alpha micro
AMED
journal
December 1987
ALPHA MICRO SERVICE DIVISION
Volume 9, No. 12

In this issue...

Special Articles

Hardware

Software

General Information

Back Issue Volume Table of Contents Pages

December 1987 **AMSD Journal** Volume 9, Number 12

Editor..... Robin Soto
Assistant Editor..... Sandra Tomlin
Artwork Director..... Bill Briscoe

Contributors to this issue: Carl Anderson, Randy Bird, Bob Currier, Brett Halle, RobHancox, Jeff Kreider, Nancy Steen, Mike Talbutt, Sandra Tomlin, Dave Yox, Mike Zappavigna

Copyright ©1987 by Alpha Microsystems.

AlphaACCOUNTING, AlphaBASIC, AlphaCALC, AlphaFORTRAN, AlphaLISP, Alpha Micro, AlphaPASCAL, Alpha-RJE, AlphaVUE, AlphaWRITE, AM-100, AMOS, and AMTEC are trademarks of Alpha Microsystems. VIDEO-TRAX is a registered trademark of Alpha Microsystems.

The *AMSD Journal* is published monthly by Alpha Micro Technical Publications for Alpha Micro Technical Services. Please address all correspondence to AMSD Journal, Alpha Microsystems, 3501 Sunflower, Santa Ana, California, 92704.

Alpha Micro has checked the information contained in this newsletter and believes it to be accurate at the time of publication. However, readers should independently determine that any information used works correctly on their system and is appropriate for their application.

Subscription Information: Subscription rates are $40 per year. Back issue sets are also available for $150 per set. Each Alpha Micro dealer receives one permanent subscription to the *AMSD Journal* free. For additional subscriptions, send your name, company name, address, customer number (if applicable) with payment of $40 for each subscription to:

AMSD Journal
Subscription Department
Alpha Microsystems
P.O. Box 25059
Santa Ana, CA 92799

AMSD Telephone Numbers

Customer Service	(714) 641-7602
Customer Education	(714) 641-6330
Technical Support	(714) 641-7608
Main Switchboard	(714) 957-8500
Spare Parts/Logistics	(714) 641-6339
AMTEC Information	(714) 641-7608

Article Filing Instructions

We hope you find the *AMSD Journal* to be a valuable reference tool, and that you will want to refer to its articles frequently in the future. To make it easy and quick to find information, current articles are designed to be filed with articles from past issues. The entire set of *Journal* back issues forms three volumes: "General Information," "Software Information," and "Hardware Information." (The set of back issues is available for purchase. See "Subscription Information," above.)

The title of each feature article in this issue includes a reference number. Use the reference number to file the article in the back issue volume indicated at the top of each page of the article. For example, if the top of the first page of the article "6.5.5 One Hundred New Uses for MULTI," contains the words "Software Information," you know that article is to be filed in Section 6 of the "Software Information" back issue volume after article number 6.5.4.

The last pages of the *Journal* are new Tables of Contents for the back issue volumes, updated with entries for articles included in this month's issue.

Holiday Schedule

Alpha Micro Wishes all our dealers and customers a Happy Holiday. Our Holiday Schedule is:

- **Thursday, 24 December 1987**
  Closing at noon.
- **Friday, 25 December 1987**
  Closed
- **Thursday, 31 December 1987**
  Closing at noon.
- **Friday, 1 January 1988**
  Closed

4.8.1
VIDEOTRAX Controller Notice: Jumper IR3

The full-size VIDEOTRAX Controller, labeled DWB-00616-xx, contains several factory-set jumpers that select controller functions.

Until recently, jumper IR3 (Figure 1) was installed at the factory. However, recent testing has shown us this jumper is not required and, in certain situations, can cause compatibility problems. Therefore, all full-size VIDEOTRAX Controllers are now configured at the factory **without** jumper IR3 installed.

Your VIDEOTRAX Controller Owner's Manual may instruct you to make sure jumper IR3 is installed. Please disregard that instruction and, if jumper IR3 is installed **remove it.**

This change affects only the full-size controller; the half-size VIDEOTRAX Controller, labeled DWB-00614-00, does not have the IR3 jumper.

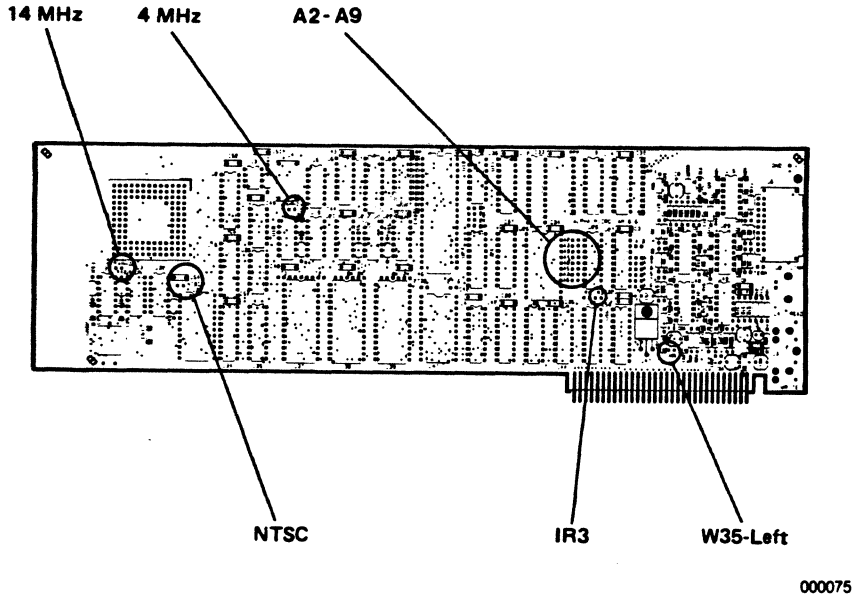

Figure 1
VIDEOTRAX Full-size Controller
Jumper Locations

5.7.1
AM-740 Memory Board - Compatibility Issues

Alpha Micro recently introduced the AM-740 VME Memory board. This board has three advantages over the AM-730 VME Memory board:

- The dynamic RAM (DRAM) chips used on the AM-740 board are 1 Mbit parts allowing for a greater level of memory density. The AM-740 is available in 8 Mbyte and 12 Mbyte versions, compared to a maximum of only 4 Mbytes on the AM-730.

- Nibble-mode dynamic RAM is used on the AM-740. While this gives no special benefit with the current generation 68020 CPU, Motorola has announced that the 68030 CPU will be able to use the additional data bandwidth which the nibble-mode parts can provide.

- The AM-740 has connections to make it possible to add an Error Checking and Correction (ECC) board to the system. Currently, however, we do not have such a design in process.

Revision Levels

In order to use an AM-740 board in an existing VME system, minimum revision levels for boot PROMs and the AMOS monitor are required.

Boot PROMs

In order to boot an AM-2000 system with an AM-740 board, the boot PROMs (DWP-00187-00, DWP-00188-00) for the AM-180 board must be:

Revision C00 or later.

On an AM-1500 system, the AM-175 board boot PROMs (DWP-00176-05, DWP-00176-06) must be:

Revision F00 or later.

If you have an older revision and need to update, you can order new boot PROM sets for the AM-2000 using part number PDB-00187-00, or for the AM-1500 using part number PDB-00176-00.

Symptoms produced by systems without the correct revision level boot PROMs include failure to boot, failure to "see" the AM-740 board, or inability to address the entire AM-740 memory array.

AMOS Monitor

The AMOS monitor (AMOS/L version 1.3C or AMOS/32 version 1.0A) must have the -2 patch installed. The symptom produced if an older monitor is present is periodic parity errors. The -2 patch enabled AMOS to "see" more than 8 Mbytes of memory, so installing it lets you obtain the full benefit of the AM-740, especially in the 12 Mbyte version.

7.3.1
DMA Bus Grant Signals for AM-1500 and AM-2000 Systems

By Michael Zappavigna,
Senior Support Specialist
Technical Support Group

This article defines the correct Bus Grant jumpers to remove on the VME motherboard for the AM-1500 and AM-2000 systems.

The Bus Grant signals, all four levels, are routed up the J1 side of the motherboard. On the 6-slot, 10-slot and 21-slot chassis, Bus Grant jumpers are designated BG0, BG1, BG2, BG3. However, some 10-slot chassis show the designations as BG1, BG2, BG3 and BG4.

When you install a DMA type circuit card--like the AM-350, AM-515, AM-520, or AM-640--into the corresponding connector pair, you must remove the IACK and either the BG3 or the BG4 shorting block from the motherboard at the position it will occupy. The shorting block pads are just above the slot the card will occupy on the J1 (left) side of the chassis.

If the shorting blocks are labeled BG0 through BG3, as they are on most of the motherboards remove BG3. If the shorting blocks are labeled BG1 through BG4, as they are on some 10 slot motherboards, remove BG4.

The difference between the BG labels does not affect the performance of the systems and was only a silk-screen change. This information will be noted in the next revision of the installation instructions for the DMA circuit cards. All new VME systems chassis are silk screened with the labels BG0, BG1, BG2, BG3.

3.2.31
New Software Patches Available

The following list gives a description of the new software patches now available from Alpha Micro's Technical Support Group. The products affected by these patches are: AMOS/PC, Unify, Cobol, AMOS/L, AMOS/32, and AlphaWRITE.

Patches in the following list include SPNs 322 through 350, and are current through 16 November 1987.

The SPN description in the purpose column ends with the software version(s) this patch is intended for.

SPN #	Module	Purpose
NOTE: 322-324	AMOS/PC Monitors	One SPN number was used for each user version of the AMOS/PC monitor. The problems resolved by SPNs 322 through 324 are listed below. - Corrects monitor call FILOTB operation when used over the network. - Fixes problems ith booting with AM-515 with moe than 8 Mb of memory. - Supports AM-740 memory board.
322	PCS004.MON	This patch applies to AMOS/PC 4-user limit and fixes the problems shown in NOTE above.
323	PCS008.MON	This patch applies to AMOS/PC 8-user limit and fixes the problems shown in NOTE above.
324	PCSUNL.MON	This patch applies to AMOS/PC unlimited-user limit and fixes the problems shown in NOTE above.
NOTE: 325-327	AMOS/PC Monitors	One SPN number was used for each user version of the AMOS/PC monitor. The problems resolved by SPNs 325 through 327 are listed below. - Corrects problem caused by TRMICP if no job was attached and function key processing was initiated. - Fixes problem where ".8-0" entered at monitor would crash the system. - Fixes problem caused by execution of EXIT monitor call with no memory available could crash the system.
325	PCS004.MON	This patch applies to AMOS/PC 4-user limit and fixes the problems shown in NOTE above.
326	PCS008.MON	This patch applies to AMOS/PC 8-user limit and fixes the problems shown in NOTE above.
327	PCSUNL.MON	This patch applies to AMOS/PC unlimited-user limit and fixes the problems shown in NOTE above.
328	–	Released in a previous month.
329	–	Released in a previous month.
330	–	Released in a previous month.
331	–	Released in a previous month.
332	DBMS.SYS	Removes restrictions on UNIFY/COBOL interface functions. This patch applies to Unify version 3.1.

(continued on next page)

3.2.31 (Continued)
New Software Patches Available

(continued from previous page)

SPN #	Module	Purpose
333	DBLOAD	Corrects problems caused by using DBLOAD with large numbers of records and BTREE indices. This patch applies to Unify version 3.1.
334	–	Released in a previous month.
335		Released in a previous month.
336	–	Released in a previous month.
NOTE: 337-340	COBOL	The problems resolved by SPNs 337 through 340 are listed below. - COBOL could crash with "." at end of display statement. - System would die if CTRL C was entered to PCOBOL. - Data errors when moving COMP-3 data items. - EXCBL with an invalid filename would crash job. - Arithmetic errors involving COMP-3.
337	PCOBOL	This patch applies to AlphaCOBOL version 1.2B and fixes the problems shown in NOTE above.
338	EXCBL	This patch applies to AlphaCOBOL version 1.2B and fixes the problems shown in NOTE above.
339	DBCBL	This patch applies to AlphaCOBOL version 1.2B and fixes the problems shown in NOTE above.
340	MOM.COB	This patch applies to AlphaCOBOL version 1.2B and fixes the problems shown in NOTE above.
341	–	Released in a previous month.
342	–	Released in a previous month.
343	–	Released in a previous month.
344	MTUSAV	Corrects a problem with MUTSAV hanging if used in partitions exceeding 8 Mb. Also, MTUSAV would report "Memory map destroyed" when used with fast disk drive/slow tape drive combinations (AM-520/Cipher.) This patch applies to AMOS/L version 1.3C and AMOS/32 version 1.0A.
NOTE: 345-348		SPNs 345 through 348 allow useof BADBLK on multiple AM-520 boards in a single system. These patches apply to AMOS/L version 1.3C and AMOS/32 version 1.0A.
345	BADBLK	See NOTE above.
346	520DVR.DVR	See NOTE above.
347	520DVR.RTD	See NOTE above.
348	CRT520	See NOTE above.
349	BAUD.LIT	Prevents privilege violation during execution of BAUD.LIT on an AM-1213 port. This patch applies only to AMOS/L 1.3C.
350	WRT200.OVR	Corrects a problem which sometimes caused a system crash if SHORT.DCT was in system memory during a spell check. This patch applies to AlphaWRITE 1.2A.

3.3.23
Software Verification and Support

by Jeff Kreider,
Senior Support Specialist
Technical Support Group

The Technical Support Group has long recognized that one of the most common software problems is mismatched software. Troubleshooting such a problem is very time consuming and, until recently, verifying the integrity of the software was a tedious task and the method itself lacked certainty.

Verification required generating a directory with hash totals and comparing it to the directory list of the system software as released. The validity of this check depended upon visual inspection of each correct hash and file name, leaving lots of room for error. Rather than spend the time to go through such a tedious exercise, it was more prudent to deal with "probabilities," "reasonable certainties" and "should be's" about the integrity of the software product.

As of AMOS/L Version 1.3, we provide a new utility with the System Software to simplify this process and greatly improve the reliability of the result. VERIFY is a command executed with a file name as an argument. The file specified should contain a specially formatted directory listing with hash totals and version numbers. Upon execution, VERIFY reads from this directory file each file specification, hash total and version number, and then looks at that file on the disk and calculates the hash total, reads the version number and compares this information with that in the directory listing.

The format of the file that VERIFY operates on is a directory list generated with the DIR/D/H/V command. The default extension is .DIR, and the default file name is AMOS. However, any file containing such a list will work. Here is a sample of VERIFY's output:

```
.VERIFY
DSK0:AMOS.LIT[1,4]          VERIFIED
DSK0:AMSORT.SYS[1,4]        VERIFIED
DSK0:AM334.LIT[1,4]         VERIFIED
DSK0:APPEND.LIT[1,4]        VERIFIED
DSK0:ASCDMP.LIT[1,4]        VERIFIED
DSK0:ATTACH.LIT[1,4]        VERIFIED
DSK0:BADBLK.LIT[1,4]        VERIFIED
DSK0:BASIC.LIT[1,4]         VERIFIED
DSK0:BAUD.LIT[1,4]          VERIFIED
DSK0:BITMAP.LIT[1,4]        VERIFIED
DSK0:CACHE.LIT[1,4]         VERIFIED
DSK0:ISAM.SYS[1,4]          VERIFIED
DSK0:CAL100.LIT[1,4]        VERIFIED
DSK0:CAL120.LIT[1,4]        VERIFIED
DSK0:CLEAR.LIT[1,4]         VERIFIED
DSK0:CMDLIN.SYS[1,4]        VERIFIED
DSK0:COMPIL.LIT[1,4]        ?Version mismatch is-1.3(204)-2 MDL=1.3(204)
DSK0:CONVRT.TBL[1,4]        VERIFIED

Exit
^C
.
```

3.3.23 (Continued)
Software Verification and Support

In the "?Version mismatch..." message, MDL is the acronym for Master Directory Listing, the data in the .DIR file.

If VERIFY is used on a file which cannot be found, you see:

```
.VERIFY MARVIN
MARVIN.DIR not found.
Exit
.
```

If executed on a file that is not a directory listing, you see:

```
.VERIFY BADBLK.SYS
?Invalid MDL format
Exit
.
```

Other discrepancies that are reported are: a file not found or an incorrect hash total:

```
DSKØ:PAUSE.LIT[1,4]     VERIFIED
DSKØ:PPN.LIT[1,4]       VERIFIED
DSKØ:PRINT.LIT[1,4]     ?File not found
DSKØ:PRNT.LIT[1,4]      VERIFIED
DSKØ:VCRDIR.LIT[1,4]    VERIFIED
```

or:

```
DSKØ:AM6Ø.TDV[1,6]      VERIFIED
DSKØ:AM62.TDV[1,6]      VERIFIED
DSKØ:AM62A.TDV[1,6]     ?Hash mismatch
DSKØ:AM7X.TDV[1,6]      VERIFIED
DSKØ:AM7Ø.TDV[1,6]      VERIFIED
```

When VERIFY is complete, you see a report:

```
Total errors - 38
Not found ---- 1

Exit
.
```

"Total errors" is misleading-- a better term might be "discrepancies." A deviation in the verification process does not necessarily mean an error exists in the software installation.

The test system on which this verification process was run had 38 discrepancies, but only a couple were actually "errors." Most of the discrepancies were due to the installation of other software packages which used different versions of the system software and patches installed in the remaining software after the AMOS.DIR file was generated.

Other discrepancies were because of changes made by the System Operator. For example, PRINT.LIT was erased from the system because the Task Manager Spooler was installed. On this system, as an aid to those not extremely familiar with AMOS, a PRINT.CMD was placed in [2,2] that displays a message on how to use PRNT.

On the other hand, a file may have verified even though it is currently incorrect. For example, the AM62.M68 source file for the AM-62 terminal driver failed with a Hash mismatch, but the AM62.TDV driver itself verified. Ironically, the source code is good and the terminal driver is missing a correction. On this system, it was not a problem because only AM-60 and AM-62A terminals are used, but it is an error and should be corrected.

3.3.23 (Continued)
Software Verification and Support

As can be seen, VERIFY is a useful troubleshooting tool, but it does not fix anything nor does it do everything. It provides a means by which the integrity of the software installed on the system can be reliably verified. It is up to the troubleshooter to resolve the discrepancies by either justifying their existence or correcting the errors. On the test system discussed above, there were 38 discrepancies. These discrepancies resolved as follows:

17 files were system files overlaid when AlphaC was installed. Using VERIFY on C.DIR showed no unjustified discrepancies for these files.

13 files contained patches released after the system software release to which the AMOS.DIR file applied.

Three files were either deleted or modified by the System Operator.

Three files were system files overlaid by installing UNIFY.

Two files showed unjustified discrepancies, consequently errors, and though these files did not cause a problem in this system, should have had corrections installed.

A feature of VERIFY that makes this analysis easier is the /F switch that causes the screen output to be placed into a file named VERIFY.LST. The syntax for this use of the command is:

VERIFY/F {Name of directory file}

The switch belongs after VERIFY and prior to the file name, otherwise the output is displayed and not deposited in a file.

On the test system, we first used VERIFY on the AMOS.DIR file. Next, we renamed the new VERIFY.LST file to VERIFY.AMS and printed it. Then we used VERIFY on the C.DIR file, renaming the resulting VERIFY.LST file to VERIFY.C, and printing it. Since AMOS was installed first, VERIFY.AMS was used as a reference point. The discrepancies resolved by comparing VERIFY.AMS with the verify listing generated from the C.DIR file were noted on the AMOS listing.

Next UNIFY was checked and found to contain three LOKSER related files that were moved into the SYS: account after being installed on the system. These three files had to be checked visually and eliminated from the AMOS listing. The remaining discrepancies were then checked against the patches released for this version of the operating system. The SPNs listed in the June 1987 issue of the AMSD Journal were used as a beginning.

As another example, the integrity of the software was in question for a computer using AMOS/L version 1.3B that had:

1. AMOS/L 1.3B
2. AM350 Phase I Update
3. AM515 Phase II Update
4. Mag tape Support Software
5. VME Streamer Support Software
6. AlphaWRITE
7. AlphaCALC
8. AlphaC
9. SMC Basic

It took a total of 16 minutes to generate and print the VERIFY listing for each of these packages. It took another ten minutes to eliminate discrepancies due to updates and patches. As can be seen, this procedure is not a quick and mindless

3.3.23 (Continued)
Software Verification and Support

check. But it is thorough and, if used on a known good .DIR file, leaves no doubt as to the integrity of the software. When troubleshooting an elusive problem, it is comforting to know, if the problem still exists, it is not due to the strange side effects of mismatched software.

This brings us the the integrity of the .DIR file itself. These are simple sequential files containing ASCII data. You can use AlphaVUE to edit them. Obviously, if the directory listing on which the VERIFY is performed is not current or has been modified, the results will not be reliable. For this reason, the .DIR files that come with the software should not be changed. If there is any doubt about the integrity of the .DIR file, it is best to reload it from the distribution media. It may be a good idea to catalog the hash codes of the .DIR files for quick reference. Released copies of the files themselves as well as their hash totals are available on AMTEC. Eventually, .DIR files for fully patched systems will also be available.

Along with the vulnerability of the .DIR files to modification comes its versatility as a tool. Anyone can make their own .DIR file and use it for a quick check of software integrity. To generate such a listing, from the command mode of AMOS, type:

DIR/H/V/D {filename}.DIR={speclist}[RET]

Where "filename" is the name you want to give the .DIR file. The "speclist" is a list of device and file specifications which may or may not contain wildcarding symbols. The speclist will identify each file and its location that you will later want verified. The results will be a sequential file containing the raw data in the format below. You will be able to use AlphaVUE to edit the file, or you can print it.

SAMPLE VERIFY DIRECTORY FILE

```
DSK0:AMOS.LIT[1,4]          1.0A(105)       623-425-561-565
DSK0:AMSORT.SYS[1,4]                        154-350-140-316
DSK0:AM334.LIT[1,4]         1.0(100)        030-306-450-155
DSK0:APPEND.LIT[1,4]        1.0             033-502-147-712
DSK0:ASCDMP.LIT[1,4]        1.0(102)        516-161-756-303
DSK0:ATTACH.LIT[1,4]        1.0(107)        531-621-703-401
DSK0:BADBLK.LIT[1,4]        1.1(116)        525-453-766-121
DSK0:BASIC.LIT[1,4]         1.3(204)        314-370-233-634
DSK0:BAUD.LIT[1,4]          2.0(103)        730-347-317-444
DSK0:BITMAP.LIT[1,4]        1.1(101)        115-402-622-173
DSK0:CACHE.LIT[1,4]         1.0(102)        756-007-155-422
DSK0:ISAM.SYS[1,4]          1.1(124)        165-620-170-054
DSK0:CAL100.LIT[1,4]        1.0             203-634-112-214
```

3.3.23 (Continued) Software Verification and Support

To execute properly, VERIFY requires the data in the directory listing file to be in the above format.

Incidentally, if you are going to modify the file before you use it (for example, if you have moved the software to another disk device than the one listed in the original directory listing file), be sure the version number still starts in column 31 (where the first column is 1, not 0) and the hash total starts in column 50. For example, if you need to change DSK0: to DSK11: using the global replace command in AlphaVUE, the whole line from the ":" to the end of the hash total will be shifted one space to the right. This will cause VERIFY to find a discrepancy with the Version number. An easy way to reformat the file in this example is to:

1. Globally replace "DSK0" with DSK11".

2. Globally replace "] " with "]".

With an understanding of and a familiarity with the VERIFY command, the integrity of the software installation need not be a mystery. When trying to isolate a problem, it is most important know what is rather than just what should be. After all, we already know what "should be" is: it "should be" working!

6.9.1
Workaround: Directory Capacity and AlphaCALC

By Brett R. Halle
Distributed Management Systems, Inc.

Currently, AlphaCALC is designed with a limitation of around 400 worksheets per account for directory displays. This is because the directory feature shows worksheets in alphabetic order and the process of internally managing a sorted list in a reasonably small area of memory is limited. Worksheets after this 400 file limit do not appear in AlphaCALC's directory, although they can be accessed as long as you know the worksheet name.

The Problem

In AlphaCALC version 1.2(150), more than 400 worksheets in a directory display can cause unpredictable results. This particular problem in this version of AlphaCALC will be resolved with the next release of the software.

However, this error condition brings to light a more general problem which may exist system wide: account directories with excessive numbers of files are difficult to manage. And, because the Alpha Micro computer can support very large disk storage devices, and has support for even larger devices under AMOS 2.0, file organization becomes an even more important issue.

In addition to the large amount of disk space excessive files in one account occupy, finding the one you want can be a nightmare. Sifting through hundreds of names in a directory can take quite a while. And, naming new ones can be a real challenge since finding a unique, yet intelligible filename can become a full time occupation.

The key to eliminating the difficulty is to implement effective organizational schemes and keep the total number of files at a manageable level. Using the solutions described below will alleviate the problem with AlphaCALC's version 1.2(150) directory feature and make your computer's storage capabilities easier for you to work with.

Some Solutions

There are two levels to solving the directory problems described above:

- Archiving and "spring cleaning" your storage devices on a regular basis.
- Organizing files by user and/or topic by using different accounts on your devices.

Archiving and Spring Cleaning

With the availability of large disks, users tend to be spoiled by having large amounts of free space. Consequently it's easy to forget to erase old and obsolete documents.

The biggest fear in erasing a file having outlived its usefulness is that it might be needed again. This fear can be resolved if the system administrator institutes a consistent and routine schedule for archiving the system's files and keeps a record of what is in the archive. Doing this might entail:

6.9.1 (Continued)
Workaround: Directory Capacity and AlphaCALC

1. On a regular basis (i.e., weekly) perform a complete archive backup of the system on VCR, streamer, or magnetic tape.

2. After the backup and certification is complete, write protect the media. Do this by removing the plastic tab on the left side of the tape label for a VCR tape, or removing the write enable ring for magnetic tape, or by activating the write protect mechanism on a streamer tape.

3. Get a complete directory of your backup media and print it, using the appropriate command for the backup media you used (VCRDIR, STRDIR, or TAPDIR.)

4. Write the date of the archive somewhere on the directory listing **and** the backup media. Store the backup media in a safe place--a fire safe, bank safe-deposit box, etc.--and store the directory listing in a binder. This binder should contain a directory of all your archive tapes.

5. For complete safety, never reuse the archive tapes but do store them as long as possible. There is <u>never</u> any good reason not to make an archive... and it is **always** better to be safe than sorry.

6. Once the archiving is complete, let users on your system know it is time to "spring clean" their word processing and spreadsheet areas by erasing obsolete files. Depending on the computer installation, "spring cleaning" may be done less frequently than making the archive backups.

Organizing Files

Organizing files can also assist you in managing large quantities of files. Many computer installations have only one account with word processing or spreadsheet documents. The result is one account with hundreds or thousands of files and a lot of hassle to find anything. The following ideas for organizing documents is just a start, adjust as needed for the installation:

- Allocate a project number--set of accounts with the same first number--[22,*],[150,*], etc.--for word processing files and another for spreadsheet files. If the installation is supporting multiple companies, allocate a project for each company.

- Allocate a programmer number, the second part of the account number for each user-- [-,1],[-,20], etc.

- If it makes more sense, allocate accounts by topic instead of by user; for example, set aside work areas for legal, proposals, follow-up, estimates, and so on.

- Instead of requiring users to know all the account numbers for the word processing or spreadsheet areas, use AlphaMENU or a security/menu product to allow them to select by name or topic from a menu. This makes managing all the accounts easier and allows users to logically group documents and spreadsheets.

About the Author

Brett Halle has been progrmming in the AMOS environment since 1977. He went on to work for Alpha Microsystems, then to start his own company, Distributed Mangement Systems, Inc. In Pleasanton, California. Currently, DMSI is developing AlphaCALC 2.0 for Alpha Micro.

8.3.8
System Troubleshooting with the Event Logging Utility

By Robert Currier
Director, Future Systems
Advanced Products Development

As system installations have grown ever larger and more complex, the task of troubleshooting a malfunctioning system has become more complex and more difficult. Many installations now make use of a number of "background" jobs whose operation is vital to the system, but which are not connected to real terminals, making it impossible to see any error messages they might report. And, of course, in a busy environment with 100 or more terminals in constant use, it is not realistic to expect terminal operators to report every message or anomaly they might see.

So, faced with a system that does not work correctly--one that crashes on a frequent basis, or one that corrupts data files, for example--the task of trying to isolate and cure a problem can appear quite daunting. Fortunately, there are tools and techniques available that make the task easier.

The Event Logger

The system event logger, LOGGER, is one of the more useful troubleshooting tools. Available under AMOS/32, and able to run under AMOS/L in troubleshooting situations, LOGGER provides you with a permanent record of the abnormal events that have occurred on a system. Each time one of these events occurs, the type of event, time, date, job name, program being run, and other pertinent information is recorded in a disk file. This disk file can then be converted into a printed report using the SYSLOG utility program.

Because the information is recorded automatically, you have a list of all unusual events without relying on terminal operators to record error messages or sequences of events. Because the event logger records all system crashes and reboots, you can tell how often the system is actually crashing when you are not there. By leaving the event logger running at all times, you can monitor the performance and reliability of the system without relying on reports from human operators. Imagine a customer's surprise when you call to ask about a problem they have not even reported yet!

Information on setting up and using the system event logger can be found in AMOS System Operator's Guide. Rather than repeat that information, this article will focus on particular methods you can use with the event logger to help pin down system problems. This information is based on actual experience with system troubleshooting with the event logger.

Troubleshooting Basics

Of course, whether you are using the event logger or not, there are some basic rules of troubleshooting that you should follow in all cases. The first is to gather as much information as possible before changing anything, so you have a good understanding of the system behavior before changes are made. Without this, you cannot really determine if a change has had any impact on system reliability.

To the end user of the system, any time the system is not fully operational is a catastrophe. This is a natural and very understandable reaction. However, for you

8.3.8 (Continued) System Troubleshooting with the Event Logging Utility

to be able to accurately monitor a system, you must understand exactly what the system is doing. To the end user, a problem with all system printers stopping may constitute a "crash," while to you it may simply indicate a spooler problem.

Because they have a markedly different perspective on the problem, end user descriptions of system behavior are notably suspect. The system event logger provides an impartial, permanent record of exactly what is happening on a system. By installing the event logger and monitoring a problem system for some period of time, you can establish a description of the problem severity in terms of crash frequency, whether crashes are preceded by other system errors, and if they occur at certain times of day or only when certain programs are run.

Plan of Attack

Once you have an understanding of how the system is behaving, you can begin to analyze the symptoms to generate a plan of attack. The thing to remember here is to change one thing at a time, carefully monitoring the system behavior before making additional changes. By referring back to the information you gathered at the beginning, you can easily determine if your changes are having a positive or negative impact, or if they are having no effect whatsoever.

As you make each change, note the change on the event logger listing so that you have a permanent record of changes and their effects. You may remember what order you did things in now, but after a few late nights of heavy system troubleshooting you will find you remember very little; a written record can be vital to understanding the problem in its entirety.

Finishing Up

After you finally make a change that makes the problem go away comes the hard part. Having solved the immediate problem your temptation will be to leave things as is and hope for the best.

However, in many situations you will have made a change which seems to fix the problem, but you may not fully understand how the change can result in a fix. By this time you will probably have made numerous changes to the system, most of which probably had no effect on the system at all. What you must now do is remove your changes, one by one, until the problem reappears. Again, the written record of your changes is vital to being able to do this. What you are trying to do is locate the one particular change that cured the problem. Once you have this, you should be much closer to understanding the real cause of the problem and thus be able to reassure yourself that you have cured it, rather than simply covered it up.

Of course, following such a methodical series of steps can be quite difficult when the end user is staring over your shoulder, demanding use of the system. Time and again, though, experience has shown the problem gets fixed quicker and more permanently when care is taken in planning and recording the troubleshooting steps. By eliminating much manual error recording, and by monitoring the system at all times, the system event logger can be a valuable aid in troubleshooting any system problem.

A Real Life Scenario

Here is a real life example of where the event logger has proven invaluable in troubleshooting a system:

8.3.8 (Continued) System Troubleshooting with the Event Logging Utility

A large data entry type of system, consisting of more than 40 terminals, five high-speed printers, and several remote modem connections was upgraded from a S-100 AM-1092 system to a VME AM-2000 system. Soon after this upgrade, the customer started to complain about "regular system crashes." The frequency reported was two or more times per day. They also reported that once a crash occurred, it seemed to continue to crash frequently for the next half hour or so.

Several hardware components were changed but to no avail. Since the upgrade was installed to improve performance, conversion from a perfectly working system to one that crashed frequently was not making a happy customer. Time for methodical troubleshooting!

The first thing we did was install the system event logger. After two days of running--the customer was able to get some work done despite the crashes--an event log was printed for analysis. We found that the system was not actually crashing, but was being rebooted because the operator's terminals were becoming "stuck" at one particular application menu. We also noted these crashes occurred most frequently at approximately 7:30 AM and 1:00 PM. And sure enough, there did seem to be a cluster of crashes around these times which disappeared as the day went on.

At this point the event logger had done two things: it had clarified what the customer meant by a "crash" and had identified a pattern to the crashes that was not reported by the customer. By using this new information, we discovered what happens at these two times of day: the operators all log-on to the application--once when they arrive in the morning and once when they return from lunch.

By focusing attention on the log-on procedure, the application developer was able to locate a potential deadlock situation in the log-on application. Improper record locking allowed two or more operators logging on simultaneously to get into a "deadly embrace," competing for the same file record. Fixing this simple problem made the lockups go away and made for a much happier customer.

But the troubleshooting didn't end here. Two issues remained unresolved: why did the crashes happen in closely grouped batches, and why did the upgrade to an AM-2000 cause this problem to appear?

The answer to the first question came by observing the behavior of the operator when the system was rebooted after the initial lockup. Their first reaction, of course, was to immediately log-on again, all at once, thus causing another lockup. After the third or fourth occurrence of this in the space of a few minutes, the operators would begin to drift away from their terminals, leaving only one or two operators to log-on at a time. By inadvertently limiting the number of simultaneous log-ons, the problem went away!

The answer to the second question was a little harder to isolate. After close examination of the application code, we determined that the dead lock situation was preceded by a chain operation between two programs. On the older AM-1092 system this chain operation required access to the disk, which had a substantial queue of pending requests. By having to wait for the disk, this effectively serialized the log-on requests, preventing the deadlock situation from occurring. When the AM-2000 was installed, additional memory was added to the disk cache, greatly reducing the size of the disk queue. This

8.3.8 (Continued) System Troubleshooting with the Event Logging Utility

had the effect of removing the serialization process, allowing two jobs to enter into the deadlock situation. Thus, while the new hardware configuration greatly boosted performance, that very performance boost uncovered a long dormant application defect.

In addition to helping track down and solve the problem, in this case the event logger provided another important service. Using the printouts from SYSLOG, a presentation was made to the customer describing the problem and its cure. Based on this presentation, the customer was able to understand the problem well enough to restore confidence in the new AM-2000 system.

The clear, step-by-step documentation provided by the logger printouts was a key factor in establishing the dealer's credibility when discussing the problem, and assuring the customer it would not reoccur.

1.2.1 Software Patch Notices: New Distribution Procedures

In the past, Alpha Micro has mailed a copy of each Software Patch Notice (SPN) to all resellers and field engineers. Last May, when AMTEC was reestablished, we began putting SPNs on the AMTEC system. Since then AMTEC usage has grown to over 1,000 downloads per month--most of which are SPNs.

Using AMTEC for patch distribution has many advantages:

- **Quicker Distribution to You:** Patches are available on AMTEC within just a few days of approval. This means you don't have to rely on postal delivery for the information you want, but can pick it up at your convenience.

- **Quicker and More Accurate Patching:** Since you don't have to re-key the source text of the code, you save time and eliminate the chances of a typing error.

- **Availability of Binary Code "Patches:"** In some cases, a software change by patching is not feasible. In these cases, we can make complete binary replacement files available to you via AMTEC--a means of correction we could not easily offer to you with direct mail.

Therefore, as of 1 January 1988, we will discontinue separate direct mailings of SPNs to resellers and field engineers, and make SPNs available exclusively through AMTEC. An SPN summary of all new patches will still appear in each Journal, along with some Software Patch Notices.

We hope this change in distribution method is one you find to be efficient and convenient. If you have any questions regarding AMTEC, or getting SPNs via AMTEC, just give us a call at:

714/641-7608

1.3.1
Alpha Micro's Software Developers Program

Alpha Micro wants to work more closely with our third party Software Developers currently active in developing software for Alpha Micro computers. There are many advantages to be gained for both Alpha Micro and the Developers, given that the program is chartered properly. Many other computer manufacturers such as Apple, DEC, IBM, etc. support a program for Software Developers and the relationship has been advantageous to both parties.

Alpha Micro's Software Developers Program (SDP) is intended to provide qualified Software Developers "special treatment" with respect to new Alpha Micro hardware and software. The idea is to encourage Software Developers to do more applications program development on Alpha Micro systems and to assist and encourage them in any reasonable way we can.

Qualified Software Developers will be able to purchase hardware and software at a significant discount with some restrictions. The restrictions are that equipment purchased under the plan can not be resold within one year of the purchase and only one system may be purchased within a one year time frame. This program is intended to allow Developers to get the latest products Alpha Micro produces. The price list of products available for purchase under this program will be distributed with the membership approval. As an indication of the pricing structure, the AM-1200 system will be available at approximately a 25% discount and AM-15xx/AM-20xx class systems and selected VME add-on boards will be available with approximately a 30% discount. Details will be available with the membership acceptance notification.

Software Developers will be included on a list of people who will be eligible to receive development versions of Alpha Micro software so they can get an early look at what Alpha Micro is preparing to release to the field. This will give Software Developers time to get their applications running and tested on new hardware/software in advance of the actual Alpha Micro product release.

Alpha Micro will establish an easy way to communicate with our Software Developers to answer questions they might have about a new product. This is **not** intended to be a fast path around the support organizations that are already in place.

Alpha Micro has established an on-line information exchange system for use by the development community to exchange programs, problems, documents--any information useful to a wide development audience.

One of the benefits to Alpha Micro from this program is a concerned technical community of Developers who will look at a new product from different points of view, test it, and discover problems or design considerations Alpha Micro can use to improve the product. However, the most important benefit to Alpha Micro is to have more applications programs running on Alpha Micro computers. Applications programs are, after all, the reason systems are sold; they solve a customers' problem.

To qualify for this program, a Software Developer must meet all of these requirements:

1. You must be actively developing software products using Alpha Micro hardware and selling them into the Alpha Micro user base.

1.3.1 (Continued)
Alpha Micro's Software Developers Program

2. You must have a good track record with users on the product(s) they develop and sell, and be willing to supply a reference list on request.

3. You must sign a non-disclosure agreement and have it on file with Alpha Micro.

4. You must sign an Early Product Test Agreement and have it on file with Alpha Micro for each Alpha Micro product delivered for review and test.

5. If you are also a dealer, you must have good credit standing with Alpha Micro.

6. You must be willing to agree and adhere to the rules set up by Alpha Micro regarding test hardware and software.

7. You must be willing to provide Alpha Micro with written information concerning any product problems discovered, design shortcomings observed, product improvement suggestions or any other information useful to Alpha Micro in improving the product.

8. You must agree to abide by the terms and conditions of the program stated on the Software Developers Program Application. (An application is provided at the end of this article.)

Application to be a member of the SDP can be made by mailing a completed Software Developers Program Application mailed to:

Alpha Micro
Advanced Products Department
Attention: Software Developers Program
P.O. Box 25059
Santa Ana, California, 92799

Alpha Micro will review your completed application. If you qualify, you will be informed of your acceptance into the program and receive details about ordering hardware, software and documentation.

alpha micro

Application for Software Developers Program

Company Name: _____ Date _____

Company Address: (where information and software is to be sent)

_____

_____

_____

Software Developer name: _____ Phone: _____

Alternate contact: _____ Phone: _____

Describe your company (dealership, software house, etc.)

_____

_____

Describe the product(s) you have developed and any you're planning to develop:

_____

_____

Why do you feel you qualify to become a member of the Alpha Micro Software Developers Program:

_____

_____

_____

_____

Describe the current hardware configuration you currently use in developing software:

_____

_____

What software do you currently run on your development system? Please include version:

_____

_____

_____

Do you use any special software (UNIFY, AlphaBASE, etc.)? Please include version:

_____

_____

Please read the following terms and conditions of this program, then sign and return the completed application to:

Alpha Microsystems
Software Developers Program
P.O. Box 25059
Santa Ana, California, 92799-5059

* *

Terms And Conditions

1. An Alpha Micro Development information exchange system has been set up specifically to help communications between Developers and the Alpha Micro Development staff. It will support 2400 baud and almost any of the communications packages known to Alpha Micro Developers.

2. Information given in response to questions is limited to documentation, technical notes and verbal answers. Source code will not be distributed for any Alpha Micro product as a part of this program.

3. Questions will be answered only for Alpha Micro products currently available through the Alpha Micro Software Developers Program. The list of supported products may change without notice.

4. Questions will be answered as quickly as possible within the constraints of the workload being handled by the Alpha Micro Development Organization.

5. Questions about current product problems will continue to be directed to Alpha Micro Technical Support for resolution.

6. Alpha Micro does not guarantee that any information provided under this program will fix a Developer's problem or, if retrofitted, be suitable to fix a similar problem in a previous version of a product. Alpha Micro assumes no responsibility for the accuracy of information supplied to the Developer nor will Alpha Micro incur any liability to the Developer for any information provided to the Developer under the Software Developers Program.

7. CONFIDENTIALITY. The parties agree to hold all information gained as a result of participating in this program in confidence and not disclose such information to any third parties. The parties may disclose such information to their respective responsible employees, but only to the extent necessary to carry out the purposes for which the confidential information was disclosed. Both parties agree to take reasonable steps to instruct their employees regarding the confidentiality obligations. The obligations of confidentiality shall not apply to information which: 1) is publicly available at the time of disclosure, 2) has been released by the disclosing party without restrictions, 3) has been lawfully obtained from a third party under no obligation of confidentiality, or 4) is independently developed by employees of the receiving party who have not had access to the information.

8. WARRANTY DISCLAIMER. Early products and documentation released to the Developer by Alpha Micro under the Software Developers Program are provided on an "as-is" basis. Alpha Micro MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, REGARDING THE PRODUCTS OR DOCUMENTATION, ITS MERCHANTABILITY, OR ITS FITNESS FOR ANY PARTICULAR PURPOSE. In no event shall Alpha Micro be liable for incidental or consequential damages resulting from the use, sale, or distribution of products incorporating or based upon information and software provided by Alpha Micro to the Developer or others, whether under theory of warranty, tort, or product liability.

9. Alpha Micro reserves the right to limit the number of members in the Software Developers Program at any one time. Alpha Micro reserves the right to terminate the membership status of a Developer without cause, upon (30) days written notice or immediately, without notice, if the Developer is in breach of the confidentiality obligations set forth in Section 7. In the event I am accepted into the Alpha Micro Software Developers Program, I agree to abide by the above terms and conditions.

Company Name: _____

By: _____

Title: _____

Date: _____

4.2.2
Customer Education - Training Class Schedule

General Information

Descriptions and tuition rates for each class shown on the schedule are given below.

Additional classes are available on request: AlphaWRITE, AlphaCALC, AlphaLEDGER, using ISAM and LOKSER and writing Terminal Driver Programs.

The Alpha Micro Customer Education Department , offers a variety of training programs for Alpha Micro users:

- o Hardware/Maintenance Courses
- o Software Operating Courses
- o Language Courses
- o Word Processing Courses
- o Video Training Tapes
- o Customized Classes
- o Courses at Your Location
- o Courses at Our Facilities
- o Training Manuals for Classes
- o Alpha Micro Technical Journal - technical newsletter

Our classes offer hands-on experience and lab time in which students put into practice the theory and information presented in the class.

Enrollment Procedures

To enroll in any one of our classes please use the enrollment form below and mail to:

Customer Education Training Classes
Alpha Micro
P.O. Box 25059
Santa Ana, California 92799

Receipt of class tuition with your enrollment form lets us reserve your seat in the class. If we do not receive payment at least ten working days prior to the first day of class, we must offer the seat to the next person on our waiting list.

If you have enrolled in a particular class and for some reason cannot attend or must change the date, please notify us at least ten working days before the class start date. This lets you avoid the ten percent service charge necessary for class cancellation. If we do not receive notification that you wish to cancel a class at least five working days before the start date of that class, it will not be possible to refund your tuition fee.

For more information contact Nancy Steen in the Customer Education Department at (714) 641-6330.

4.2.2 (Continued)
Customer Education - Training Class Schedule

= = = = = = = = = = = = = = = Enrollment Form = = = = = = = = = = = = = =

Class (Title & Date) ______________________________

Student Name(s) ______________________________

Company Name ______________________________ DI# ________

Address ______________________________

City ______________________________ State ______ Zip ________

Phone# (   ) ______________   Tuition $ ______________

= =

CUSTOMER EDUCATION CLASS SCHEDULE
JANUARY TO JUNE 1988

Class/Date	Jan	Feb	Mar	Apr	May	Jun
AlphaBASE					16-20	
AlphaBASIC				11-15		
C Language		15-19				6-10
UNIFY Database		22-26				13-17
Intro. to AMOS	11-15		14-18		2-6	
Advanced AMOS	18-22		21-25		9-13	
System Maint. Wk 1		1-5		11-15		13-17
System Maint. Wk 2		8-12		18-22		20-24
VME Systems Seminar	18-20		9-11		4-6	

Alpha Micro Customer Education
Class Descriptions

The classes described below are being offered by Alpha Micro Customer Education during January through June 1988 at Alpha Micro in Santa Ana, California.

Customized classes are also available. (60 day notice, please). Special, on-site, and custom classes vary in cost. Travel and living expenses are in addition to course prices. All tuition prices are subject to change without notice. For more information please call Nancy Steen at (714) 641-6330.

4.2.2 (Continued)
Customer Education - Training Class Schedule

AlphaBASE

Two segment class: first 3 days for beginners, last 2 days for advanced students. Provides overview of program operation, shows basic steps necessary to create a data base, menu and report. Subsystems described include MESSAGE, SELECT/SORT and PASS. ZIP CODE and TABLE file maintenance also discussed. Additional programming techniques and installation procedures are also given.

5 Day Class - Tuition $725/week Offered: May 16-20, 1988.

AlphaBASIC

This class is offered as a one week session.

Topics include purpose and implementation of structured programming, writing and debugging AlphaBASIC programs, using both interpreter and compiler modes. Unique AlphaBASIC features, lower-level file I/O and file manipulation are also described. Also, statements and extended statement techniques. Advanced file handling, dealing with sequential, random access ISAM files, XCALLS, and program chaining are also covered. All concepts will be applied to actual programming applications.

5 Day Class - Tuition $725. Offered: April 11-15, 1988

AlphaC

Takes you you through the development of a collection of programs ranging from introductory to advanced. The programs illustrate AlphaC features using modern programming exercises. This in-depth course covers various aspects of AlphaC: expressions, declarations, initializations, arrays, pointers, structures, and more.

5 Day Class - Tuition $725.00/week. Offered: February 15-19 1988, June 6-10 1988.

Unify Data Base

This one week class provides an introduction to the Unify Data Base Manager with several advanced programming exercises. Topics include database design, menu and report generation, SQL (Structured Query Language), and installation.

5 Day Class - Tuition $725.00/week. Offered: February 22-26 1988, June 13-17 1988.

Introduction to AMOS (Alpha Micro Operating System)

Provides an overview of the Alpha Micro Operating System (AMOSL and AMOS/32). Emphasis placed on system commands, their functions and use; the System Initialization Command File. An introduction to AlphaVUE and command files is also given. Alpha Micro users develop and improve the skills and procedures necessary to implement system commands, add jobs, peripherals and/or backup devices to an existing system.

5 Day Class - Tuition $625/week Offered: January 11-15 1988, or March 14-18 1988. or May 2-6 1988.

Advanced AMOS (Alpha Micro Operating System)

Provides a more advanced study of the Alpha Micro Operating Systems - AMOSL and AMOS/32. Emphasis on troubleshooting the system from a software perspective and a more detailed study of system commands: how they may be used to solve

4.2.2 (Continued)
Customer Education - Training Class Schedule

or prevent system problems. Also described: extended command and DO file features, disk allocation, system housekeeping methods, recovery of lost data files and memory configurations available to Alpha Micro users.

5 Day Class - Tuition $625/week (if taken consecutively with "Introduction to AMOS," total tuition is $1000 for both classes.) Offered: January 18-22, 1988, or March 21-25, 1988, or May 9-13, 1988.

Systems Maintenance

This class is offered as two one-week sessions and is especially for maintenance technicians and system engineers.

Week One concentrates on maintenance and troubleshooting of the AM-100/L S-100 bus based systems.

Week Two covers operation and troubleshooting of the AM-1000, AM-1500 and AM-2000 Series Systems. A cursory coverage of the Workstation, Terminal and Printers is also included.

Week One - 5 Day Class - $625 tuition. Offered February 1-5, 1988, or April 11-15, 1988, June 13-17, 1988.

Week Two - 5 Day Class - $625 tuition. Offered February 8-12, 1988, or April 18-22 1988, or June 20-24, 1988.

Week One and Week Two - 10 Day Class - $1000 tuition. Offered February 1-12, 1988, or April 11-22, 1988, June 13-24, 1988.

VME Systems Seminar

Covers description and operation of the AM-1500/2000 Series Systems and the new VME Bus. Includes instruction on installation and setup of the VME Bus based boards. Boards discussed include the AM-175 and AM-180 CPU; AM-182 Cache Memory board; AM-117 S-100 Bus adapter; AM-212 Floppy controller; AM-350 Intelligent I/O controller; AM-355 I/O; AM-515 Intelligent Disk Accelerator; AM-520 Disk Controller; AM-630 VCR controller; AM-706, AM-730 and AM-740 Memory boards; AM-433, AM-435 and AM-437 Disk Subsystems; AM-905 SIO Subsystem; AM-640 Magnetic Tape Subsystem. Selftest and software requirements will also be discussed.

3 Day Seminar - Tuition $550 (includes training manual.) Offered: January 18-20, 1988, or March 9-11, 1988, or May 4-6, 1988.

AMSD JOURNAL TABLE OF CONTENTS UPDATE PAGES

The next pages of the Journal are updated Table of Contents pages for your back issue volumes. The updated pages are:

HARDWARE INFO. VOLUME:	- Hardware Volume Table of Contents (3 pages) - Section 4 - Tape Drives & Interface Boards (2 page) - Section 5 - Memory and Memory Management (1 page) - Section 7 - Integrated Systems (1 page)
SOFTWARE INFO. VOLUME:	- Section 3 - AMOS/L Operating System (1 page) - Section 6 - Software Packages (1 page) - Section 8 - AMOS/32 Operating System (2 pages)
GENERAL INFO. VOLUME:	- Section 1 - Policies and Procedures (2 page) - Section 4 - Training (1 page)

- A footer line at the bottom of each table of contents page shows you revision information. This line shows month, year, volume and issue number of the Journal this table of contents page arrived with.

- All table of contents pages have a title line showing which volume they belong in: Hardware, Software or General Information.

- Entries for articles published since 1983 show the month and year of publication.

- Cross reference article entries use this format:

  "Article Name" - Cross reference: See Volume Name Article #.#.# - [Month Year]

  Where Volume Name is Hardware, Software or General Information. Where #.#.# is the article number designating section, category and article number. (For example, article 6.4.3 is filed in section 6, under category 4 and is the 3rd article in category 4.) [Month Year] is the Journal publication date for the article.

VOLUME I - HARDWARE

Table Of Contents

(continued on next page)

VOLUME I - HARDWARE (continued)

VOLUME I - HARDWARE (continued)

SECTION 4 - TAPE DRIVES & INTERFACE BOARDS

4.1 AM-600

4.2 AM-610

4.3 AM-620

4.4 AM-640

4.5 Video Cassette Recorders and Related Hardware

4.6 Magnetic Tape Drives

(continued on next page)

SECTION 4 - TAPE DRIVES & INTERFACE BOARDS (continued)

SECTION 5 - MEMORY & MEMORY MANAGEMENT

SECTION 7 - INTEGRATED SYSTEMS

Section 3 - AMOS/L Operating System (continued)

3.2 AMOS/L Patches (continued)

3.3 AMOS/L Utilities

3.4 Programming Information

3.5 Program Hash Totals

SECTION 6 - SOFTWARE PACKAGES

SECTION 8 - AMOS/32 OPERATING SYSTEM

8.1 AMOS/32 Releases

8.2 AMOS/32 Patches

(continued on next page)

Section 8 - AMOS/32 Operating System (continued)

8.2 AMOS/32 Patches (continued)

8.3 AMOS/32 Utilities

8.4 Programming Cautions

8.5 Program Hash Totals

1.0 Policies and Procedures (continued)

(continued on next page)

SECTION 4 - TRAINING

4.0 Training

4.1 Training - General Information

4.2 Training Class Schedules

4.3 Video Training Tapes