DBPATH` directory when modifying AFAs.
12. SCHENT: Under certain circumstances, it is impossible to lower the expected number of records for a given record type (of course, never less than the actual number present in the database). This value represents the "high-water" mark achieved some time in the past for this record type. The solution is to dump the data for that record out of the database using SQL, delete the record, reconfigure the database, add the record exactly as the old one except with the lower expected record number, reconfigure again, and then load the data back in with `DBLOAD` or SQL. This might be easier and quicker than the procedure on page 2-18 of the Reference Manual if you have a lot of records. The upcoming release of Unify will make this easier by permitting the administrator to dump the data out in an internal binary form (which allows it to be reloaded much more reliably and quickly).

QUADRATRON

Q-Office

1. Install script causes problems if specific user-defined files are not saved prior to installation. Requires redefinition and/or recustomizing. Specific requirements for files to be resident in both root directory and each user's home directory.
2. Install script causes problems with `crontab` (`QDCRON`); displays error messages to root.
3. Backspace key does not work per `termcap` entry for `QVT102` terminal.
4. In Q-Form cannot enter new data on form after defining database and form. Message "unable to open form.f" is displayed. If `form.f` is copied to

/usr/qlib/qimenu, then new data can be entered.

5. In Q-Form cannot search/inquiry on form after defining database and form. If form.f is copied to /usr/qlib/qimenu, then new data can be viewed.
6. In Q-Form cannot move box, line, etc. when defining forms, though this capability is described in the documentation.

Q-One

1. Install script causes problems if specific user-defined files are not saved prior to installation. Though documentation spells this out fairly well, there is really no need for this if installation script were written differently.
2. Backspace key does not work per termcap entry for QT102 terminal.
3. On parallel printers set with multi-pass on, printing a Q- One document, then changing the printer setting to off, does not release the mode. All printing (from Q-One to UNIX) remains in multi-pass mode.
4. Multiple edits of headers and/or footers within the same edit session causes terminal hangs and file corruption.
5. Documentation procedures for headers and footers is incorrect.
6. Conversion programs for files from Fortune to Word do not work accurately for files with multiple format lines, multiple columns, and/or special character attributes (boldface, underscore, double underscore, overstrike, or a combination of these).
7. When Q-One exits, the high intensity is left on with tvi970 terminals.
8. Q-One files converted to "external" files still contain quadratron control characters.
9. Key repeats and other means of terminal input overflow cause immediate exits to Q-One menu from the editor. Screens also seem to hang.
10. Use of tabbed indentation print incorrectly, especially when multiple lines are tabbed, causing tabs and lines to be printed in the left margin.
11. When a print job is killed in the middle of printing a document, the printer queue "jams" so that the status of waiting documents cannot be changed.

GREEN HILLS C COMPILER

The following is a list of things to watch out for when using the Green Hills C Compiler.

1. Order of evaluation of an assignment is not specified by the C language and cannot be relied upon to be performed in a given order. Previous C compilers evaluated the right-hand side of the expression first, while Green Hills C evaluates the left-hand side. This is only significant when the left hand side of the assignment has side effects which will affect the right hand side. Example:

```
*ptr++ = toupper(*ptr)
```

Assuming that ptr is (char *), this will work in earlier versions of the standard sys5/sys3 C compiler, but will not work under Green Hills in the same fashion. Assignments of this form should not be used in ANY C compiler, as the side effects of such operations are implementation dependent.

2. Certain arguments to the compiler front end (cc) are not accepted. Argument processing by the Green Hills C version of cc conforms to that found under sys3. Specifically, arguments of the form -Wx,xx (valid only under sys5) will not be processed correctly by the Green Hills C version of cc. This may require extensive changes to your make files.
3. The dash character '-' in an argument name confuses the Green Hills C version of cc. File or directory names which are passed to this version of cc must not contain the '-' character.
4. Compiling with the -g option may not be totally effective unless the -X18 option is used as well. The -X18 option prevents Green Hills C from performing register optimizations (and thus hiding local variables from the programmer).
5. Type and line number information for modules compiled with the -g option may be incorrect when dealing with local variables which are in registers. This results in local variables allocated to registers to be hidden when debugging with a source level debugger. Green Hills is fixing this serious problem.
6. Deep recursive functions, functions that make large numbers of function calls, and functions which pass very large arguments to function calls may cause memory faults. This is due to an optimization which the Green Hills C Compiler performs. The optimization causes the stack to grow (and never shrink) during the life of a block (from opening brace to closing brace). Exit from the block causes the stack to be adjusted. This optimization may be defeated with little or no performance impact by compiling with the -X23 option.
7. Users of systems with the 68881 floating point coprocessor should contact the Software Support Center to obtain versions of the C and Math libraries compiled with the -81 option (floating point support) to Green Hills C.
8. You cannot preprocess non-C code as you could with the old C compiler preprocessor.
9. SCCS "what" strings are removed by the built-in optimizer if they are not used by the program.
10. The option -X36 does not work well. Avoid using this option.
11. #include statements cannot have comments which extend beyond one line, otherwise compilation errors will result.
12. The -O and -I options must be passed to the compiler separately. The syntax -O<path> which worked with the old C compiler is not allowed with the Green Hills compiler.
13. Non-optimal code may be generated in two cases. There may be branches generated within switch statements, and there may be extraneous code generated at function entry which loads unused arguments into registers. These problems will be solved in the next release of this product.

14. Assembler output files (.s files) are kept in the working directory rather than in a temporary directory. If you have any existing .s files you wish to keep, you must rename them.

R:Office and R:Word

1. Version 3.0.3 as installed does not permit users to transfer to the shell by typing `sh`. The fix is to extract the file `/RWORD/rmenu` from the release tape and copy it to the `/RWORD` directory on your system. Use the command:

```
cpio -idumBv RWORD/rmenu </dev/tapedrive
```

Software Patches and Updates

This column is designed to notify users of new updates and bug fixes to software as they become available.

CDB Update

A new version of the C debugger is available. Version 6.1 of CDB can now be purchased from the support center. As update support is not provided for this product, existing customers with a CDB license must pay a nominal upgrade fee.

TCP/IP Update

A new release of the Excelan TCP/IP network software is now in production. This version is labeled 3.2Q, and includes numerous bug fixes to all aspects of the network software. It also includes enhancements to permit up to 16 network logins (via telnet or rlogin) onto each system on the network. The previous limit was eight network logins. All network application programs should be completely recompiled under this version for better reliability and speed.

NOS

A Software Update is currently available to Plexus Sys5 NOS users. This update prevents system crashes with the message "smfree: mem already free" and includes improvements in Sys5 NOS interprocess communications.

Hayes Modem Dialer Program

A new version of `/usr/plx/dialh`, the Hayes modem dialer program is now available on the Plexus Bulletin Board. This new version `fhayes.c` no longer dumps core and provides the capability to dial 2400 baud Hayes modems.

ACP Patch "C"

The patch "C" version of the ACP download modules improves ACP reliability. It includes corrections to the DMA handling, improves the "flush" capability, and provides additional diagnostic data. Currently, patch "C" is only functional on level S1.5, 68020 based systems. 68000 compatibility has been added.

Sys5 S1.5 Patch "E"

A new version of the file "copy.s" is provided in this patch. This patch resolves a problem causing coredumps to occur randomly in some I/O intensive programs on 68020 machines. This patch is available on the bulletin board as the file `/usr/spool/uucppublic/4300/1.5/e.z` in packed cpio format. Patch "S1.5E" is also provided on the "S1.5" patch tape.

Sys5 S/M1.5 Patch "F"

A new version of the file `/usr/lib/uucp/uucico` is provided in this patch. This patch is available on the bulletin board as the file `/usr/spool/uucppublic/4300/1.5/f.z` in packed cpio format. Patch "S1.5F" is available on both the "S1.5 patch tape" and the "M1.5 Patch tape."

NOTE: The S1.5 patch tape currently contains Patches E and F while the M1.5 patch tape contains Patch F. The naming of these patches are arbitrary. For example, there is no S/M/R 1.5 patch "A", "B" or "D".

Fall 1987

If you are experiencing any of the symptoms mentioned above, or feel you could benefit from any of these patches, please contact the Plexus Software Support Center for details.

PLEXUS COMPUTERS INC.

Current Software Releases

as of 10/30/87

OPERATING SYSTEMS					
Part #	# of Users	Model	Systems	Software Product	Current Release
Sys3					
67-00294	16	4100AC	P/25	Sys3-Z8000-cartridge	1.2
67-00343	16	4100AC	P/25	Sys3 Update	1.21
67-00295	16	4100AT	P/40	Sys3-Z8000-9-track	1.2
67-00344	16	4100AT	P/40	Sys3 Update	1.21
67-00296	32	4100BT	P/40	Sys3-Z8000-9-track	1.2
67-00345	32	4100BT	P/40	Sys3 Update	1.21
67-00297	64	4100CT	P/40	Sys3-Z8000-9track	1.2
67-00346	32	4100BT	P/40	Sys3 Update	1.21
67-00327	16	4200AC	P/35	Sys3-68000-cartridge	3.3
67-00353	16	4200AC	P/35	Sys3 Update	3.31
67-00328	16	4200AT	P/60	Sys3-68000-9-track	3.3
67-00354	16	4200AT	P/60	Sys3 Update	3.31
67-00329	32	4200BT	P/60	Sys3-68000-9-track	3.3
67-00355	32	4200BT	P/60	Sys3 Update	3.31
67-00330	64	4200CT	P/60	Sys3-68000-9-track	3.3
67-00356	64	4200CT	P/60	Sys3 Update	3.31
Sys5					
Domestic					
67-00947	80	4300AT	P/60	Sys5-68000-9-track	M1.7
67-00948	80	4300AC	P/35	cartridge	M1.7
67-00949-01	32	4300GC	P/35	20mb cartridge vol. 1	M1.7
67-00949-02	32	4300GC	P/35	20mb cartridge vol. 2	M1.7
67-00754	16	4300DC	P/15-20	Sys5-68010-cartridge	1.5
67-00941	80	4302AT	P/60-75	Sys5-68020-9-track	S1.7
67-00942	80	4302AC	P/35-55-75	cartridge	S1.7
67-00943-01	32	4302GC	P/35	20mb cartridge vol. 1	S1.7
67-00943-02	32	4302GC	P/35	20mb cartridge vol. 2	S1.7
67-00953	128	4302CT	P/90-95	Sys5-VME - 9-track	K1.7
67-00954	128	4302CC	P/90-95	cartridge	K1.7
Sys5					
International					
67-00950	80	4300IT	P/60	Sys5-68000 - 9-track	M1.7
67-00951	80	4300IC	P/35	cartridge	M1.7
67-00952-01	80	4300IG	P/35	20mb cartridge vol. 1	M1.7
67-00952-02	80	4300IG	P/35	20mb cartridge vol. 2	M1.7
67-00755	16	4300IF	P/15-20	Sys5-68010-cartridge	1.5

67-00944	80	4302IT	P/60-75	Sys5-68020 - 9-track	S1.7
67-00945	80	4302IC	P/35-55-75	cartridge	S1.7
67-00946-01	80	4302IG	P/35	20mb cartridge vol. 1	S1.7
67-00946-02	80	4302IG	P/35	20mb cartridge vol. 2	S1.7
67-00955	128	4302ICT	P/90-95	Sys5-VME - 9-track	K1.7
67-00956	128	4302ICC	P/90-95	cartridge	K1.7
NOS					
67-00543		4310AC	P/35-55-75	Sys5 - NOS - cartridge	1.4
67-00544		4310AT	P/60-75	- 9-track	
67-00545		4310DC	P/20	Sys5 - NOS - cartridge	1.4

PLEXUS COMPUTERS INC.
Current Software Releases
as of 10/30/87

COMMUNICATIONS				
Part #	Model	Systems	Software Product	Current Release
RJE/HASP				
67-00319	4211AC	P/35	68000-Sys3-cartridge	3.1
67-00320	4211AT	P/60	9-track	3.1
67-00599	4311AT	P/60-75	68000/20-Sys5 - 9-track	4.0
67-00600	4311AC	P/35-55-75	cartridge	4.0
BATCH 2780/3780				
67-00333	4231AC	P/35	68000-Sys3-cartridge	3.2
67-00334	4231AT	P/60	9-track	3.2
67-00623	4331AC	P/35-55-75	68000/20-Sys5-cartridge	4.0
67-00624	4331AT	P/60-75	9-track	4.0
TCP/IP				
67-00625	4371AT	P/60-75	68000/20-Sys5 - 9-track	3.2
67-00626	4371AC	P/35-55-75		
67-00833	4371DC	P/20	68010-Sys5-cartridge	3.2
67-00824	4371DC	P/95	68020-Sys5 - 9-track	3.2
67-00825	4371CC	P/95	cartridge	3.2
Multiplex				
67-00699	4374DF	P/15-20	68010-IBM PC floppy disk	1.1
67-00730	4374AC	P/35-55-75	68000/20-Sys5-cartridge	1.1
67-00731	4374AT	P/60-75	9-track	1.1
67-00732	4374DC	P/15-20	68010-Sys5-cartridge	1.1
3270 SNA				
67-00715	4378AT		mSNA/3270 - 9 track	2.0
67-00716	4378AC		cartridge	2.0

PLEXUS COMPUTERS INC.
Current Software Releases
as of 11/10/87

XDP-RELATED SOFTWARE				
Part #	Model	Systems	Software Product	Current Release
68-00465	MC4365	PC	Brief Program Editor	
68-00251	4219AC	PC	Telnet & FTP	
68-00252	4218AT	PC	Socket Library, Telnet & FTP	

PLEXUS COMPUTERS INC.
Current Software Releases
as of 10/30/87

APPLICATIONS				
Part #	Model	Systems	Software Product	Current Release
UNIFY				
67-00250	4218AC	P/35	68000-Sys3-cartridge	3.1b
67-00251	4219AC	P/35	(Runtime) cartridge	3.1b
67-00252	4218AT	P/60	9-track	3.1b
67-00253	4219AT	P/60	(Runtime) 9-track	3.1b
67-00504	4318AC	P/35-55-75	68000/20-Sys5-cartridge	3.2
67-00505	4318AT	P/60-75	9-track	3.2
67-00507	4319AC	P/35-55-75	(Runtime)-cartridge	3.2
67-00508	4319AT	P/60-75	(Runtime) 9-track	3.2
67-00506	4318DC	P/15-20	68010-Sys5-cartridge	3.2
67-00509	4319DC		(Runtime) 9-track	3.2
67-00702	P/35-55-75		Unify-RM/Cobol Interface-cartridge	
67-00703	P/60-75		-9-track	
67-00704	P/15-20		-cartridge	
20/20				
68-00423	4320DC/AC	P/20-35-55-75	68000/10/20-Sys5-cartridge kit	1.13
68-00424	4320AT	P/60-75	9-track kit	1.13
Q-Office				
67-00564	4222AC	P/35	68000-Sys3-cartridge	1.94
67-00565	4222AT	P/60	9-track	1.94
67-00561	4322AC	P/35-55-75	68000/20-Sys5-cartridge	1.94
67-00562	4322AT	P/60-75	9-track	1.94
67-00563	4322DC	P/15-20	68010-Sys5-cartridge	1.94
Q-One				
67-00659	4221AC	P/35	68000-Sys3-cartridge	1.94
67-00660	4221AT	P/60	9-track	1.94
67-00656	4321AC	P/35-55-75	68000/20-Sys5-cartridge	1.94
67-00657	4321AT	P/60-75	9-track	1.94
67-00658	4321DC	P/15-20	68010-Sys5-cartridge	1.94
R:Office				
67-00818	4332AC	P/35	68000/20-Sys5-cartridge	3.0.3
67-00819	4332AT	P/60	9-track	3.0.3
67-00820	4332DC	P/15-20	68010-Sys5-cartridge	3.0.3
R:Word				
67-00821	4333AC	P/35	68000/20-Sys5-cartridge	3.0.3
67-00822	4333AT	P/60	9-track	3.0.3

67-00823	4333DC	P/15-20	68010-Sys5-cartridge	3.0.3
R:Desk				
67-00475	4234AC	P/35	68000-Sys3-cartridge	3.0.2
67-00476	4234AT	P/60	9-track	3.0.2
67-00469	4334AC	P/35	68000-Sys5-cartridge	3.0.2
67-00470	4334AT	P/60	9-track	3.0.2
67-00453	4334DC	P/15-20	68010-Sys5-cartridge	3.0.2
67-00575	4344AC	P/35-55-75	68020-Sys5-cartridge	3.0.2
67-00576	4344AT	P/60-75	9-track	3.0.2
WORD:ERA				
67-00893	4360AT	P/60-75	Word:Era-cartridge	1.1
67-00893	4360BT	P/95	Word:Era-cartridge	1.1
67-00987	4360BC	P/35-55	Word:Era-9-track	1.1
67-00987	4360CC	P/90	Word:Era-9-track	1.1
67-00987	4360DC	P/15-20	Word:Era-9-track	1.1
67-00987	4360EC	P/75	Word:Era-9-track	1.1
INFORMIX SQL				
67-00913	4345AC	P/35	SQL-Sys5-cartridge	2.1
67-00913	4345BC	P/55-90	"	2.1
67-00913	4345CC	P/75-95	"	2.1
67-00917	4345BT	P/60	9-track	2.1
67-00917	4345CT	P/75	"	2.1
67-00913	4345DC	P/15-20	cartridge	2.1
INFORMIX 4GL				
67-00919	4346AC	P/35	4GL-Sys5-cartridge	1.1
67-00919	4346BC	P/55	"	1.1
67-00919	4346CC	P/75	"	1.1
67-00923	4346BT	P/60	9-track	1.1
67-00923	4346CT	P/75	"	1.1
67-00919	4346DC	P/15-20	cartridge	1.1
INFORMIX ESQL/C				
67-00925	4347AC	P/35	ESQL/C-Sys5-cartridge	2.1
67-00925	4347BC	P/55	"	2.1
67-00925	4347CC	P/75	"	2.1
67-00929	4347BT	P/60	9-track	2.1
67-00929	4347CT	P/75	"	2.1
67-00925	4347DC	P/15-20	cartridge	2.1
INFORMIX File-It!				
67-00781	4349AC	P/35	File-It!-Sys5-cartridge	3.00
67-00812	4349BC	P/55-90	"	3.00
67-00813	4349CC	P/75-95	"	3.00
67-00782	4349BT	P/60	9-track	3.00
67-00814	4349CT	P/75-95	"	3.00
67-00783	4349DC	P/15-20	cartridge	3.00

INFORMIX C-ISAM

67-00931	4350AC	P/35	C-ISAM-Sys5-cartridge	3.1
67-00931	4350BC	P/55	"	3.1
67-00931	4350CC	P/75	"	3.1
67-00935	4350BT	P/60	9-track	3.1
67-00931	4350CT	P/75	"	3.1
67-00786	4350DC	P/15-20	cartridge	3.1

PLEXUS COMPUTERS INC.
Current Software Releases
as of 10/30/87

LANGUAGES				
Part #	Model	Systems	Software Product	Current Release
RM/COBOL				
67-00487	4104AC	P/25	Z8000-Sys3-cartridge	2.1A
67-00488	4104AT	P/40	9-track	2.1A
67-00489	4105AC	P/25	(Runtime) cartridge	2.1A
67-00490	4105AT	P/40	(Runtime) 9-track	2.1A
67-00483	4204AC	P/35	68000-Sys3-cartridge	2.1A
67-00484	4204AT	P/60	9-track	2.1A
67-00485	4205AC	P/35	(Runtime) cartridge	2.1A
67-00486	4205AT	P/60	(Runtime) 9-track	2.1A
67-00477	4304AC	P/35-55-75	68000/20-Sys5-cartridge	2.1A
67-00478	4304AT	P/60-75	9-track	2.1A
67-00480	4305AC	P/35-55-75	(Runtime) cartridge	2.1A
67-00481	4305AT	P/60-75	(Runtime) 9-track	2.1A
67-00479	4304DC	P/15-20	68010-Sys5-cartridge	2.1A
67-00482	4305DC	P/15-20	(Runtime) cartridge	2.1A
PHILON COBOL				
67-00501	4336AC	P/35-55-75	68000/20-Sys5-cartridge	1.32
67-00502	4336AT	P/60-75	9-track	1.32
67-00503	4336DC	P/15-20	68010-Sys5-cartridge	1.32
PHILON FAST/BASIC-M				
67-00794	4337AC	P/35-55-75	68000/20-Sys5-cartridge	1.35
67-00795	4337AT	P/60-75	9-track	1.35
67-00796	4337DC	P/15-20	68010-Sys5-cartridge	1.35
SMC BASIC				
67-00419	4209AC	P/35	68000-Sys3-cartridge	6.5.8
67-00420	4209AT	P/60	9-track	6.5.8
67-00707	4309AC	P/35-55-75	68000/20-Sys5-cartridge	6.5.10
67-00708	4309AT	P/60-75	9-track	6.5.10
67-00709	4309DC	P/15-20	68010-Sys5-cartridge	6.5.10
CDB C-Debugger				
67-00325	4223AC	P/35	68000-Sys3-cartridge	1.0
67-00326	4223AT	P/60	9-track	1.0
67-00826	4323AC	P/35-55-75	68000/20-Sys5-cartridge	6.1
67-00827	4323AT	P/60-75	9-track	6.1

67-00828	4323DC	P/15-20	68010-Sys5-cartridge	6.1
SVS PASCAL				
67-00400	4215AC	P/35	68000-Sys3-cartridge	2.4
67-00401	4215AT	P/60	9-track	2.4
67-00837	4315AC	P/35-55-75	68000/20-Sys5-cartridge	2.6
67-00838	4315AT	P/60-75	9-track	2.6
67-00839	4315DC	P/15-20	68010-Sys5-cartridge	2.6
SVS FORTRAN				
67-00425	4225AC	P/35	68000-Sys3-cartridge	2.4
67-00426	4225AT	P/60	9-track	2.4
67-00834	4325AC	P/35-55-75	68000-Sys5-cartridge	2.6
67-00835	4325AT	P/60-75	9-track	2.6
67-00836	4325DC	P/15-20	68010-Sys5-cartridge	2.6
67-00789	4326AC	P/35-55-75	68020-Sys5-cartridge	2.6
67-00790	4326AT	P/60-75	9-track	2.6
67-00787	4327AC	P/35-55-75	68020 w/68881-cartridge	2.6
67-00788	4327AT	P/60-75	9-track	2.6
Greenhills C Compiler				
67-00745	4303AT	P/60-75	68000/20-Sys5-9-track	1.8.01
67-00746	4303AC	P/35-55-75	-cartridge	1.8.01
67-00747	4303DC	P/15-20	-cartridge	1.8.01

PLEXUS COMPUTERS INC.
Current Documentation Releases
as of 10/30/87

SYSTEMS DOCUMENTATION	
Part #	Description
SYSTEMS MANUALS	
98-05075.1 - Ver. A	P/15-20 Users Manual for Sys5 - 68010
98-05014.7	P/25 Users Manual for Sys3 - Z8000
98-05043.7 - Ver. C	P/35 Users Manual for Sys3 - 68000
98-05001.6 -	P/40 Users Manual for Sys3 - Z8000
98-05041.6 - Ver. C	P/60 Users Manual for Sys3 - 68000
OPERATING SYSTEMS	
Sys3	
98-05045.8	UNIX Programmer's Manual - Vol. 1A
98-05046.8	UNIX Programmer's Manual - Vol. 1B
98-05036.5	UNIX Programmer's Manual - Vol. 2A
98-05037.5	UNIX Programmer's Manual - Vol. 2B
98-05047.2	UNIX Programmer's Manual - Vol. 2C
98-05054.3	UNIX Programmer's Manuals - 5-volume set
98-40092.2 - Ver. A	Sys3 - 1.2 Release Notice for Z8000
98-40135.2 - Ver. A	Sys3 - 1.21 Update Notice for Z8000
98-40099.1 - Ver. B	Sys3 - 3.3 Release Notice for 68000
98-40140.1 - Ver. B	Sys3 - 3.31 Update Notice for 68000
Sys5	
98-05082.1 - Ver. A	Sys5 Description
98-05079.0 - Ver. A	Sys5 Operator's Guide
98-05077.2 - Ver. A	Sys5 Error Guide
98-05078.1 - Ver. A	Sys5 Graphics Guide
98-05081.1 - Ver. C	Sys5 Support Tools Guide
98-05076.1 - Ver. D	Sys5 Administrator's Guide
98-05083.1 - Ver. B	Sys5 User's Guide
98-05084.1 - Ver. F	Sys5 Administrator's Reference Manual
98-05100.2 - Ver. C	Sys5 Administrator's Set (Guide & Reference)
98-05080.1 - Ver. B	Sys5 Programmer's Guide
98-05085.1 - Ver. F	Sys5 Programmer's Reference Manual
98-05086.1 - Ver. F	Sys5 User Reference Manual
98-05096.2 - Ver. B	Sys5 Documentation (14 volume set)
98-05091.1 - Ver. A	Sys5 Documenter's Workbench - Introduction
98-05092.1 - Ver. A	- Text Formatter
98-05093.1 - Ver. A	- Work Macro
98-05094.1 - Ver. A	- Work Preprocessor

98-05101.1 - Ver. A	- Complete Set
98-05103.1 - Ver. A	Sys5 UNIX Run-time Manual
98-05133.1 - Ver. B	Sys5 Administrator's Handbook (includes P/90)
98-05133.2 - Ver. B	Sys5 Administrator's Handbook (includes P/90-95)
98-40199.7 - Ver. E	Sys5 - S/M/K 1.7 Release Notice
98-40318.1 - Ver. A	Sys5 - S/M/K 1.7 Update Notice
NOS	
98-05107.2 - Ver. D	NOS Users Manual for Sys5 (incl. Release Notice)
98-40142.2 - Ver. D	NOS Users Manual for Sys5
98-40044.1	Using Plexus NOS
98-40046.1	Installing NOS Hardware
98-40212.0 - Ver. A	NOS 1.2 Patch for P/15-20

PLEXUS COMPUTERS INC.
Current Documentation Releases
as of 11/10/87

APPLICATIONS DOCUMENTATION	
Part #	Description
Unify	
98-40077.2 - Ver. A	Unify 3.1b Release Notice (for Sys3 or Sys5)
98-40161.1 - Ver. B	Unify 3.2 Release Notice (for Sys5 only)
98-05052.3 - Ver. A	Unify 3.2 Reference Manual
98-05053.4 - Ver. A	Unify 3.2 Tutorial Manual
98-05116.1	Unify 3.2 Programmer's Manual
Informix	
98-05152.1	C-ISAM Reference Manual
98-40261.1	C-ISAM Release Notice
98-05151.1	ESQL-C Reference Manual
98-40262.1	ESQL-C Release Notice
98-05150.1	SQL Reference Manual
98-40263.1	SQL Release Notice
98-05153.1	File-It! Reference Manual
98-40264.1	File-It! Release Notice
98-05149.1	4GL 3-volume set (Reference Manuals & User Guide)
98-40265.1	4GL Release Notice
20/20	
98-05117.1	20/20 User's Manual (includes Release Notice)
Quadratron	
98-05099.2 - Ver. A	Q-Office 1.94 User's Manual
98-05099.3 - Ver. A	Q-Office 1.94 Set (incl. Q-One W/P Manual)
98-40145.2 - Ver. A	Q-Office 1.94 Release Notice
98-05061.0	Q-One Word Processing Manual
98-05064.1	Q-One Tutorial Guide
98-05065.1	Q-One Reference Guide
98-05065.3 - Ver. A	Q-One Set (both Tutorial and Reference Guide)
98-40094.3 - Ver. A	Q-One Release Notice
R Systems	
98-05108.1 - Ver. A	R:Office Operator's Manual
98-40165.3 - Ver. A	R:Office 3.0.3 Release Notice (includes R:Word)
98-05109.1 - Ver. A	R:Word Operator's Manual
98-05110.1 - Ver. A	R:Desk Operator's Manual
Word:Era	
98-05188.1 - Ver. A	Word:Era Manual Set

**LANGUAGE
SOFTWARE DOCUMENTATION**

RM/COBOL

98-05066.2 - Ver. A	RM/COBOL 2.0c Reference Manual
98-40163.1 - Ver. A	RM/COBOL 2.1A Release Notice (Sys3 - Z8000)
98-40163.2 - Ver. A	RM/COBOL 2.1A Release Notice (Sys3 & 5 - 68000/10/20)

SVS FORTRAN

98-05097.3 - Ver. A	(includes use with MC68881 co-processor)
98-40234.2	SVS FORTRAN 2.6 Reference Manual
	SVS FORTRAN 2.6 for 68020 Release Notice

SVS PASCAL

98-05057.2 - Ver. A	SVS PASCAL 2.3 Reference Manual
98-40159.1	SVS PASCAL 2.4 Release Notice (Sys3)
98-40158.1	SVS PASCAL 2.4 Release Notice (Sys5)

SMC BASIC

98-05051.1	SMC BASIC Thoroughbred Manual
98-40102.4 - Ver. A	SMC BASIC 6.5.8 for Sys3 Release Notice
98-40102.5 - Ver. A	SMC BASIC 6.5.10 for Sys5 Release Notice

Philon FAST/BASIC-M

98-40225.1 - Ver. A	PHILON BASIC-M 1.35 Release Notice
98-05122.1 - Ver. A	PHILON BASIC-M Programmer's Reference Manual

Philon COBOL

98-40188.1	PHILON COBOL 1.32 Release Notice
98-05119.1	PHILON COBOL Programmer's Reference Manual

Greenhills C Compiler

98-40240.1 - Ver. C	GREENHILLS C COMPILER Release Notice
98-05144.1 - Ver. B	GREENHILLS C COMPILER Users Manual

CDB C-Debugger

98-40155.3	CDB 6.1 Release Notice
------------	------------------------

**COMMUNICATIONS
SOFTWARE DOCUMENTATION**

BATCH 2780/3780

98-40059.3	BATCH 2780/3780 3.2 Release Notice
98-40200.1	BATCH 2780/3780 4.0 Release Notice

RJE/HASP

98-40201.1 - Ver. B	RJE/HASP 4.0 Release Notice
---------------------	-----------------------------

TCP/IP

98-40211.3	TCP/IP 3.2 Release Notice
98-05131.1	Excelan EX0S - 8010 Reference Manual

MULTIPLEX

98-05124.2

98-40244.1

98-40245.1

98-30029.1

Multiplex User's Manual

Multiplex Host Installation Guide

Multiplex PC Installation Guide

One Page Notice

mSNA/3270

98-40223.1 - Ver. A

98-40226.2

mSNA/3270 User's Manual

mSNA/3270 2.0 Release Notice