

PLEXUS SOFTWARE SUPPORT NEWSLETTER

Spring 1988

VT20 and VT20 are trademarks of Digital Equipment Corporation

PLEXUS SOFTWARE SUPPORT NEWSLETTER

Spring 1988

Editor:

Mickey D. Mestel

Associate Editor:

Lois Nemecek

Editorial Review Board:

Lynn Macey

Ric Urrutia

Contributing Authors:

Linda Allread

Ray Estrada

Alan Henley

Jeff Huntington

Lynn Macey

John Maddox

Mickey Mestel

Lois Nemecek

Stephen Palmer

Sheryl Truby

Ric Urrutia

Dianna 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.

A Note from the Editor

Welcome to the Spring 1988 edition of the Plexus Software Support Newsletter! Our feature articles this month are a discussion on the use of **fsdb(1m)** for patching damaged file systems and the use of **uucp(1)** for efficiently transferring files between systems. The article on **fsdb** is somewhat of a continuation of the article in the Summer Newsletter on **fsck(1m)**. From the information contained in these articles, you should gain a better understanding of these two major file system utilities.

To improve call response time, we have implemented a new database to track software calls and a new phone system to more quickly route calls to the analysts. This created some response delays in the beginning, but problems are quickly being worked out. We should be up to speed soon and ask for your patience in the meantime.

We have also undergone some staff changes: Steve Palmer has moved to the Communications Development Group in our Engineering department, Harry Wong is no longer with Plexus, and Lois Nemecek and Ray Estrada have joined the group as new analysts. Both Lois and Ray have previous software support experience and have received extensive training on all of our products. You may have talked with one or both of them if you have placed any calls recently, as they both are now taking phone calls.

All the Analysts have been undergoing training for the last few months on various products. This has caused a shortage on the phones that we hope hasn't been an inconvenience. The major portion of the training is now complete and we are looking forward to working as a full staff again.

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.

PLEXUS SOFTWARE SUPPORT NEWSLETTER Spring 1988

DEPARTMENTS

Customer Satisfaction	1
1988 Software Support Services	5
Plexus Customer Education News	8
Technical Publications Update	10
Feature Terminfo: The Wyse 60	12
Public Domain Software Spotlight	15
Tips and Techniques for Your Plexus System and Software	17

FEATURES

Using fsdb To Repair Damaged File Systems	20
Efficient UUCP File Transfers	28

APPENDIX

Software Limitations in UNIX S/M/K1.7	31
Third Party Software Limitations	34
Software Patches and Updates	38
Current Software Releases	39
Current Documentation Releases	49

Customer Satisfaction

by Ric Urrutia, Response Center Supervisor

In August of 1987 we implemented the "Response Center" for the morning hours. The concept was so successful that it is now being run all day.

Here are the response statistics for the last three months. In August, we changed the way statistics are reported. In the past, exceptions were not included in the averages. The statistics are now being done so that all calls are included in the averages. Starting with this newsletter, we are also adding overall statistics for each month, along with the total calls that were processed for the month.

Month	Average Call Response Time	Response Exceptions	Average Call Closure Time	Closure Exceptions	Total Calls
Dec A	0 Minutes	0	9.9 Hours	3	21
B	12 Minutes	1	27.7 Hours	2	42
C	37 Minutes	4	29.4 Hours	16	171
D	22 Minutes	0	11.3 Hours	1	24
Overall	29 Minutes	5	26.66 Hours	22	258
Jan A	8 Minutes	1	8.3 Hours	1	37
B	32 Minutes	5	23.2 Hours	7	100
C	22 Minutes	1	20.3 Hours	11	182
D	4 Minutes	0	23.6 Hours	2	39
Overall	22 Minutes	7	20.0 Hours	21	358
Feb A	3 Minutes	1	9.2 Hours	3	36
B	20 Minutes	5	17.7 Hours	13	126
C	11 Minutes	1	15.1 Hours	4	93
D	31 Minutes	1	18.6 Hours	2	28
Overall	16 Minutes	8	15.8 Hours	22	283

We publish statistics for all calls that we handle, including exceptions. Our published goals are: Respond to "A" and "B" calls within 2 hours and "C" and "D" calls within 8 hours; Close "A" calls within 2.5 working days, "B" calls within 5 working days and "C" and "D" calls within 10 working days.

The major emphasis since August has been on 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 320.

In late November, Customer Support installed an Automatic Call Distributor, or ACD, to our phone system. You should have received more information about how it works in a separate mailing. From the Software Support perspective, it gives 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 functions that Software Support performs: 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 it 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 Plexus. It may be true that your VAR has an agreement with Plexus, 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 had to be 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 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 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 your 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 your 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 \$125/hr. SUS customers receive service at \$100/hr. The minimum billing is 1 hour.

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.

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

1988 Software Support Services

1. SOFTWARE SUPPORT SERVICES

1.1 GENERAL INFORMATION

All post-sales support for software sold by Plexus Computers is coordinated from the Software Support Center (SSC) in San Jose California. The Software Support Center provides:

- A "Toll-free" telephone hotline for contract customers to report problems or request support. The amount of consulting time provided each month is set by the contract level purchased by the customer. Services provided in excess of the contract limit per month will be billed at the prevailing consulting rate.
- A quarterly newsletter of articles on the use of Plexus equipment, optional software, and administrative information.
- An Electronic Bulletin Board containing information, patches and software.
- Automatic distribution of new software and documentation releases.
- Coordination of software error reporting.

1.2 WARRANTY

Plexus warrants software products to the original buyer for 90 days from 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, or that operation of the CPU or software will be uninterrupted.

If, during the warranty period, an update release of the software becomes available, it will be provided at no additional charge.

1.3 SUPPORT SERVICES AVAILABLE

Plexus has established Software Support as part of the Customer Support Group. The charter of this group is to provide post-sales technical assistance to customers, allowing customers to better utilize their systems and staffs.

The product delivered by this group is information and it takes two forms. The first is periodic updates to licensed software. As updated versions of software become available, customers with support agreements receive them automatically.

The second is access to experts. The normal method for customers receiving support is via phone. Contract customers may call in on a toll free "hotline" and will be connected with a Systems Analyst who will discuss the problem and work with the customer until a satisfactory resolution is obtained.

Plexus offers software support contracts at different levels that allow the customer to choose the most cost effective support plan for their needs.

In all cases, a contract covers specific software on a machine identified by serial number. Customers having multiple machines must also cover those machines if they intend to receive support without paying time and materials (T&M) rates.

1.3.1 CONTRACT SERVICES

The support plans are:

CSS - Comprehensive Software Support is a full service contract. It covers the Unix operating system and any third party packages listed with it on the contract. It provides:

- [1] A toll free telephone "hotline" that allows authorized individuals to call into Software Support and receive up to two (2) hours of support per month. Phone services provided in excess of 2 hours per month will be billed at the \$100 rate.
- [2] Access to the Electronic Bulletin Board, with the ability to read information and offload programs and files.
- [3] Automatic distribution of software product and documentation updates. One copy per contract is sent to the primary customer contact address.
- [4] Automatic distribution of quarterly newsletters.
- [5] Ability to report software bugs (verbally or in written form).
- [6] A six hour phone allowance for installation support services (for new systems only, if CSS is purchased at time of system order).
- [7] Support for Plexus approved peripherals.
- [8] Reduced consulting rates.

SUS - Software Update Service is a reduced level of service that keeps the customer up to date on software and information distribution, but does not allow the customer to call in for support. It covers the Unix operating system and any third party packages listed on the contract. It includes:

- [1] Access to the Electronic Bulletin Board, with the ability to read information and offload patches only.
- [2] Automatic distribution of software product and documentation updates. One copy per contract is sent to the primary customer contact address.
- [3] Automatic distribution of quarterly newsletters.
- [4] Ability to report software bugs (written form only).
- [5] Reduced consulting rates.

BSS - Basic Software Support is intended for out-of-production products only and includes:

- [1] A toll free telephone "hotline" that allows authorized individuals to call into Software Support and receive up to two (2) hours of support per month. Phone services provided in excess of 2 hours per month will be billed at the \$100 rate.
- [2] Access to the Electronic Bulletin Board, with the ability to read information and offload programs and files.

- [3] Automatic distribution of quarterly newsletters for the latest version of the operating system.
- [4] Attention to bug reports is on a time and materials basis.
- [5] Reduced consulting rates.

XDP - XDP coverage is for Extended Data Processing systems and covers the datasever and workstations. It has the same basic features as a CSS agreement. Included in XDP coverage is a larger allotment of analyst coverage, support for MS-DOS based products and datasever tools. The XDP contract does not include updates to products not developed or manufactured by Plexus. Customers are required to register these products with the original manufacturer and obtain updates from them.

- [1] A toll free telephone "hotline" that allows authorized individuals to call into Software Support and receive up to three (3) hours of support per month. Phone services provided in excess of 3 hours per month will be billed at the \$100 rate.
- [2] Access to the Electronic Bulletin Board, with the ability to read information and offload programs and files.
- [3] Automatic distribution of software product and documentation updates. One copy per contract is sent to the primary customer contact address.
- [4] Automatic distribution of quarterly newsletters.
- [5] Ability to report software bugs (verbally or in written form).
- [6] Telephone installation support is 1.5% of the list price of the system software.
- [7] Support for Plexus approved peripherals.
- [8] Reduced consulting rates.

Plexus Customer Education News

Training Calendars for April through June, 1988 were mailed in the latter part of February. Listed below are a few changes we have made to some of the classes:

P/90 Familiarization - This course has been changed from a two day course to a one day course.

Hardware Maintenance - The five day Hardware Maintenance course and the five day P/90-95 Maintenance course have been combined into one (8) day Hardware Maintenance course. A five day P/90-95 Maintenance Update course will also be available for those students who have already taken the P/55-60-75 Maintenance course.

Any of our courses can be customized to fit your needs. For customers with a group of employees to train, these classes can be taught at the customer site or at our training facilities in San Jose or Washington, DC. For on-site training rates or more information, contact the Customer Education group.

Our new training facility in the Washington, DC area is scheduled to open sometime in the second quarter. Initially it will be staffed with one instructor and as the year progresses, additional instructors will be added. The facility will be located in our Vienna, Virginia office.

Due to the many requests for the **System Tuning** class, we have scheduled a class on **June 1 - 3** at our San Jose Training Center. This class will fill up fast, so register early.

Our schedule for the second quarter is as follows:

Dates	Course	Location
April 5 - 14	Hardware Maintenance	San Jose, CA
April 7 - 8	Introduction to UNIX	San Jose, CA
April 11 - 15	UNIX System Administration	San Jose, CA
April 18 - 22	P/90-95 Maintenance Update	San Jose, CA
May 2 - 6	XDP System Administration	San Jose, CA
May 9 - 13	XDP System Administration	Washington, DC
May 16 - 27	XDP Application Development	San Jose, CA
June 1	P/90 System Familiarization	San Jose, CA
June 1 - 3	System Tuning	San Jose, CA
June 2 - 3	P/95 System Familiarization	San Jose, CA
June 2 - 3	Introduction to UNIX	Washington, DC
June 6 - 10	UNIX System Administration	Washington, DC
June 6 - 17	XDP Application Development	San Jose, CA
June 6 - 7	P/55 - 60 System Familiarization	San Jose, CA
June 8 - 9	P/75 System Familiarization	San Jose, CA
June 14 - 23	Hardware Maintenance	San Jose, CA
June 20 -24	XDP System Administration	San Jose, CA
June 23 - 24	Introduction to UNIX	San Jose, CA
June 27 - July 1	System Administration	San Jose, CA

If there are any other courses you would like to see offered, please let us know. We would appreciate hearing from you. You can contact the Customer Education group at 408/943-2237.

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.

Customers covered by a software support agreement have automatically been sent one copy of each of these updates. If you would like additional sets or individual items, or if you have not received your automatic shipment, please contact Linda Allread at Plexus Field Engineering at (408) 943-2418.

If you are not covered by a software support agreement, you may purchase a complete set of updates, or individual update packages. Contact Linda Allread at Plexus Field Engineering at (408) 943-2418 to place your order or for more information.

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. For example, you cannot insert version E unless you have versions A,B,C, and D. Not all Volumes of the UNIX manual set have been updated; each manual is treated separately for update purposes. 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.

Complete Set of All Sys5 Documentation Updates \$125.00

BOOK No.	TITLE	Version From - To	PUP No.	Price
98-05076.1	Sys5 UNIX Administrator's Guide	A to B	98-05076.11	\$10.00
		B to C	98-40105.11	5.00
		C to D	98-40105.12	5.00
98-05080.1	Sys5 UNIX Programmer's Guide	A to B	98-40109.11	5.00
98-05081.1	Sys5 UNIX Support Tools Guide	A to B	98-40110.11	5.00
		B to C	98-40110.12	10.00
98-05083.1	Sys5 UNIX User's Guide	A to B	98-40112.11	5.00
98-05084.1	Sys5 UNIX Admin. Reference Manual	A to B	98-05084.11	15.00
		B to C	98-40113.12	5.00
		C to D	98-40113.13	15.00
		D to E	98-40113.14	5.00
		E to F	98-40113.15	5.00
98-05085.1	Sys5 UNIX Prog. Ref. Manual	A to C	98-40114.12	5.00
		B to C	98-40114.12	5.00
		C to D	98-40114.13	5.00
		D to E	98-40114.14	5.00
		E to F	98-40114.15	5.00
98-05086.1	Sys5 UNIX User's Ref. Manual	A to B	98-05086.11	10.00
		B to C	98-40115.12	5.00
		C to D	98-40115.13	5.00
		D to E	98-40115.14	5.00
		E to F	98-40115.15	5.00
98-05077.2	Sys5 UNIX Error Messages Manual	.1 to .2	98-40106.11	5.00
98-05133.2	Sys5 Administrator's Handbook	A to B	98-40219.21	5.00

Feature Termino: The WYSE 60

This month's feature terminfo entry is for the Wyse60 terminal. Compilation of this terminfo source file will result in the creation of the terminfo file `wy60`. `wy60` may then be used as your TERM variable to permit the operation of visual programs such as vi and programs relying on the `curses(3x)` 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**.

This file will be available on the Software Support Center bulletin board system. It is located in the directory `/SSC-INFO/tools/terminfo`. The file name is `wy60.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 `wy60` in the directory `/usr/lib/terminfo/w`.

NOTE: Do not compile the source code in `/usr/lib/terminfo/w`. `tic` will automatically put it there for you. Make sure the source is in another directory when you run `tic`.

```
#
# The Wyse 60 is like the Wyse 50 but with more padding.
# The reset strings are slow and the pad times vary depending
# on other parameters such as font loading. I have tried
# to follow the following outline:
#   rs1 -> set personality
#   rs2 -> set number of columns
#   rs3 -> set number of lines
#   is1 -> select the proper font
#   is2 -> do the initialization
#   is3 -> If this string is empty then rs3 gets sent.
#       So I do some initialization here.
#
# The Wyse 60s that have vt100 emulation are slower than the
# older Wyse 60s. This change happened mid-1987.
# The capabilities effected are (dch1) (dl1) (il1) (ind) (ri)
#
#####
#
# The Wyse-99GT looks a lot like the Wyse 60 except that it
# does not have the 42/43 line mode. In the Wyse-60 the "lines"
# setup parameter controls the number of lines on the screen.
# For the Wyse 99GT the "lines" setup parameter controls the
# number of lines in a page. The screen can display 25 lines max.
# The Wyse-99GT also has personalities for the VT220 and
# Tektronix 4014. But this has no bearing on the native mode.
#
wy60|wyse60|Wyse 60 and Wyse 99GT,
acsc=0wa_hjjukslrmqnxqztuyv]wpxv,am,bel=~G,blink=\EG2,bw,cbt=\EI,
civis=\E'0,clear=\E*$<100>,cnorm=\E'1,cols#80,cr=~M,cub1=~H,
cud1=~J,cuf1=~L,cup=\E=%p1%' '%+%c%p2%' '%+%c, cuu1=~K,
dch1=\EW$<10>,dim=\EGp,dl1=\ER$<5>,dsl=\EF~M,ed=\Ey$<100>,
el=\Et,fsl=~M,home=~^,hs,ht=~I$<1>,hts=\E1,il1=\EE$<4>,
ind=~J$<5>,invis=\EG1,ip=~$<3>,is1=\EcB0\EcC1,
```

Spring 1988

```

is2=\Ed$\EcD\E'\Er\EH^C\Ed\EO\Ee1\Ed*\E'@\E'9\E'1^N^T, is3=^N^T,
kbs=^H, kcbt=\EI, kcub1=^H, kcu1=^J, kcu1=^L, kcu1=^K, kdch1=\EW,
kdl1=\ER, ked=\EY, kel=\ET, kent=\E7, kf1=^A@^M, kf2=^AA^M, kf3=^AB^M,
kf4=^AC^M, kf5=^AD^M, kf6=^AE^M, kf7=^AF^M, kf8=^AG^M, kf9=^AH^M,
kf10=^AI^M, kf11=^AJ^M, kf12=^AK^M, kf13=^AL^M, kf14=^AM^M, kf15=^AN^M,
kf16=^AO^M, kHOM=\E{, khome=^^, kich1=\EQ, kil1=\EE, km, knp=\EK,
kpp=\EJ, kprt=\EP, krpl=\Er, lh#1, ll=^^K, lines#24, lw#8, mc0=\EP,
mc4=^T, mc5=^R, mir, msgr, nel=^M^J$<3>, nlab#8,
pfloc=\EZ2%p1%'?'+%c%p2%s\177, pfx=\EZ1%p1%'?'+%c%p2%s\177,
pln=\Ez%p1%'?'+%c%p2%s^M, prot=\E), rev=\EG4, ri=\Ej$<7>,
rmacs=\EH^C, rmam=\Ed., rmir=\Er, rmln=\EA11, rmso=\EG0, rmul=\EG0,
rmxon=\Ec20, rs1=\E~!\E~4$<150>, rs2=\EeF$<150>, rs3=\EwG\Ee($<150>,
sgr=%?%p8%t\E)%e\E(%;%?%p9%t\EH^B%e\EH^C%;\EG%'0%'?%p2%p6%|t%{8}%|%;%?%
p1%p3%|p6%|t%{4}%|%;%?%p4%t%{2}%|%;%?%p1%p5%|t%{64}%|%;%?%p7%t%{1}%|%;
sgr0=\E(\EH^C\EG0, smacs=\EH^B, smam=\Ed/, smir=\Eq, smln=\EA10,
smso=\EGt, smul=\EG8, smxon=\Ec21, tbc=\E0, tsl=\EF, uc=\EG8%p1%c\EG0,
wsl#45, .xon,

```

=====

The following is the termcap entry for those applications that require one.

To install the termcap entry, simply use your favorite text editor and add this into your existing `/etc/termcap` file. A suggestion is to put this wy60 entry near the top of your `/etc/termcap` file so that applications will find the wy60 termcap quicker.

```

#####
### NOTE:
### The wyse 60 runs faster when the XON/XOFF handshake
### is turned off.
#####
w6|wy60|wyse60|Wyse 60:
:am:bw:hs:km:mi:ms:bs:pt:
:co#80:li#24:kn#9:
:ae=\EH^C:al=4\EE:as=\EH^B:bt=\EI:cd=100\Ey:ce=\Et:
:cl=100\E*:cm=\E=%+ %+ :ct=\E0:dc=11\EW:dl=5\ER:
:ds=\EF\r:ei=\Er:fs=\r:ho=^^:i1=\EcB0\EcC1:im=\Eq:
:ip=3:\

:is=\Ed$\EcD\E'\Er\EH^C\Ed\EO\Ee1\Ed*\E'@\E'9\E'1^N^T:
:k1=^A@\r:k2=^AA\r:k3=^AB\r:k4=^AC\r:k5=^AD\r:
:k6=^AE\r:k7=^AF\r:k8=^AG\r:k9=^AH\r:kb=\b:kd=\n:
:kh=^^:kl=\b:kr=\f:ku=^K:nd=\f:r1=150\E~!\E~4:
:r2=150\EeF:r3=200\EwG\Ee(se=\EG0:so=\EGt:sr=7\Ej:
:st=\E1:ts=\EF:uc=\EG8%.\EG0:ue=\EG0:up=^K:us=\EG8:
:ve=\E'1:vi=\E'0:ko=bt,le,do,nd,up,dc,dl,ho,al:
w7|wy60-w|wyse60-w|wyse 60 132-column:
:cm=\Ea%i%DR%dC:co#132:dc=16\EW:ip=5:r2=300\EeF\E';:
:tc=wy60:
w8|wy60-25|wyse60-25|wyse 60 80-column 25-lines:
:li#25:r3=200\EwG\Ee):
:tc=wy60:
w9|wy60-25-w|wyse60-25-w|wyse 60 132-column 25-lines:

```

```

:li#25:r3=200\EwG\Ee):
:tc=wy60-w:
wa|wy60-42|wyse60-42|wyse 60 80-column 42-lines:
:al=11\EE:cd=260\Ey:cl=260\E*:cm=2\E=%+ %+ :dc=16\EW:
:dl=11\ER:i1=\EcB2\EcC3:ip=5:li#42:r3=150\Ee*:sr=10\Ej:
:tc=wy60:
wb|wy60-42-w|wyse60-42-w|wyse 60 132-column 42-lines:
:cm=2\Ea%i%dR%dC:co#132:dc=19\EW:ho=2^^:ip=6:r2=300\EeF\E';:
:tc=wy60-42:
wc|wy60-43|wyse60-43|wyse 60 80-column 43-lines:
:li#43:r3=150\Ee+:
:tc=wy60-42:
wd|wy60-43-w|wyse60-43-w|wyse 60 132-column 43-lines:
:li#43:r3=150\Ee+:
:tc=wy60-42-w:
we|wy60-vb|wyse60-vb|Wyse 60 Visible bell:
:vb=\E'8\E'9:
:tc=wy60:
wf|wy60-w-vb|wy60-wvb|wyse60-wvb|Wyse 60 132-column Visible bell:
:vb=\E'8\E'9:
:tc=wy60-w:
#
# Wyse 60 in economy mode with 2 pages of memory
# It may be useful to assign two function keys with the
# values \E=(\s look at old data in page 1
# \E=W, look at bottom of page 1
wg|wy60-2p|wyse 60 with 2 pages of memory:
:is=150\Ed$\EcD\E'\Er\EH^C\Ed/\EO\Ee1\Ed*\E'@\E'9\E'1\EwJ\Ew1^N^T:
:r2=150\EeG:te=\Ew1:ti=\Ew0:
:tc=wy60:
#

```

Public Domain Software Spotlight

by Lynn Macey

In this issue, our public domain spotlight will shine on an area where many users have found frustration. The Unix operating system has, from the beginning, supported communication utilities to connect with other systems as terminals, or to transfer files. In order for this to happen, a local system must be able to connect to a remote system through the phone lines using modems. However, the dial programs to accomplish this task are not very straight forward to write and are generally modem specific.

Before the release of System V, AT&T's stock Unix operating system included an Automatic Call Unit (ACU), a piece of AT&T hardware for dialing modem lines. The ACUs were system specific and allowed little flexibility to accommodate modems that are common today. With the release of system V, ACU hardware was replaced with software.

Communication ports that can be used by the **cu(1)** and **uucp(1)** utilities are defined in the `/usr/lib/uucp/L-devices` file. As described in the "UUCP Administration" chapter of the **Sys5 Unix Administrator's Guide**, the third parameter of each definition has been changed to be a dial program to execute rather than the address of the ACU. Two programs are supplied in the Plexus release, one for a Racal-Vadic 3451 (`/usr/plx/dial`) and one for the internal modem on P/75 and P/95 systems (`/usr/plx/mdial`). A dialer program for Hayes has been available from Software Support for several years (`/usr/local/fhayes`).

Writing dialer programs is not an exercise for the impatient and requires significant knowledge of the modem and the system. A public domain program was recently circulated on *Usenet* that allows the programmer to issue high level commands to the dial program that drives the modem. This program significantly simplifies the process of defining a new modem to the system for outdial operation. The software includes a replacement for the **dial(3)** function so that it is easy to build outdial support into user written software. Plexus has no plans to modify `cu` or `uucp` to use this code. The author of this program is Gene Olson of Quest Research in Burnsville, Minnesota.

Installing the dialer software on a Plexus system is easy. The example below will set up a Hayes dialer and install the programs into the `/usr/local` directory. On your system follow these steps:

1. Get `dialer.tar.Z` off the bulletin board and uncompress it into `dialer.tar`.

2. Enter:

```
tar xvf dialer.tar
```

to extract the files.

3. Enter:

```
make doc|lp
```

to print the documentation.

4. If you do not use `/usr/local` as your public executable directory, modify the **DESTDIR** variable in `Makefile` to be the directory where you want the software to be loaded and then continue.

5. As *root*, enter:

```
make install
```

to install the package.

6. Modify ACU definitions in `/usr/lib/uucp/L-devices` to use `/usr/local/fhayes`. A sample line will look like this:

```
ACU tty7 /usr/local/fhayes 1200
```

This definition will use `/dev/tty7` as an outdial line at 1200 baud with `/usr/local/fhayes` as the dial program. `/usr/local/fhayes` looks like this:

```
exec /usr/local/dialer -n -l 'basename $1' -s $2 $3
```

At this point, `cu` and `uucp` should work just fine. If you have modems that are not defined in the dialer software `dialinfo` file and you develop scripts for them, please send them in and we will publish them here.

Tips and Techniques for Your Plexus System & Software

compiled by the Staff at Plexus Software Support

1. A helpful hint for modems on ACP ports:

When you set up modem lines on any port, you usually have a line in */etc/rc* that looks something like the following:

```
stty -clocal < /dev/tty##
```

where **##** is the port number that the modem is on. When the port is opened, it will be opened with the parameters corresponding to the line in */etc/gettydefs* that matches the entry for that port in */etc/inittab*. One of the parameters that you want is **HUPCL** so that if the line is actually hung up, then all processes associated with that port will automatically die. This doesn't always happen though, which can leave the port hung. A simple trick that seems to fix this is to put the following line in */etc/rc*:

```
stty -clocal hupcl < /dev/tty##
```

This will ensure that the port retains the **HUPCL** parameter and makes for cleaner exits with modem ports.

2. How can I improve the performance of my cartridge tape drive?

For all machines capable of running Sys 5 release 1.7 (except P/90 & P/95), cartridge drive performance can be significantly improved by using */dev/rft* instead of */dev/rpt* for tape operations. The *rft* driver caches to memory in blocks of 128K which writes and reads the tape in a streaming mode instead of "shoe shining" in smaller tape blocks.

```
ls filenames | cpio -oBv > /dev/rft/0m
```

```
cpio -iBv < /dev/rft/0m
```

As with */dev/rpt*, you would use *0m* to rewind tape when complete and *0mn* to prevent rewinding.

The Plexus supplied tape commands that work for the cartridge drive do not recognize *rft* as a tape device. For those commands it is necessary to refer to */dev/rpt/0m(n)* when rewinding or retensioning.

```
/usr/plx/tape -f /dev/rpt/0m rew
```

For the P/90 and P/95, caching is already implemented in the *rpt* driver so the *rft* driver does not exist on those machines.

3. Is there an easier way to reset hung ports other than rebooting the machine?

You bet there is!! For ACP's, kill all the processes that are associated with the port and then use the **flush** option for **stty**. It should look something like this:

```
stty -f /dev/tty## flush & (where ## is the port number)
```

You will want to run the command in the background in case there are processes connected with the port that did not die. If you don't, and there are processes hanging around, then the **stty** command will hang also.

If you have VCP's, then you will again kill all of the processes on the port, then do the following **stty** commands:

```
stty -f /dev/tty## flush
```

```
stty -f /dev/tty## reset
```

NOTE: The **reset** command usually isn't necessary for ACP's, but can be used if just the **flush** doesn't work.

Killing all the processes associated with a port usually involves changing the **respawn** in */etc/inittab* to **off** for that port, doing a **telinit q**, then doing the **stty** command(s), changing the **off** back to a **respawn**, and then doing a **telinit q** again. Sometimes when you kill the processes on a hung port, the port will stay hung because a **getty** has already started up on the port, which disables you from clearing it with the **stty**, which makes the above steps necessary.

Following these procedures will clear most of the hung ports that you run into, so you should have to reboot much less often.

4. *What's a quick and easy way to restore stty settings to my tty port without specifying each stty parameter individually?*

This will be done differently, depending on the shell you're using. If you're using the C shell, put this alias definition as the last line in your *.login* file:

```
alias reset stty 'stty -g'
```

The command **stty -g** produces a string representing the current **stty** settings that can be used as an argument to other **stty** commands. By backquoting the command in your *.login* you are executing the command **stty -g** immediately and assigning the results to the command **reset**. When your **tty** settings become messed up due to abnormal termination of application programs or scripts, etc., typing in **reset** will return your **tty** to the settings you had when you first logged in.

If you're using the Bourne shell, put this command into your *.profile* file:

```
GTTY='stty -g'  
reset () stty $GTTY
```

To reset your **tty** settings, type in **reset**.

And as long as we're on the subject of backquotes...

5. *How exactly do these backquotes work?*

The purpose of backquotes is to take the output from the command(s) executed within the backquotes and use this as input to another command, or assignment to a variable. Referring to the previous example, specifically the

```
alias reset stty 'stty -g'
```

command, the command in the backquotes gets executed first, which produces output similar to the following:

```
536:5:3cbd:3b:7f:1c:8:18:4:0:0:0
```

So our line now looks like this:

```
alias reset stty 536:5:3cbd:3b:7f:1c:8:18:4:0:0:0
```

Then the variable **reset** is set to be the string

```
stty 536:5:3cbd:3b:7f:1c:8:18:4:0:0:0
```

Thus, every time you type **reset**, what you actually execute is the above command, which will reset your terminal.

Here is another backquotes example that could be used to print out the date, either at the command line or in a shell script.

```
echo "The date is `date`"
```

The double quotes (") prevent expansion of shell metacharacters, (except for the backquote (`), \$, and in certain cases the backslash (\)); and the **'date'** string is actually replaced by the output of the **date(1)** command to print out the string **The date is** , followed by the actual date.

A last example uses the assignment of shell variables. In the bourne shell, if **k** has a predefined value, and we say:

```
k=`expr $k + 1`
```

then **k** will be set to the output of the executed command within the backquotes. In this case the command is **expr** which evaluates expressions, the expression being **\$k + 1**, so we increment **k** by 1.

The best way to really get a feel for how backquotes (and the other quotes in the shell) work is to practice with each to see what specific action(s) they produce. They can be tricky at first, but once you get used to the way they work, it will become second nature to you to use them to your advantage in shell programming.

Using fsdb To Repair Damaged File Systems

by Mickey Mestel

In the summer newsletter, the feature article described the **fsck(1)** utility, which is used to periodically check and perform limited maintenance on file systems. There are times, though, that the file system is damaged to the point that **fsck** cannot repair it; or, that the problem has to do with something that **fsck** cannot correct, such as changing the contents of a file. This is where **fsdb(1m)** (file system debugger) comes into play. With **fsdb** you can, relatively easily, fix a damaged file system in a matter of a few keystrokes, saving hours of time over the only other alternative: restoring the entire file system. You can also cause more harm than good with the same few keystrokes, if not done correctly.

The best possible way to learn to use **fsdb** is to practice on a file system created just for that purpose. This way you are free to experiment and make mistakes without risk of losing valuable data. If things do get corrupted, you can just **restore** the file system and practice some more. Unfortunately, not all of us have the extra room on disk to put aside a whole partition just to experiment with. The only other alternative is to perform necessary work on a live file system. If this is the case, extreme caution and care must be used when using **fsdb**, and you generally don't want to play around; just get in, do what must be done, and then get out.

One thing to remember when using **fsdb** is that any assignment that you make is made on that data object (i-node, byte address, etc...) which is presently current. You therefore need to double check where you are before you make any assignments. Also, assignments are written to disk as soon as they are made, so there is no margin for error here. If you know what you have done, and it is wrong, it can usually be undone; but zeroing out the i-node number for a directory can take hours to fix, if it can be fixed at all.

The intent of this article is to cover some of the most common problems encountered with file systems which can be fixed by **fsdb**. It is beyond the scope of this article to cover all the different combinations of commands possible in **fsdb**, so we will work with specific examples and problems. Again, if you have never used **fsdb** and are faced with what appears to be the need to use it, call Plexus Software Support first, before attempting the work on your own. As powerful and useful as **fsdb** can be, it can also be equally destructive.

Our first scenario is one in which we need to change the contents of a file. A good example is: somehow the password file has been corrupted. In order for the system to boot, the password file must be present, and it must have root as the first entry with a correct path to a shell. Two common ways that we have seen the password file become corrupt are: either the file has a newline as the first character in the file, (It has been said that if the first character in the password file is a newline, then anyone can log in as root. What it really does is deny anyone access to the system!!) or the file has a bad path name to a shell for the root entry.

If you had a corrupted password file and you were at the boot prompt, there would be no way for you to boot the machine (and correct the file). The only alternative (yech!!) would be to restore the root file system!

What could be done instead is to call up **fsdb** from the release tape and use it to

patch the password file. The first thing to do is find the inode number of the password file itself. Since the password file is on the root file system, we need to give **fsdb** the name of the device where root resides, usually */dev/dsk/0s1*. We then start by listing the contents of the root directory using:

2i.fd

The following is a detailed description of this command:

- **2i** - access i-node number two of the filesystem. Note that i-node two is always the root inode of any file system. It is the only i-node in the list that has any special meaning; i-node number one is not used at this time and all the other i-nodes are simply assigned as needed.
- **."** - the dot is just a separator and is ignored by the command parser in fsdb. It is just there to break up the commands and make them a little less cryptic (hah!).
- **fd** - **f** is the general print function telling fsdb that the object we are currently specifying will be printed out in some format. The **d** tells fsdb that we want to print out the contents of this object as directory entries.

The output will look something like this (make sure you get ready to stop the output or it will scroll off the screen!):

```
d0:    2  .
d1:    2  . .
d2:    3  . p r o f i l e
d3:    4  b i n
d4:    5  d e v
d5:    6  e t c
d6:  3081 u n i x
d7:    8  t m p
d8:    9  u s r
d9:   225 l i b
d10:  451 l o s t + f o u n d
d11:  453 s t a n d
.
.
.
d64:   0
```

Each directory entry is assigned a number, as shown above. You can do operations on directory entries directly, as we will go into later. For now, what we need is the i-node number for the */etc* directory (where the password file resides). We note that the i-node number for */etc* is 6, so we use the following command to display the contents of the */etc* directory:

6i.fd

If we don't find the password file in the entries listed by this command, it's because the print command only works with one block of the file at a time. So we need to look at the next block of data. To do this, type the command:

6i.f1d

The **1** after the **f** says to use data block number 1. The first command, **6i.fd**, without a numeric argument after the **f**, assumes block number 0.

After whichever command lists the password file within */etc*, (the first command almost always does it), we'll see something like this:

```
d0:      6  .
d1:      2  . .
d2:    123 b c h e c k r c
d3:    124 b r c
d4:    267 c h e c k l i s t
d5:    126 c r a s h
d6:    128 p a s s w d
d7:    131 p a s s w d . o l d
.
.
.
d64:  344 r m d i r
```

Now we see that the i-node number of the password file is 128. We need to see the information associated with i-node 128, so we do:

128i

and this will give us output that looks like the following. Note that these numbers will differ for every system:

```
i#:  128 md: f---rwxrwxr-x ln:   1 uid:   0 gid:   2 sz:  5906
a0: 2954 a1: 2961 a2: 2968 a3: 2975 a4: 2982 a5: 2989 a6:  0
a7:   0 a8:  0 a9:  0 a10:  0 a11:   0 a12:   0
at: Tue Jan 12 14:51:52 1988
mt: Tue Jun 30 12:00:00 1987
ct: Wed Sep  2 10:48:14 1987
```

This tells us all the information contained in i-node 128. We see the i-node number, the mode of the file with a slot representation for every possible mode, setuid, setgid, etc... It also shows the number of links to the file, the owner, group and size of the file, but note that it also shows us the data blocks on disk that make up this file. The **2954** in the **a0** field is the first data block of the password file.

Now that we know where the file is, let's take a look at the file itself. To do this, type the command:

128i.fc

In the above command, the **fc** says to print out the first block of i-node 128 in character format. We should then see something like this:

```

13424000: \n r o o t : d f ' R s T m u 4 3
13424020: 8 u : 0 : 3 : r o o t l o g i
13424040: n : / : / b i n / c s h \n m i k
13424060: e f : 2 , i R f M 3 2 e e . 8 :
13424080: 3 : 4 0 8 0 : M i c h a e l S
.
.
.
13426000: c k m : / u s r / p l x / c s h

```

What we see displayed is the first 1024 characters in the password file. Remember, this is the first block of data from the file. The above display, as well as the list of data blocks shown in the inode, indicates that the password file is larger than 1 block. The numbers on the left are the actual byte count offset into the disk of each byte of data. Note that these numbers are in octal.

The file is displayed with 16 characters per line, hence the offset numbers increment by 20 (octal) each line. Looking at the first line, we see the newline character that is preventing us from booting the system. We need to get rid of it, but we can't edit the file as we would with a regular text editor. (This is possible to a small degree with **fsdb**, but it can be quite painful and time consuming.) What we want to do is replace the newline character with something that will let the boot prompts skip over it and continue on to find the shell that we need to run when we bring the system up.

We also see another problem here - the shell that is specified for root. It is **/bin/csh**, which doesn't exist, (it should be **/usr/plx/csh**), so even after we fix the first newline, we will have to correct the shell entry so the system will be able to find a shell to run when it boots up.

To continue, we need the value from the **a0** field of the i-node. Remember, that is the block number on disk of the first block of the file. We can then figure out exactly where the character we need to work with is located in the file, then we can change things from there. There are two ways to go about this: one is to calculate the exact byte offset of the first byte we want to change, the other is to use the block address as the beginning offset and start from there. For both methods we need the value in the **a0** field of the inode. In this case it is 2954, so to find the offset to the first byte in the file we simply multiply 2954 by 1024, the size of blocks for the file system. This value comes out to 3024896, which is the address of the first byte of the file. We can then issue the following command to **fsdb**:

```
3024896B="root:"
```

The **B** indicates the previous number should be interpreted as a byte address, and the string following the "=" is a simple assignment. This will leave our password file with two colons after **root** (see example below), but we can correct that after we boot the system.

```

13424000: r o o t : : d f ' R s T m u 4 3
13424020: 8 u : 0 : 3 : r o o t l o g i
13424040: n : / : / b i n / c s h \n m i k
13424060: e f : 2 , i R f M 3 2 e e . 8 :
13424080: 3 : 4 0 8 0 : M i c h a e l S
.
.
.
13426000: c k m : / u s r / p l x / c s h

```

The second method is to use the block address directly:

```
2954b="root:"
```

In this command, the **b** says to interpret the previous number as a block address, then do the assignment. This works well because the bytes we are changing are at the beginning of the block.

Now we need to change the string for the incorrect shell. This requires counting the number of bytes into the file where the shell is specified, then making the correction. The procedure will vary, depending on whether we use the byte offset or the block address method.

If we count from the first newline, that position being the zeroth position, the first / of the path for the shell is 36 characters in from the beginning. Adding this 36 to the original 3024896 value we get 3024932. In the case of the byte offset command we would type:

```
3024932B="/bin/sh\n" or,
```

```
3024932B="/usr/plx/csh\n"
```

Make sure the final newline is included or the problem won't be corrected. The system looks for a newline character to indicate the end of the string. Without the newline, the system would just continue on with the next character, whatever it is, and treat the entire string as a shell to be used. Chances are there is no such shell.

To use the block offset command method, we would type the following:

```
2954bB+36="/bin/sh\n" or,
```

```
2954bB+36="/usr/plx/csh\n"
```

Here we see a combination of both the **b** for block address conversion and the **B** for byte conversion. The **b** is used for the block address, and the **B** is used for the **+36**. The **B** says to interpret the **+36** as a byte offset into the file (the default is word offset). In this case, without the **B**, we would actually have inserted the string 36 words, or 72 bytes, into the file. Once again, watch the newlines.

At this point, we have sufficiently corrected the password file so that we can boot the machine. It still needs further correction, as we now have things like an extra ":" after root, but we can boot the machine and then make these changes.

Now that we have gone through all the work to modify the password file, if we look at the listing of i-node 2 printed as directory entries, we see there is a file called **passwd.old** in slot **d7**. If the current password file were corrupted and we were lucky enough to have a backup copy of an old password file available (ALWAYS a good idea to backup the original file!), then we have a simpler, faster recourse. (This is assuming, of course, that **passwd.old** is usable.) As I mentioned earlier, **fsdb** allows you to work directly with directory entries as data items. In this case, to restore our old password file, all we would need to do is the following:

```
d6.nm="passwd.old"
```

```
d7.nm="passwd"
```

The **nm** mnemonic is for the name field in the directory entry. This will simply swap the names in the **/etc** directory so that on boot, the old version of the password file will be found. Note that it doesn't matter what state the old copy is in, *as long as it has a good entry for root*. We're assuming that the backup file is usable, otherwise we would have to manually patch the file, as we did in the previous example.

Another situation where it is necessary to manipulate directory entries is if the root file system has used 100% of the space available to it and the system will not boot. What we would need to do is go in with **fsdb** and free up some space by zeroing out 1 or more inodes (which will, for all practical purposes, delete the files). Since we are not going through the usual Unix file system management routines, however, zeroing out an inode number does that and only that. The data blocks associated with that inode will not be put on the free list and will thus be in a sort of limbo out on the disk. We will need to run **fsck** and let it find the inode as an unreferenced inode and then put the blocks on the free list.

The first thing we need to do is to decide which files we are going to delete. Obviously, some discretion needs to be used. Removing **/etc/inittab** or **/unix** to clear up some space won't do much in the way of solving our problem. On the other hand, removing a file that a user has been working on for two weeks will certainly get the system up and going, but it might not do much for your physical well-being once said user finds out what happened to his/her work!

Possibly, all the space was used on the file system because the **/tmp** directory was being used extensively and took up much of the space available to the root file system. This is the best situation because we can delete tmp files without worrying too much about content and such; after all, they were in the **tmp** directory! If this is not the case - there are no files in **/tmp** that can be deleted - then we can look around a bit and see if possibly there are any leftover **core** files. However, this can be time consuming since we are down at the boot prompt. A good file to remove if we don't see anything obvious is **/etc/wtmp**. While it contains useful system accounting information, most people don't use it and it is certainly not crippling to lose it.

There are a couple of ways to look around the file system from the boot prompt, and **fsdb** is certainly one of them. The other is a standalone version of **ls**. Both of these are present on the boot tape AND in the **/stand** directory. The commands for standalone **ls** are the same as the normal **ls**, and it can be invoked from the boot prompt using:

```
:/stand/ls - from /stand directory on disk
```

: ls - from release tape.

You can invoke **fsdb** using these same methods.

We will go through the process of removing one file here. The process for removing two or more files would be the same. Once we know which file(s) to remove, we need to call up **fsdb**, (if we are not already in it). Let's say we are going to remove **/etc/wtmp**. We start with the command:

2i.fd

and get the inode number of the **/etc** directory. Take a look at the example listing for the root file system back on the 2nd page. According to this, the **/etc** directory has inode number **6**. We then issue the command:

6i.fd

and look for the **wtmp** file and get its **directory entry** number. There is an example on back on the third page. Let's say that the entry for **wtmp** is **d14**. To zero out the inode, all we have to do is issue the command:

d14=0

We will see a confirmation from **fsdb** that will look similar to the following:

d14: 0 w t m p

Remember that **fsdb** works with the current data object, so it is important that after the **6i.fd** command we don't do any other commands before we zero out the inode number. If, for instance, we wanted to look at the **wtmp** file to see how big it was, and did a **2234i**, (if that is the inode number for **wtmp**), we would have changed our current data object to be the **wtmp** file, instead of the directory table. If we then did the **d14=0** command, we would be attempting to interpret the first block of the **wtmp** file as directory entries, and then zero out the 14th entry, which makes no sense here, as the data in the **wtmp** file does not correspond to the way data is set up for directory entries. To reset the current data object, we would need to issue the command:

6i.fd

or just

6i

to make the inode for the **/etc** directory current again before we zeroed out the inode.

This finishes our work with **fsdb**. Now we need to run standalone **fsck** and let it clean up the file system. What it will find is an inode that appears to be allocated, but has a number of zero. (Normally, when a file is removed in Unix, the inode is pretty much zeroed out.) **fsck** will find this inode and also several data blocks which aren't on the free list and yet aren't tied to an allocated inode. It will ask if we want to **RECONNECT?** the file, and we need to answer **no** here. It will ask if we want to **CLEAR?** this inode, and it is important to answer **yes**, as this will actually return use of these blocks to the file system.

The follow up to this is to figure out exactly why the file system ran out of space. Some *Spring 1988*

administrative duties need to be performed, ranging from using `/etc/rc` to clean up the `/tmp` directory every time the system is booted to multi-user, to increasing the size of the file system.

The above constitutes a good overview of the most common uses for **fsdb** in repairing file systems. As was stated above, the commands can get very complex and there are sometimes many ways to accomplish the same task. For instance, you can look at a file using the block numbers for each specific block, or you can make the inode for that file current and look at each block with:

`fXc` where X is the block number starting with 0 (or blank)

It is relatively easy to step through a file and look at it this way, much easier than specifying one block number at a time.

Once again, be extremely careful when using **fsdb**. You can really cause some damage with this tool. Unless you have previous experience with it or feel you have enough background to handle using it, call Plexus Software Support before you attempt to fix something. If you have a situation exactly like one of the above, then these steps will certainly get you through it, but they won't help much if you make any mistakes, so do be careful.

As far as other sources of information, besides the **Sys5 UNIX Administrator's Reference Manual**, the best documentation I've seen for **fsdb** is **UNIX System Administration** by David Fiedler and Bruce H. Hunter, Hayden Books. They go through patching parts of the superblock of a file system, and their explanation is very good. For that reason, and the fact that I've never seen the instance they describe come up, (although it is quite possible), I have not included it in this article.

Efficient UUCP File Transfers

by John Maddox

When transferring files with **uucp(1C)**, the transfer can be made more efficient by combining all the smaller files into one large file, packing this file, and sending the packed copy. The advantages are: only one file is being sent, therefore the thrashing that is involved with determining if permissions are available for writing the file to the slave or the remote is reduced; also, a packed file occupies less space than the original unpacked version.

However, there are several reasons why this file must not be too large. The first has to do with time. Since (most of us) are transferring data over 1200 baud modem lines, files that are too large are going to take much too long to transfer and will run up nice phone bills. A 500k file could theoretically take somewhere around 55 minutes to transfer, but when the overhead of **uucp** and things like retransmission of bad packets is taken into account, we're realistically looking at an hour or more.

There are two inherent problems with transferring large files. The first is that after an hour, the lock file that **uucp** sets up is ignored. After this amount of time, it is assumed that the lock file was not deleted for some reason, and hence another **uucp** process could be started up and interfere with the first, which would completely garble the transmission. The second problem is if a bad packet of information is transferred and the receiving process can't get a good copy of it after a certain number of retries, the process will die. You are then stuck with transferring the whole file again. If the file being transferred is small, this is not such a problem, as it might take only a few seconds or minutes to transfer it again. What if, however, we were transferring our 500k file and were 50 minutes into the transmission, then the process died? We'll have used 50 minutes of time and phone cost, and we will have to start the transmission over again. Remember also that noisy lines can get worse the longer you are on them, so this can also be detrimental to transmission of large files.

On the average, keeping files below 100k is a very good idea, and basically, the smaller the better. It's only going to take a couple extra minutes to set up two or three files for transmission instead of one, and this is really the safest way to go.

For accuracy, checksums for each file should be generated and included in the large transmission file so that the individual files can be verified after transmission to the remote site. The program **sum(1)** can be used to generate this file.

The first step in this process is determining the UNIX metacharacter(s) that will get the proper file names from the shell. For example, consider a directory containing several files - those that end with **.1** and those that end with **.2**. Suppose that you wanted to transmit all files ending with **.2** to a node called *remote*. The most efficient method, while insuring accuracy, is to follow these steps:

1. Generate checksums for all files.

```
$ sum *.2 >SUMS.2
```

This places the checksum for all files ending with **.2** into a file named *SUMS.2*. This filename was chosen for convenience later when creating the single transmission file.

2. Create a single file to transmit.

This can be done using either **tar(1)** or **cpio(1)** - both work equally well. I will give examples using both commands.

tar

```
$ tar cvf XMIT *.2
```

This command creates a file called *XMIT* and places into it all files that end with *.2*. Of course *XMIT* (or whatever name you may choose) must NOT be present in the directory you are working from.

cpio

```
$ find . -name '*.2' -print | cpio -oBv >XMIT
```

This command creates a file called *XMIT* and places into it all files that end with *.2*. Of course *XMIT* (or whatever name you may choose) must NOT be present in the directory you are working from.

3. Pack the single transmission file.

pack(1) is a command that can be used to reduce the physical size of a file. The file cannot normally be used until it is unpacked, but packing does not damage the file.

To pack your *XMIT* file, key in:

```
$ pack XMIT
```

This will pack the file and place the result in *XMIT.z*. It will also return a percentage amount of compression that was achieved.

NOTE:

pack(1) and **unpack(1)** are Sys5 commands. For Berkeley systems, you can use **compress** and **uncompress**. These can be obtained from the bulletin board.

4. Send file.

Use your normal uucp sequence for transmitting the file. In our case:

```
$ uucp -nroot XMIT.z remote! /usr/spool/uucppublic
```

The previous command spools *XMIT.z* (our packed tar or cpio file) for transmission to node *remote*, to be placed in the */usr/spool/uucppublic* directory. The *-n* option notifies user *root* on *remote* that our file has arrived.

5. Unpacking on remote.

You must now unpack the file. This is achieved by changing to the directory where the file is located and using the **unpack(1)** command:

```
$ unpack XMIT.z
```

6. Extract individual files.

Wait for *unpack* to return *XMIT* unpacked before proceeding. Depending on your

choice of command for creating *XMIT*, use the correct example below to extract the individual files:

```
tar      =      $ tar xvf XMIT
```

```
cpio     =      $ cpio -idumv < XMIT
```

7. Generate checksums to verify accuracy.

Use the following command to create a checksum file to compare with the original checksum file and verify accurate transmission:

```
$ sum *.2 >REMOTE.SUMS
```

Compare the values in *REMOTE.SUMS* with those contained in *SUMS.2*. If a file has a different sum, that file must be retransmitted. If the sum is the same, the file was transmitted without errors.

8. Things to watch out for:

Sys3 checksums may vary from Sys5. This method only valid for Sys3 to Sys3 or Sys5 to Sys5 transfers.

Software Limitations in UNIX S/M/K1.7

Following is a list of known problems in version S/M/K1.7 of the Sys5 operating system:

/bin/adb	adb does not treat numbers input 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.
/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 be sure 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. Note that lint libraries must be built by the system administrator with the -o option.
/usr/lib/uucp/uuxqt	uuxqt will not correctly restart communications pending for the uux requester, leaving the X. file in the spool directory.
/usr/bin/ex	If you attempt to use ex or vi on a text file that has a pattern <i>ei:string:</i> or <i>ex:string:</i> in the first five or last five lines, vi will attempt to interpret the string as an editor command. Thus, a file with the string <i>ex:q:</i> can never be edited with vi , as the <i>q</i> causes vi to quit. Use ed instead.
/etc/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.
/usr/plx/tape	The 1.7 version of this command will not work with any version of /unix prior to Sys5 S/M/K1.7. The eraseall option fails to erase the tape on cache tape drives.
password aging	Login names of more than 7 characters cause password aging to fail. In addition, login names of other than alpha-numeric characters may cause problems with system accounting.
regex(3c)	The regex metacharacter "\$", used to indicate end-of-string, does not work.
/bin/tail	This utility will not report the number of lines correctly after line 81.
/bin/dd	It is advised that the conv=flip option be used when writing the tape out rather than when reading the

	tape in. An I/O error message will occasionally occur if you do the conversion while reading in the tape. The dd conv=sync option inserts nulls when small piped blocks are read by dd which expects 20Kb blocks. The meaning of "b" on buffer size is now 512Kb rather than its previous value of 1024Kb. This conforms to the SVID. If you are currently using "b", change it to "k". This problem is mostly noticed when using reel-to-reel tapes.
/usr/bin/cu	Redials (re-executes) the dial program after successful exit from cu . This occurs when tee is executed. Redials after "" when cu is initialized under " sh tee file ".
/usr/bin/vi	Dumps core or exits when it encounters an unknown terminal type.
/etc/termcap	Does not contain Wyse-75 entries.
/usr/bin/vi	Pages backward two lines too far.
/bin/cc	Greenhills "C" 1.8.1 evaluates expressions differently from PCC and K & R definition.
/bin/mailx	mailx permits a text line that contains only a dot to be sent, even though this will always terminate the message when piped to /bin/mail .
/usr/plx/tape	The eraseall option fails to release reel-to-reel tape on caching tape drive. Program hangs when invoked.
td	mm firmware doing td(0,0) for memory dumps does not put eof mark at end of file. Additional files cannot be written out to the cartridge.

Problems Fixed In This Release

Following is a list of the problems fixed in S/M/K1.7:

/etc/dd	The size specification "b" is now set to 512.
/bin/mail	Creates a unique temporary file name /tmp/maXXXXXX . Previously, the temporary file name had only five X's.
/etc/mount	Mount reports correct device name. Previously it put an incorrect name of the block device in mnttab .
/etc/ttytype	This file is now included in the release tape.
/usr/lib/uucp/uusub	The -r option now reports send and receive counts correctly matching its headings. Previously the value was mistakenly swapped.
/etc/dump	Creates the file /etc/ddate if it does not exist and you are making a full dump.
setvbuf()	The parameters now match SVID. It is now setvbuf(stream, buf, type, size) . Previously it was setvbuf(stream, type, buf, size) .

<code>longjmp()</code>	Fixed longjmp to never return zero.
<code>/bin/od</code>	Previously this would crash if you tried to create a partition on a full optical disk.
<code>/stand/fbackup</code>	fbackup can now access multiple disk controllers. Previously, it would not back up any drive beyond number three.

ACP Problems Fixed In S/M/K1.7

- The problem with the ACP port hanging in several situations has been fixed. In some cases you may need to kill all the processes associated with that port and use the **stty flush** command to clear a line that has hung due to excessive noise. Make sure you put the **stty flush** in the background in case there are any processes that haven't died.
- The problem with interrupt sharing in the NOS ethernet driver has been fixed.

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`.

2. There is a problem that sometimes occurs when a pseudo-tty is closed during a remote session. The problem seems to occur as the shell that is on that port does not complete the flushing of its data before the socket closes, and this can hang the port. There is a quick and simple solution to this problem, and that is to put a `sleep 2` in the `.logout` for the `cs`, and something similar to the following in the `.profile` for `sh`:

```
LOGOUT () { sleep 2 }
```

```
trap "LOGOUT" 0
```

Of course, you can get more creative in your LOGOUT function, such as clearing your screen and displaying a message, but this will take care of the TCP/IP problem.

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 `addrec` with an existing key will succeed, but will result in btree problems on that record. Always test using `acckey` to determine if a record with that key already exists, then call `addrec` 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 -OI<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".

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

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-Z8000-9-track	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

COMMUNICATIONS				
Part #	Model	Systems	Software Product	Current Release
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
RJE/HASP				
67-00319	4211AC	P/35	68000-Sys3-cartridge	3.1
67-00320	4211AT	P/60	9-track	3.1
67-00628	4311AT	P/60-75	68000/20-Sys5 - 9-track	4.0
67-00629	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				
67-00876	4371AT	P/60-75	68000/20-Sys5 - 9-track	3.2Q-2.1
67-00877	4371AC	P/35-55-75		
67-00833	4371DC	P/20	68010-Sys5-cartridge	3.2
67-00878	4371CT	P/95	68020-Sys5 - 9-track	3.2Q-2.1
67-00879	4371CC	P/95-90	cartridge	3.2Q-2.1
Multiplex				
67-00699	4374DF	P/15-20	68010-IBM PC floppy disk	1.1
67-00730	4374AC	P/35-55-75-95-90	68000/20-Sys5-cartridge	1.1
67-00731	4374AT	P/60-75-95	9-track	1.1
67-00732	4374DC	P/15-20	68010-Sys5-cartridge	1.1
3270 SNA				
67-00972	4378AT	P/60-75	mSNA/3270 - 9 track	2.1
67-00973	4378AC	P/35-55-75	cartridge	2.1
67-00972	4378CT	P/95	mSNA/3270 - 9 track	2.1
67-00973	4378CC	P/90-95	cartridge	2.1

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
67-00897	4318DC	P/15-20	68010-Sys5-cartridge	3.2C
67-00897	4318AC	P/35	68000/20-Sys5-cartridge	3.2C
67-00897	4318BC	P/55-90	cartridge	3.2C
67-00897	4318GC	P/75	cartridge	3.2C
67-00897	4318CC	P/95	cartridge	3.2C
67-00902	4318FT	P/60	9-track	3.2C
67-00902	4318GT	P/75	9-track	3.2C
67-00902	4318CT	P/95	9-track	3.2C
67-00905	4319DC	P/15-20	68010-Runtime cartridge	3.2C
67-00905	4319AC	P/35	68000/20-Runtime cartridge	3.2C
67-00905	4319BC	P/55-90	Runtime cartridge	3.2C
67-00905	4319GC	P/75	Runtime cartridge	3.2C
67-00910	4319FT	P/60	Runtime 9-track	3.2C
67-00910	4319GT	P/75	Runtime 9-track	3.2C
67-00910	4319CT	P/95	Runtime 9-track	3.2C
67-00702	P/35-55-75-90-95		Unify-RM/Cobol Interface-cartridge	
67-00703	P/60-75-95		9-track	
67-00704	P/15-20		cartridge	
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/75-95	"	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/55-90	"	1.1
67-00919	4346CC	P/75-95	"	1.1
67-00923	4346BT	P/60	9-track	1.1
67-00923	4346CT	P/75-95	"	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/55-90	"	2.1
67-00925	4347CC	P/75-95	"	2.1
67-00929	4347BT	P/60	9-track	2.1
67-00929	4347CT	P/75-95	"	2.1
67-00925	4347DC	P/15-20	cartridge	2.1

Informix C-ISAM

67-00931	4350AC	P/35	C-ISAM-Sys5-cartridge	3.1
67-00931	4350BC	P/55-90	"	3.1
67-00931	4350CC	P/75-95	"	3.1
67-00935	4350BT	P/60	9-track	3.1
67-00931	4350CT	P/75-95	"	3.1
67-00931	4350DC	P/15-20	cartridge	3.1

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
67-00813	4349CT	P/75-95	"	3.00
67-00783	4349DC	P/15-20	cartridge	3.00

20/20

68-00423	4320AC	P/35-55-75-90-95	68000/20-Sys5-cartridge kit	1.13
68-00424	4320AT	P/60-75-95	9-track kit	1.13
68-00423	4320DC	P/15-20	68010-Sys5-cartridge kit	1.13

Q-Office

67-00564	4222AC	P/35	68000-Sys3-cartridge	1.94
67-00565	4222AT	P/60	9-track	1.94
67-01030	4322AC	P/35-55-75-90-95	68000/20-Sys5-cartridge	1.94i.3
67-01031	4322AT	P/60-75-90	9-track	1.94i.3
67-01032	4322DC	P/15-20	68010-Sys5-cartridge	1.94i.3

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/35-55-75-90-95	68000/20-Sys5-cartridge	1.94
67-00657	4321AT	P/60-75-95	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

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/35-55-75-90-95	68000/20-Sys5-cartridge	2.1A
67-00478	4304AT	P/60-75-95	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				
67-00662	4336AC	P/35-55-75-90-95	68000/20-Sys5-cartridge	1.32
67-00663	4336AT	P/60-75-95	9-track	1.32
67-00664	4336DC	P/15-20	68010-Sys5-cartridge	1.32
Philon FAST/BASIC-M				
67-00794	4337AC	P/35-55-75-90-95	68000/20-Sys5-cartridge	1.37
67-00795	4337AT	P/60-75-95	9-track	1.37
67-00796	4337DC	P/15-20	68010-Sys5-cartridge	1.37
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/35-55-75-90-95	68000/20-Sys5-cartridge	6.5.10
67-00708	4309AT	P/60-75-95	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/35-55-75-90-95	68000/20-Sys5-cartridge	6.1
67-00827	4323AT	P/60-75-95	9-track	6.1
67-00828	4323DC	P/15-20	68010-Sys5-cartridge	6.1
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/35-55-75-90-95	68020-Sys5-cartridge	2.6
67-00790	4326AT	P/60-75-95	9-track	2.6
67-00787	4327AC	P/35-55-75-90-95	68020 w/68881-cartridge	2.6
67-00788	4327AT	P/60-75-95	9-track	2.6
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/35-55-75-90-95	68000/20-Sys5-cartridge	2.6
67-00838	4315AT	P/60-75-95	9-track	2.6
67-00839	4315DC	P/15-20	68010-Sys5-cartridge	2.6
Greenhills C Compiler				
67-00745	4303AT	P/60-75-95	68000/20-Sys5-9-track	1.8.01
67-00746	4303AC	P/35-55-75-90-95	-cartridge	1.8.01
67-00747	4303DC	P/15-20	-cartridge	1.8.01

XDP-RELATED SOFTWARE

XDP DataServer Software				
Part #	Model	Systems	Software Product	Current Release
67-01020	4352BC	P/90	XDP DataManager SQL cartridge	1.0
67-01020	4352CC	P/75	XDP DataManager SQL cartridge	1.0
67-01020	4352DC	P/95	XDP DataManager SQL cartridge	1.0
67-01021	4352CT	P/75	XDP DataManager SQL 9-track	1.0
67-01021	4352DT	P/95	XDP DataManager SQL 9-track	1.0
67-01022	4358BC	P/90	XDP DataManager ESQL cartridge	1.0
67-01022	4358CC	P/75	XDP DataManager ESQL cartridge	1.0
67-01022	4358DC	P/95	XDP DataManager ESQL cartridge	1.0
67-01023	4358CT	P/75	XDP DataManager ESQL 9-track	1.0
67-01023	4358DT	P/95	XDP DataManager ESQL 9-track	1.0
67-01025	4362BC	P/90	XDP Media Manager cartridge	1.0
67-01025	4362CC	P/75	XDP Media Manager cartridge	1.0
67-01025	4362DC	P/95	XDP Media Manager cartridge	1.0
67-01024	4362CT	P/75	XDP Media Manager 9-track	1.0
67-01024	4362DT	P/95	XDP Media Manager 9-track	1.0
Standard Workstation Development Software				
Part #	Model	Systems	Software Product	Current Release
68-00471	S4410		XDP Developer Kit	1.0
Kit Components				
67-01009	S4347D		ESQL/C Development	2.11.3
67-01011	S4351D		AIP Library	4.0
68-00464	S4353D		MS-Windows Development Package	1.04
68-00463	S4353R		Microsoft Windows	1.04
68-00462	S4354		MS-C Compiler	4.0
68-00466	S4363		Socket Library, Telnet FTP	3.2.5
67-01013	S4353M		XDP Environment Install	1.0
68-00473	S4410U		XDP Developer UPDATE Kit	1.0
Kit Components				
67-01009	S4347D		ESQL/C Development	2.11.3
67-01011	S4351D		AIP Library	4.0
67-01013	S4353M		XDP Environment Install	1.0

Standard Workstation Runtime Software

Part #	Model	Systems	Software Product	Current Release
68-00472	S4420		XDP Runtime Kit	1.0
Kit Components				
67-01010	S4347R		ESQL/C Runtime License	2.11.3
67-01012	S4351R		AIP Runtime	4.0
68-00463	S4353R		Microsoft Windows	1.04
67-01013	S4353M		XDP Environment Install	1.0
<hr/>				
68-00474	S4420U		XDP Runtime UPDATE Kit	1.0
Kit Components				
67-01010	S4347R		ESQL/C Runtime License	2.11.3
67-01012	S4351R		AIP Runtime	4.0
67-01013	S4353M		XDP Environment Install	1.0

Standard Print Server Software

Part #	Model	Systems	Software Product	Current Release
67-01010	S4347R		ESQL/C Runtime License	2.11.3
67-01012	S4351R		AIP Runtime	4.0
68-00463	S4353R		Microsoft Windows	1.04
67-01014	S4361		Image Print Spooler	2.0

Optional Workstation Software

Part #	Model	Systems	Software Product	Current Release
67-01015-01	S4360D		XDP Application Designer #1	1.0
67-01015-02	S4360D		XDP Application Designer #2	1.0
67-01016	S4360R		XDP Application Designer Runtime	1.0
68-00467	S4364		TCP/IP, FTP, Telnet & TE	
68-00465	S4365		BRIEF Programmer's Editor	

PLEXUS COMPUTERS INC.
Current Documentation Releases

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
98-05133.2 - Ver. B	Sys5 Administrator's Handbook (includes P/90-95)
98-40199.7 - Ver. A	Sys5 - S/M/K 1.7 Release Notice
98-40318.1 - Ver. A	Sys5 - S/M/K 1.7 Update Notice

COMMUNICATION SOFTWARE DOCUMENTATION

Part #	Description
NOS	
98-05107.2 - Ver. D	NOS Users Manual for Sys5 (includes 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
RJE/HASP	
98-40201.1 - Ver. B	RJE/HASP 4.0 Release Notice
Batch 2780/3780	
98-40059.3	BATCH 2780/3780 3.2 Release Notice
98-40200.1	BATCH 2780/3780 4.0 Release Notice
TCP/IP	
98-40211.4	TCP/IP 3.2 Release Notice
98-05131.1	Excelan EXOS - 8010 Reference Manual
Multiplex	
98-05124.2	Multiplex User's Manual
98-40244.1	Multiplex Host Installation Guide
98-40245.1	Multiplex PC Installation Guide
98-30029.1	One Page Notice
3270 SNA	
98-40223.1 - Ver. A	mSNA/3270 User's Manual
98-40226.2 - Ver. C	mSNA/3270 2.0 Release Notice

APPLICATIONS DOCUMENTATION

Part #	Description
Unify	
98-40077.2 - Ver. A	Unify 3.1b Release Notice (for Sys3 or Sys5)
98-40161.1 - Ver. C	Unify 3.2C Release Notice (for Sys5 only)
98-30031.1 - Ver. A	Unify 3.2C Users Package Update
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	
98-05150.2	SQL Reference Manual
98-40263.2 - Ver. A	SQL Release Notice
98-05149.2	4GL 3-volume set (Reference Manuals & User Guide)
98-40265.2 - Ver. A	4GL Release Notice
98-05151.2	ESQL-C Reference Manual
98-40262.2 - Ver. A	ESQL-C Release Notice

98-05152.2	C-ISAM Reference Manual
98-40261.2 - Ver. A	C-ISAM Release Notice
98-05153.1	File-It! 3.00 Reference Manual
98-40264.1	File-It! 3.00 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
98-05099.4	Q-Office 1.94i.3 2 Book Set
98-05099.5 - Ver. A	Q-Office 1.94 Set (Qoffice 2 bk Set & Qone Tutorial)
98-40145.1 - Ver. C	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 and R:Desk)

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
98-40299.1	Word:Era Release Notice

LANGUAGE SOFTWARE DOCUMENTATION

Part #**Description****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)

Philon COBOL

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

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

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

CDB C-Debugger

98-40155.3 CDB 6.1 Release Notice

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)

Greenhills C Compiler

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

XDP DOCUMENTATION	
Part #	Description
DATA MANAGER - SQL	
Software Model Code 4352	
98-05194.1	INFORMIX-TURBO Manual Ver. 1.1
98-05150.2	INFORMIX-SQL Manual Ver. 2.1
98-05211.1	XDP DataManager Supplement to SQL
98-05201.1	XDP System Admin. Guide
98-05210.1	XDP System Error Messages
98-40326.1	DataManager-SQL & TURBO Release Notice
DATA MANAGER - ESQL/C	
Software Model Code 4358	
98-05194.1	INFORMIX-TURBO Manual Ver. 1.1
98-05151.2	INFORMIX-ESQL/C Manual Ver. 2.1
98-05211.1	XDP DataManager Supplement to SQL
98-05201.1	XDP System Admin. Guide
98-05210.1	XDP System Error Messages
98-40323.1	DataManager-ESQL/C & TURBO Release Notice
MEDIA MANAGER	
Software Model Code 4362	
98-40324.1	XDP MediaManager-DMU Release Notice
DEVELOPER'S KIT	
Software Model Code 4410	
98-05151.2	INFORMIX-ESQL/C Manual Ver. 2.1
98-05211.1	XDP DataManager Supplement to SQL
98-05191.1	XDP Programmer's Manual
98-05210.1	XDP System Error Messages
98-05204.1	XDP Workstation Software Installation
98-40357.1	XDP Developer's Kit Release Notice
RUNTIME KIT	
Software Model Code 4420	
98-05204.1	XDP Workstation Software Installation
98-40358.1	XDP Runtime Kit Release Notice
PRINT SPOOLER	
Software Model Code 4361	
98-05204.1	XDP Workstation Software Installation
98-40361.1	XDP Print Spooler Release Notice
APPLICATION DESIGNER DEVELOPMENT	
Software Model Code 4360D	
98-05199.1	XDP Application Designer Reference Manual Set
98-05210.1	XDP System Error Messages
98-40359.1	XDP Application Designer Development Release Notice
APPLICATION DESIGNER RUNTIME	
Software Model Code 4360R	
98-40360.1	XDP Application Designer Runtime Release Notice