sun microsystems

Software Technical Bulletin
August 1989

Technical Information Services

Part Number 812-8908-01
Issue 1989–08
August 1989

sun microsystems

Software Technical Bulletin
August 1989

Technical Information Services

Sun Microsystems, Inc. • 2550 Garcia Avenue • Mountain View, CA 94043 • 415-960-1300

Software Technical Bulletins are distributed to customers with software/hardware or software only support contracts. Send comments or corrections to 'Software Technical Bulletins' at Sun Microsystems, Inc., 2550 Garcia Ave., M/S 2-318, Mountain View, CA 94043 or by electronic mail to *sun!stb-editor*. U.S customers who have technical questions about topics in the Bulletin should call the Sun Customer Software Services AnswerLine at **800 USA-4-SUN**. Other customers should call the numbers listed in *World Hotlines* appearing in Section 1.

Contents

1

NOTES & COMMENTS

1

NOTES & COMMENTS

Editor's Notes

Editor's Notes

The editor's notes for this August 1989 issue include the items of interest listed below.

- TOPS networking product and ordering information
- World hotlines for customer service calls
- World-wide bug reporting information
- Limited permission to duplicate your STB
- Hints and Tips: using your `/etc/printcap` file
- The Hackers' Corners: email icons showing email received
- Configurations: updated software release level tables, effective June 23, 1989

TOPS: Product and Ordering Information

See the note later in this Notes and Comments section containing the address and telephone number to use to get more information on TOPS networking products.

World Hotlines

For Sun customers world-wide served by your local service groups, use the customer service telephone numbers listed in this monthly item. Also, look to this section during the upcoming year for details on your local support call policies and procedures.

Reporting Bugs World-Wide

A list of Sun service centers, addresses, email hotlines, and telephone hotlines appears. The information in this monthly note continues to be expanded as Sun software service centers are added world-wide.

STB Duplication Permission

This notice is published monthly, giving customers useful information regarding ordering and duplicating additional STB copies. This duplication permission is limited, as detailed in the note.

Hints and Tips

This month's hints and tips contain information on three different subjects. The first is about differences in finding zero-divides when done in C and in FORTRAN. The second is a series of tips on how to use floppy disks as a convenient way to browse filesystems. Finally, the third includes hints on how to do tapeless installations.

The Hackers' Corner

This month's **Hackers' Corner** contain three programs that demonstrate the capabilities of the `cg6` accelerated, 8-bit color frame buffer.

Again, please note that such applications, scripts, or code are not offered as released Sun products, but as items of interest to enthusiasts wanting to try out something for themselves. They may not not work in all cases, and may not be compatible with future SunOS releases. Please consult your local shell script or programming expert regarding any application, script, or code problems.

Configurations: Current Sun Software Products and Release Level Tables

The seven tables showing current Sun software product release levels appear monthly. These tables show release levels for operating systems, communications products, unbundled languages, unbundled applications, unbundled graphics, other products, and TOPS networking products. The tables in this issue are updated through June 23, 1989.

Thanks.

The STB Editor

TOPS Ordering Information

TOPS Product and Ordering Information

This issue of the Software Technical Bulletin features the first of a series of articles on TOPS, the group of networking products that link together IBM PCs or compatibles, Apple Macintoshes, and Sun workstations or both over an Ethernet and or AppleTalk network or both.

For TOPS product and ordering information, contact the TOPS sales group directly at the address shown below.

TOPS, a Sun Microsystems Company
950 Marina Village Parkway
Alameda, CA 94501

(415) 769-8700

World Hotlines

World Hotlines

Sun Customers throughout the world have service hotlines available for both software and hardware support questions. The service hotlines are shown below. If your country is not shown in the table, please phone your local Sun sales office.

The world hotlines are divided into those for Canada and the USA, CSD Europe, and Intercon. Intercon includes those countries outside the USA, Canada, Europe, and northern Africa.

Canada and the United States

Canada	Montreal	(514) 738-4885
	Ottawa	(613) 723-8112
	Toronto	(416) 475-6745
	Winnipeg	(204) 222-2333
	Edmonton	(403) 482-7264
	Calgary	(403) 262-6722
	Vancouver	(604) 684-4120
United States	All, including Puerto Rico	1-800-USA-4-SUN

CSD Europe

European Customer Service	Surrey Sun Microsystems Europe Inc.	(44) 276 51440
France	Paris Sun Microsystems France SA	(33) 1 4094 8080
Germany	Munich Sun Microsystems GmbH	(49) 089/46008-321
The Netherlands	Soest Sun Microsystems Nederland BV	(31) 2155 24888
Sweden	Solna Sun Microsystems AB	+46 8 764 78 10
Switzerland	Zurich Sun Microsystems (Schweiz) AG	(41) 1 828 9555
United Kingdom	Albany Park Sun Microsystems UK Ltd	(44) 0276 691052

Intercon		
Australia	Sun Microsystems Australia	(011-61-2) 436-4699
Hong Kong	Sun Hong Kong	(011-852-5) 865-1688
Japan	C. Itoh Data Systems Nihon Sun	(011-81-3) 497-4676 (011-81-3) 221-7021
Countries Not Listed	All countries outside the USA, Canada, Europe, northern Africa, Australia, and Japan	(415) 496-6119

Reporting Bugs

Submitting Bugs and Email Service Calls

This article contains two sections for submitting bugs. The first section describes procedures to use within the United States. The second section describes Customer Service Division (CSD) Europe procedures.

Submitting Software Bugs: United States and Canada

This section contains information on reporting bugs within the U.S., for customers holding and not holding support contracts.

Sun's United States Answer Center (USAC) within CSD accepts software bug reports from Sun users via electronic mail and by phone. The method you use to submit a bug report varies with your needs.

U.S. users holding support contracts can report bugs to USAC via the **(800) USA-4-SUN** phone hotline. Canadian users holding support contracts should call (800) 225-2615. The USAC phone hotline is the fastest way for a customer to find out if a problem is known and if a workaround exists. The status of previously-reported bugs can also be obtained in this way. The list of open software bugs is contained in the Customer Distributed BugsList (CDB).

Customers holding support contracts can also submit bug reports electronically to the address *sun!hotline* (*hotline@sun.COM*). This method generates a service order, and can be used when lines of code or other information difficult to relay over the phone is needed to describe the bug.

- Whenever possible, customers should use the Online Bugs Database (OBD) described below before submitting bugs, to avoid resubmitting an already-known bug.

- Please note, however, that the alias *onlinebugs-db@sun.com* is **not** the appropriate avenue for submitting bugs.

Customers who do not hold Sun software support contracts can report bugs via electronic mail to the address *sun!sunbugs* (or *sunbugs@sun.COM*). These reports are reviewed periodically to determine proper disposition. Those reports determined to be from supported customers are forwarded to the U.S. Answer Center for handling. Reports from customers who cannot be verified as holding a support contract are reviewed by Sun's Software Quality Assurance (SQA) personnel. An internal bug report is generated if the reported bug is new and verifiable.

Finally, customers not holding software support contracts may call the **(800) USA-4-SUN** phone hotline to report a problem and request support on a Time and Materials (T&M) basis. Canadian customers should call (800) 225-2615. In this case, please have a Purchase Order (PO) number for billing purposes.

The Online Bugs Database (OBD)

The OBD contains the same information as the Customer Distributed BugsList (CDB). The information available through the OBD is updated during the first week of each month. As a result, you receive the most timely information available on open known bugs and temporary workarounds for Sun software in an easily-accessible, online format.

The OBD service is initially available only within the United States. Future plans include worldwide introduction and distribution.

Information Provided by the OBD

The OBD provides you with rapid telephone access to the following information.

- Software Bug Reference Number--a unique identification number assigned to each valid software bug by Sun
- Online Bug Synopsis--a one-line summary of the software bug
- Bug Description--a brief description of the bug, with examples if available
- Software release(s) in which the bug was reported
- Affected configurations
- Temporary workarounds, where available

To use the OBD, simply dial the telephone number and enter the system password; both provided in the *Online Bugs Database Reference Manual*, part number 812-1001. This manual is automatically sent to the site contact of all Sun customers holding valid support contracts. The OBD is available at all hours, except for scheduled updates and preventive maintenance. System support is available during standard U.S. Answer Center business hours by calling the support numbers given above.

OBD Search Criteria

After logging in, you can quickly search the OBD by any one of the below parameters.

- Software Bug Reference Number
- Software Category (such as kernel, SunINGRES, or Datacomm)
- Software Subcategory (such as documentation related to a specific category)
- Software Release (such as 4.0, 3.5, 3.4, 3.2, 3.0)

Search capabilities can be enhanced by combining several of the primary search parameters. For example, all release 3.4 NFS bugs within the network category

can be searched. In most situations, you can locate a particular software bug and its related workaround within 30 seconds.

To ensure that your OBD use is as efficient as possible, a fast, easy-to-use Help facility is also provided. Help is available throughout your OBD session.

Summary: United States and Canada

For U.S. contract customers, **(800) USA-4-SUN** is the best method to report bugs. Canadian contract customers should report bugs to (800) 225-2615. The electronic mail address *sun!hotline* is available to submit materials that are difficult to relay over the phone. The OBD is available to research currently-known bugs.

For non-contract customers, the electronic mail address *sun!sunbugs* is available to report bugs.

To help us serve you better, please include the following information with all electronic mail reports.

- Your name
- The name and address of your organization
- Your Sun site code, if available
- Your workstation model and serial number
- The software release(s) you are running
- A description of the problem that you are experiencing
- Please do *not* submit bugs to *sun!onlinebugs-db*

Submitting Software Bugs: CSD Europe

This section contains information on reporting bugs within CSD Europe, for customers holding and not holding support contracts.

Procedures for submitting bugs are similar to those used in the United States. All customers should use their local country Answer Center to report bugs, with contract customers receiving a specific follow-up.

Sun customers not holding software service contracts can call their local Answer Center, and will need to provide a Purchase Order (PO) number at the time of the call.

Summary: CSD Europe

To help CSD Europe service centers serve you better, please include the following information with all electronic mail reports:

- Your name

- The name and address of your organization
- Your Sun site code, if available
- Your workstation model and serial number
- The software release(s) you are running
- A description of the problem that you are experiencing

Detailed information for European Customer Service and individual countries follows.

European Customer Service

The European Customer Service office is located at the address shown below.

Sun Microsystems Europe, Inc.
Bagshot Manor
Green Lane
BAGSHOT
Surrey GU19 5NL
United Kingdom

Telephone: (44) 276 51440

Telefax: (44) 276 51287

Telex: 859017

France

Report bugs to the France Answer Center at the postal address shown below.

Service "HOT LINE"
SUN Microsystems France
La Boursidiere
R.N. 186
92357 Le Plessis Robinson Cedex

Hotline Telephone: (33) 1 4094 8080

Telefax: 0276 691774

Special Dispatch Arrangements:

Please provide Dispatch with the following items:

System serial number *or* Contract number

Arrangements for Non-Contract Customers:

Please provide a valid PO number for billing on a Time and Materials (T&M) basis, and order this support at the above address.

Germany

Report bugs to the Germany Answer Center at the postal address shown below.

Hotline
Sun Microsystems Gmbh
Stoerungsannahme
Am Hochacker 3
D-8011 Grasbrunn 1
West-Germany

Hotline Telephone: (49) 089/46008-321

Telex: 5 218 197 sun

Telefax: 089/46008-400

Email Address: *{sunuk,unido}!sunmuc!hotline*

Arrangements for Non-Contract Customers:

> Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

The Netherlands

Report bugs to The Netherlands Answer Center at the postal address shown below.

Sun Microsystems Nederland BV
Birkstraat 95-97
3768 HD SOEST
The Netherlands

Hotline Telephone: (31) 2155 24888

Arrangements for Non-Contract Customers:

> Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Sweden

Report bugs to the Sweden Answer Center at the postal address shown below.

Sun Microsystems AB
Hemvarnsgatan 9
S 171 54 Solna
Sweden

Hotline Telephone: +46 8 764 78 10

Email Address: *hotline@sunswe.se* or *sunswe!hotline*

Arrangements for Non-Contract Customers:

> Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Switzerland

Report bugs to the Switzerland Answer Center at the postal address shown below.

Sun Microsystems (Schweiz) AG
Postfach
Rohrstrasse 36/38
CH-8152 GLATTBRUGG
Switzerland

Hotline Telephone: (41) 1 828 9555

Email Address: *sunuk!sunswis!hotline*

Special Dispatch Arrangements:

Provide Dispatch with the following item:

Contract number

Arrangements for Non-Contract Customers:

Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

United Kingdom

Report bugs to the UK Answer Center at the postal address shown below.

Hotline
Sun Microsystems (UK) Ltd
Technical Centre
Unit 3D
Albany Park
Frimley
Surrey
GU15 2PL

Hotline Telephone: (44) 0276 691052

Telefax: 0276 691774

Special Dispatch Arrangements:

Please provide Dispatch with the following items:

System serial number *or* contract number

Arrangements for Non-Contract Customers:

Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

Submitting Software Bugs: Intercon

This section contains information on reporting bugs within Intercon, for customers holding and not holding support contracts.

Procedures for submitting bugs are similar to those used in the United States. All customers should use their local country Answer Center to report bugs, with contract customers receiving a specific follow-up.

Sun customers not holding software service contracts can call their local Answer Center, and will need to provide a Purchase Order (PO) number at the time of the call.

Summary: Intercon

To help Intercon service centers serve you better, please include the following information with all electronic mail reports:

- Your name
- The name and address of your organization
- Your Sun site code, if available
- Your workstation model and serial number
- The software release(s) you are running
- A description of the problem that you are experiencing

Detailed information for individual countries follows.

Australia

Report bugs to the Australian Answer Center at the postal address shown below.

Hotline
Sun Microsystems Australia Pty Ltd
PO Box 320
Artarmon
NSW 2064

Hotline Telephone: (011-61-2) 436-4699

Telefax: 02 436 1084

Special Dispatch Arrangements:

Please provide Dispatch with the following items:

System serial number *or* contract number

Arrangements for Non-Contract Customers:

Please provide a valid PO number for billing on a Time and Materials (T&M) basis.

STB Duplication

Duplicating the STB

Your company's software support contract includes a monthly issue of the STB. Each month, the copy of your STB is mailed to your company's primary contact person or department. Sites with more than one contract may receive more than one STB copy, depending on how the contracts are set up.

Your primary contact person or department may duplicate this 'master' STB copy for all Sun workstation end-users. So long as you duplicate copies and route them only internally, there are no copyright infringement problems.

This limited permission for duplication is for your convenience only, however, and does not include any duplication for resale, for distribution outside your company, or for distribution to employees of companies not having a Sun software support contract.

Direct STB Purchase

The STB is sent to the primary contact person named in all software support contracts. Sun is looking into methods by which customers holding these contracts may purchase extra copies directly.

Look to this column for an announcement regarding the purchase of extra STB copies.

Further Questions

If you have any questions, comments, or articles regarding the STB or CDB, please send your ideas and questions to *sun!stb-editor*.

Errata

Errata

The following errata applies to the Dependency Tables appearing on pages 980, 982, and 983 of the July, 1989 *Software Technical Bulletin.*

The Sun Operating System (SunOS) release level shown for the Sun386*i* system is incorrect. The release level shown is SunOS 4.0.3. This should be changed to Sun386*i* SunOS 4.0.1. The tables on page 981, showing SunOS 4.0 for the Sun386*i*, are correct and are not affected by this errata.

Please note that the 'Sun386*i*' portion of the SunOS for Sun386*i* machines does not appear in the table headers due to space limitations.

2

ARTICLES

2

ARTICLES

Large Disk Filesystems

File Systems and Large Disks

Customers with large disks may not get an adequate number of inodes per cylinder group. `mkfs` does not allow creation of more than 2048 inodes per group.

The Problem Defined: Bug ID 1008866

The following numbers are from `newfs` on a Fujitsu 2344 disk drive. The first example shows the initial condition of 50.0 Mbytes storage.

```
# /etc/mkfs /dev/rxy0e 97686 67 27 8192 1024 16 10 60 2048 t 0
/dev/rxy0e:     97686 sectors in 54 cylinders of 27 tracks, 67 sectors
        50.0Mb in 4 cyl groups (16 c/g, 14.82Mb/g, 2048 i/g)
super-block backups (for fsck -b#) at:
 32, 29056, 58080, 87104,
```

This second example shows the error message when the number of tracks times the number of cylinders per group is an odd number.

```
# /etc/mkfs /dev/rxy0e 97686 67 27 8192 1024 8 10 60 2048 t 0
cylinder groups must have a multiple of 16 cylinders
```

The number of cylinders per group depend indirectly upon `fs_cpc`, the cylinders per cycle in the position table. The number of cylinders per group has to be a multiple of the cylinders per cycle as shown below.

```
cpc = 16 >> x,
```

where x is the largest integer such that

```
track * sector = N * (2**x),
```

where N is an integer.

The Workaround

You must reduce the number of tracks from 27 to 26 and reduce the number of cylinders per group from 16 to 8. This increases the number of cylinder groups and, hence, the number of available inodes for large disks.

Note that the reduction of tracks is required, since the product of the number of tracks and cylinders per group must be an even number. See the example below.

```
# /etc/mkfs /dev/rxy0e 94086 67 26 8192 1024 8 10 60 2048 t 0
/dev/rxy0e:     94068 sectors in 54 cylinders of 26 tracks, 67 sectors
       48.2Mb in 7 cyl groups (8 c/g, 7.14Mb/g, 2048 i/g)
super-block backups (for fsck -b#) at:
 32, 14048, 28064, 42080, 56096, 70112, 84128,
```

Note that you now have 48.2 Mbytes of storage, a 1.8 Mbyte reduction. There are now 67 sectors, 26 tracks, giving a total number of 94068 sectors.

Sun386*i* Remote OS Installs

Remote Installation of Sun386*i* SunOS 4.0.1 From Tape or Disk Image

This article contains information of use to those remotely installing Sun386*i* SunOS 4.0.1.

Remote Installation Kit

The Sun386*i* comes with a standard remote installation kit which allows the user to install Sun386*i* SunOS 4.0.1 on a Sun386*i* in two ways:

1. from a remote 1/4" (QIC-24) or 1/2" (6250 bpi) tape
2. from a file system on a remote system in which the contents of the Sun386*i* SunOS 4.0.1 distribution tapes must first be loaded onto the remote system via an NFS mount.

Both types of installations require that the Sun386*i* be configured onto an existing network. This allows access to remote tape or file system. The installation requires an unused host name and an unused INET address to be assigned to the Sun386*i* (only) for the duration of the installation procedure. The final host name and INET address of the Sun386*i* is determined when the Sun386*i* boots Sun386*i* SunOS 4.0.1 for the first time.

Boot Diskettes

Both types of remote installations require two recovery/install diskettes. Each of these diskettes contains a Sun386*i* SunOS file system. The first diskette is bootable. It boots a memory-based kernel with its root file system in main memory. The second diskette contains the file system used to initialize the root file system. This file system contains the installation script and its required programs. The file system images for these diskettes are file numbers 25 and 26 of the Application SunOS tape. Diskettes must be created from these images before proceeding.

Constructing Boot Diskettes

Boot diskettes can be constructed by following the instructions below:

1. Copy the tapes files to regular files.

```
# mt -f /dev/nr<tape> rew
# mt -f /dev/nr<tape> fsf 24
# dd if=/dev/nr<tape> of=munix bs=9k count=160
# mt -f /dev/nr<tape> fsf 1
# dd if=/dev/nr<tape> of=munixfs bs=9k count=160
```

   where <*tape*> is either `st8` for 1/4" tapes or `mt8` for 1/2" tapes.

2. Copy the regular files to diskette.

   NOTE: First, the image files may need to be transferred to a system that has a diskette drive.

```
# dd if=munix of=/dev/rfd0c bs=9k count=160
# dd if=munixfs of=/dev/rfd0c bs=9k count=160
```

For the remainder of these instructions, these diskettes will be referred to as *Remote Installation Diskette Number 1* and *Remote Installation Diskette Number 2* respectively.

Installing from a Remote Tape Drive

To install Sun386*i* SunOS 4.0.1 on a Sun386*i* from a remote tape drive, first gather the following information:

- The host name and INET address of the system with the remote tape drive
- The host name and INET address of the Sun386*i* system. Any unused host name and unused INET address will do. (The system will be assigned a real host name and INET address at boot time.)
- The optional Applications Supplement and Developers Toolkit clusters to be installed.

Make sure that the system with the tape drive allows root equivalence to the Sun386*i* via the `/.rhosts` file. To perform the installation, follow the instructions below:

1. Insert the *Remote Installation Diskette Number 1* and reboot.
2. The following prompt will appear:

   ```
   Insert Application SunOS diskette 2, confirm (y/n):
   ```

   Insert *Remote Installation Diskette Number 2*, type **y**.
3. If there are more than one disk, the following menu will appear:

   ```
   Which disk do you want to be your system disk?
     1 - the one in the system unit (sd2)
     2 - the one in the expansion unit (sd0)
     3 - shutdown
   Enter [1-3]:
   ```

   Enter **1** or **2**, depending upon which disk will be designated as the system disk.
4. The following menu will appear:

   ```
   How do you want to install SunOS?
     1 - from a network file system
     2 - from a remote tape drive
     3- shutdown
   Enter [1-3]:
   ```

   Type **2**.

5. Another menu will appear prompting for the size of the distribution tapes:

```
What size are your distribution tapes?
  1 - 1/4"
  2 - 1/2"
Enter [1-2]:
```

Type **1** for 1/4" tapes or **2** for 1/2" tapes.

6. Enter the host name and INET address of the remote system and the Sun386*i*. The prompts and examples of the required input are shown below:

```
Enter the host name of the remote system: fred
The remote system is fred, correct? (y/n): y
Enter the INET address of fred:192.9.200.10
The INET address of fred is 192.9.200.10, correct? (y/n): y
Enter the host name of this system: barney
This system is barney, correct? (y/n): y
Enter the INET address of barney:192.9.200.20
The INET address of barney is 192.9.200.20, correct? (y/n): y
```

Where **bold** represents the example input.

7. At the prompt, load (mount) the Application SunOS tape of the distribution:

```
Insert Application SunOS tape 1, confirm (y/n):
```

Mount the tape and type **y**.

NOTE: The remainder of this procedure is very much installation from a local 1/4" tape drive.

8. At the prompts, specify which Applications Supplement and Developers Toolkit clusters will be loaded. You can load all or selected Applications Supplement or Developers Toolkit clusters.

Example:

```
Do you want to load all the Application SunOS optional cluster? (y/n): y
Do you want to load all the Developers Toolkit optional cluster? (y/n): n
Do you want to load any Developers Toolkit optional clusters? (y/n): y
Do you want to load the base_devel cluster? (y/n): y
Do you want to load the sys_devel cluster? (y/n): n
     .
     .
     .
Do you want to load the config cluster? (y/n): n
```

Where **bold** represents the example input.

NOTE: You will eventually be asked to change tape if loading any Developers Toolkit clusters.

9. When the installation completes, The following message will appear:

```
Installation complete.
Remove the diskette from the drive and power cycle system.
      .
      .
      .
>
```

Installing from a NFS File System

To install Sun386*i* SunOS 4.0.1 from a network file system, first load the Sun386*i* SunOS 4.0.1 tapes on the system that will act as the NFS server for the installation.

The Sun386*i* SunOS 4.0.1 distribution consists of several components. The component disk space requirements are described in the following table:

component	size	optional	
CoreOS	21666K	no	*
extended_commands	281K	yes	*
spellcheck	468K	yes	*
doc_prep	718K	yes	*
networking_plus	615K	yes	*
comm	327K	yes	*
name_server	315K	yes	*
accounting	73K	yes	*
disk_quotas	76K	yes	*
audit	176K	yes	*
advanced_admin	544K	yes	*
mail_plus	116K	yes	*
old_commands	145K	yes	*
plot	229K	yes	*
sysV_commands	1853K	yes	*
man_pages	5114K	yes	*
games	2363K	yes	*
base_devel	5399K	yes	**
sysV_devel	3712K	yes	**
proflibs	2410K	yes	**
plot_devel	210K	yes	**
sccs	390K	yes	**
sunview_devel	1996K	yes	**
help_guide	213K	yes	**
dos_net_toolkit	566K	yes	**
config	2795K	yes	**

Where

* represents Applications Supplement clusters (total size = 13463K).

** represents Developers Toolkit clusters (total size = 17692K).

NOTE: CoreOS + Applications Supplement + Developers Toolkit = 52821K. At this point, two decisions must be made:

A. What optional Application SunOS and Developers Toolkit clusters are going to be loaded?

B. Where will the Sun386*i* software be placed? By default, the software will be placed in the directory `/export/sun386install`.

NOTE: Loading optional clusters on a server mean that those clusters can be installed on a Sun386*i*. The actual clusters installed on each Sun386*i* is determined at installation time.

To load the Sun386*i* software on a server system, follow the instructions below:

1. Login as root to the system.

2. Extract the necessary scripts from the tape list shown below:

   FOR LOCAL TAPE:

   ```
   # cd /usr/etc/install
   # mt -f /dev/nr<tape> rew
   # mt -f /dev/nr<tape> fsf 3
   # tar xvpbf 64 /dev/nr<tape>
         .
         .
         .
   ```

   FOR REMOTE TAPE:

   ```
   # cd /usr/etc/install
   # rsh <host> mt -f /dev/nr<tape> rew
   # rsh <host> mt -f /dev/nr<tape> fsf 3
   # rsh <host> dd if=/dev/nr<tape> bs=64k | tar xvpBbf 64 -
         .
         .
         .
   ```

   Where <*host*> is the name of system with the tape drive and <*tape*> is either `mt8` for 1/2" tapes or `st8` for 1/4" tapes.

3. Run the extract script to load the Sun386*i* distribution onto the file system. The following information will appear:

```
tape size (1/4" or 1/2") of distibution tapes
location of tape drive (local or remote)
the directory where you want to load the Sun386i distribution
what optional clusters you wish to load
```

NOTE: When loading from a remote tape drive, make sure that the system with the tape drive allows root equivalence to the Sun386*i* via the `/.rhosts` file.

Example:

```
# cd /usr/etc/install/sun386
# extract_dist
What size are your distribution tapes?
  1 - 1/4"
  2 - 1/2"
Enter [1-2]: 2
Are you loading from a remote tape drive? (y/n): y
Enter the remote host name: wilma
Enter the path of the sun386 distribution [/export/sun386install]:
Do you want to load all the Application SunOS optional cluster? (y/n): y
Do you want to load all the Developers Toolkit optional cluster? (y/n): n
Insert Application SunOS tape 1, confirm (y/n): y
      . . .
Insert Developers Tookit tape 1, confirm (y/n): y
      . . .
#
```

Where **bold** represents the example input.

4. Make sure that the directory containing the Sun386*i* distribution is exported *read-only* with root equivalence.

Installing Sun386*i* SunOS 4.0.1 from a Network File System

Install Sun386*i* SunOS 4.0.1 from a network file system by following the steps below:

1. Insert the *Remote Installation Diskette Number 1* and reboot.

2. The following message will appear:

   ```
   Insert Application SunOS diskette 2, confirm (y/n):
   ```

   Insert *Remote Installation Diskette Number 2* and type **y**

3. If there is more than one disk, the following message will appear:

   ```
   Which disk do you want to be your system disk?
     1 - the one in the system unit (sd2)
     2 - the one in the expansion unit (sd0)
     3 - shutdown
   Enter [1-3]:
   ```

   Enter **1** or **2**, depending upon which disk will be designated as the system disk.

4. The following prompt will appear:

```
How do you want to install SunOS?
  1 - from a network file system
  2 - from a remote tape drive
  3- shutdown
Enter [1-3]:
```

Type **1**.

5. Enter the host name and INET address of the remote system and the Sun3*i*. The prompts and examples of the required input are shown below:

```
Enter the host name of the remote system: fred
The remote system is fred, correct? (y/n): y
Enter the INET address of fred:192.9.200.10
The INET address of fred is 192.9.200.10, correct? (y/n): y
Enter the host name of this system: barney
This system is barney, correct? (y/n): y
Enter the INET address of barney:192.9.200.20
The INET address of barney is 192.9.200.20, correct? (y/n): y
```

Where **bold** represents the example input.

6. At the prompt below, specify the path name of the Sun386*i* distribution on the remote host:

```
Enter the path of the software on system barney: /export/sun386install
```

Where **bold** represents the example input.

7. Finally, at the prompts, specify which Applications Supplmement and Developers Toolkit clusters will be loaded. You can load all or selected Applications Supplement or Developers Toolkit clusters.

Example:

```
Do you want to load all the Application SunOS optional cluster? (y/n): y
Do you want to load all the Developers Toolkit optional cluster? (y/n): n
Do you want to load any Developers Toolkit optional clusters? (y/n): y
Do you want to load the base_devel cluster? (y/n): y
Do you want to load the sys_devel cluster? (y/n): n
     .
     .
     .
Do you want to load the config cluster? (y/n): n
```

Where **bold** represents the example input.

8. When the installation completes, the following message appears:

```
Installation complete.
Remove the diskette from the drive and power cycle system.
      .
      .
      .
>
```

RPC Source

Remote Procedure Call (RPC) Source

Remote Procedure Call (RPC) source, which includes the code of `portmap`, is a freely licensed source package copyrighted by Sun Microsystems. RPCSRC 4.0, which is based on SunOS 4.0, includes eXternal Data Representation (XDR), a specification for portable data transmissions which provides, in conjunction with RPC, a standard I/O library for interprocess communication. This gives programmers a standardized access to sockets without concern about the low-level details of the `accept()`, `bind ()`, and `select()` procedures. Sun RPC/XDR version 4.0 is now available via anonymous `ftp` from `bcm.tmc.edui` and from the `archive-server@bcm.tmc.edu`.

Using the Archive Server

To use the archive server, send mail to `archive-server@bcm.tmc.edu` with `Subject:` **`send nfs index`** for a list and description of all files in the archive. There are 17 shar files in RPCSRC and 4 for secure RPC. Note that the secure RPC package does not have any DES code due to export restrictions.

The archive will email the index shar file. Unpack the shar file by following the steps below:

1. Remove everything above the `#! /bin/sh` line.
2. Save the resulting text in a file.
3. Unpack the file by typing **`sh`** *filename*, where *filename* is the name of your file. It will create the Index file in your current directory.

RPCSRC

`RPCSRC` contains the following information.

- `RPC/XDR` library
- `portmapper`
- Nearly the entire *Network Programming* manual in `nroff -ms` format
- `rpcgen`
- Three demo RPC services
- One real RPC server (the `rpc.rstatd` service) and client
- The *rpc(3)* and *xdr(3)* manual pages
- All the RPC protocol definition files found in `/usr/include/rpcsvc`

- Secure RPC (except DES)

Contents of Index File — The Index contents appear below.

aliases	The current aliases list for NFS-LIST
inbox	Current Mailbox of NFS-LIST
inbox.1	Postings to NFS-LIST from 2/10/88 to 10/28/88
inbox.2	Postings to NFS-LIST from 10/29/88 to 2/2/89
pc-rdate.c	The source code for "rdate" on the PC under PC/NFS. This version is known to compile under MSC 5.10.
pc-rdate.exe.uu	The compiled version of "rdate" as compiled under MSC 5.10.
popd.c	The source code for the pop-2 daemon useful with PC/NFS Lifeline mail. [This is UNIX source code.]
rpc_39-40.01.shar	RPC SRC Version 3.9 to Version 4.0 patch kit (3 parts)
rpc_39-40.02.shar	
rpc_39-40.03.shar	
rpc_40.01.shar	RPC SRC Version 4.0 (17 parts)
rpc_40.02.shar	
rpc_40.03.shar	
rpc_40.04.shar	
rpc_40.05.shar	
rpc_40.06.shar	
rpc_40.07.shar	
rpc_40.08.shar	
rpc_40.09.shar	
rpc_40.10.shar	
rpc_40.11.shar	
rpc_40.12.shar	
rpc_40.13.shar	
rpc_40.14.shar	
rpc_40.15.shar	
rpc_40.16.shar	
rpc_40.16.shar	
rpc_40.17.shar	
secure_rpc.01.shar	The secure RPC sections of RPC/XDR for RPCSRC 4.0 (1)
secure_rpc.02.shar	The secure RPC sections of RPC/XDR for RPCSRC 4.0 (2)
secure_rpc.03.shar	The secure RPC sections of RPC/XDR for RPCSRC 4.0 (3)
secure_rpc.04.shar	The secure RPC sections of RPC/XDR for RPCSRC 4.0 (4)
sos.doc.shar	Documentation for LBL-SOS, an nfs server for DOS
sos.readme.shar	Important addendum to LBL-SOS
sos.rpc.1.shar	The RPC library for LBL-SOS part 1 of 3
sos.rpc.2.shar	The RPC library for LBL-SOS part 2 of 3
sos.rpc.3.shar	The RPC library for LBL-SOS part 3 of 3
sos.unfsd.1.shar	The NFS server code for LBL-SOS part 1 of 5
sos.unfsd.2.shar	The NFS server code for LBL-SOS part 2 of 5
sos.unfsd.3.shar	The NFS server code for LBL-SOS part 3 of 5
sos.unfsd.4.shar	The NFS server code for LBL-SOS part 4 of 5
sos.unfsd.5.shar	The NFS server code for LBL-SOS part 5 of 5
sos.exe.uu	LBL-SOS executable in uuencoded format
subjects.1	The Subject: lines from inbox.1
subjects.2	The Subject: lines from inbox.2

TAAC 2.3 Announcement

TAAC Software Release 2.3 Product Announcement

This article is a brief overview of Transcript Application ACcelerator software (TAAC). TAAC software release 2.3 is compatible with the TAAC hardware-equipped Sun-3 running SunOS 3.x, or any Sun-3 or Sun-4 system running SunOS 4.0.x. TAAC release 2.3 is distributed as either an upgrade or a full install release. Two versions are available: one for US domestic and one for overseas users. SunOS 4.0.3 is required for all new hardware platforms. TAAC release 2.3 includes support for the following new platforms:

- Sun-3/470 Deskside Workstation
- Sun-3/480 Server
- SPARCsystem 370 and 390 Deskside Workstations and Servers

TAAC release 2.3 replaces all other released code and contains the most current version of the TAAC-1 C compiler and linker. All existing customer code must be recompiled and relinked to be compatible with this release.

TAAC Release 2.3 Software Features

TAAC release 2.3 introduces the Advanced Rendering Toolkit (ART) for creating photo-realistic images of three-dimensional geometric data such as automotive and aircraft design, broadcast animation, molecular modeling, and scientific visualization. ART supports an advanced lighting model that includes specular highlights, multiple colored light sources, and infinite, point, and soft light source types. It also supports texture mapping, bump mapping, reflectance mapping, user-defined shading, translucency, and antialiasing. ART is script-driven, using routines from the graphics library and from the new advanced rendering library of shading and rendering functions.

TAAC release 2.3 introduces `voxcu` which renders volumetric data in volume clipping or ray casting mode. `voxvu` replaces the `cubevu` and `rayvu` programs from the 2.2 release and introduces the following new features:

- Support for volumes up to 64 Mbytes
- Limited support for 16-bit data
- Successive refinement with both low- and high-resolution passes
- `voxel` editing
- Movie Maker which rotates a volume and generates encoded frames to use with the `2bitmovie` utility.

The IP library which is called from the host, provides access to accelerated image processing directly from the host. This library allows programs executed on the Sun host to call image processing functions that execute on the TAAC-1. The IP library includes arithmetic and logical operations, morphological functions, Fourier transforms, geometric remapping, and filtering functions.

A subset of SunOS system calls and C function calls have been included on the release tape, although the library is unsupported in this release. These procedure calls include the following functions:

- `printf`
- `fopen`
- `gets`
- `scanf`
- `open`
- `read`
- `write`
- `close`

TAAC Release 2.3 Right To Use License

TAAC software release 2.3 is distributed with source to help users write software for TAAC-1 hardware and follow-on visualization products. A Right-to-Use License is included with the software that limits the use of any code derived from the TAAC software source, or the porting of that code to any equipment that does not include a TAAC-1 Accelerator. The Right-to-Use License also limits third parties from developing code based on TAAC software without obtaining a separate distribution license from Sun Microsystems, Incorporated.

TAAC Release 2.3 Documentation

Documentation for TAAC release 2.3 includes the following documents:

- *Read This First*
- *Release Notes for TAAC Release 2.3*
- *TAAC-1 Software Installation Guide*, part number 800-2441-12
- *TAAC-1 User Guide*, part number 800-2177-11
- *Change Pages*, part number 800-3712-10
- *TAAC-1 Software Reference Manual*, part number 800-3202-12

Sun386*i* Serial Cards

Configuring Serial Cards on the Sun386*i*

Many Sun customers are interested in configuring modem/system/AT serial cards on Sun386*i* machines. This article contains information of interest to those wanting both dial-in and dial-out capability from all of the added ports.

The configuration presented below permits both dial-in and dial-out capability from serial port A, `ttym0`, and `ttym1`. This configuration works well and gives Sun386*i* users fuller use of their system at a low cost. Please note that this configuration does not use SNAP to add the modem.

Equipment Used

The following equipment comprises the configuration described in this article.

- US Robotics Courier 2400 Modem
- Everex AT Multi I/O Card
- PC AT Serial Port Expansion Kit
  (provides the chip for the second serial port, `/dev/ttym1`)

AT Multi I/O Card Settings

The primary serial port is set to the values listed below.

1. `irq 4`
2. Memory base address `3f8`
3. This will be device `/dev/ttym0`

The secondary serial port is set to the following values.

1. `irq 3`
2. Memory base address `2f8`
3. This will be device `/dev/ttym1`

US Robotics Courier 2400 Modem Settings

The modem dip switch settings are shown below.

1. `sw1`: off, DTR normal
2. `sw2`: off, verbal result codes
3. `sw3`: on, display result codes

4. sw4: off, echo off-line commands
5. sw5: off, auto answer on ring

   NOTE: Testing has shown that having sw5 set to 'on, suppress auto answer' also works with no problems.

6. sw6: off, normal carrier detect
7. sw7: off, single phone connection
8. sw8: on, normal AT command set
9. sw9: on, on-line after +++
10. sw10: off, reserved
11. large unnumbered dip switch: up, pins 2 and 3 standard

Needed Changes to Files

You will need to make changes in the following three files: /etc/rc.local, /etc/ttytab, and /etc/remote files.

/etc/rc.local Changes

Uncomment the modload line for the ats.script as shown below. This is done to load the AT serial port driver.

```
modload ats.o -exec ats.script -conf ats_conf && chat 'AT serial port driver.'
```

/etc/ttytab Changes

Make the appropriate modification to show which port will have the modem attached. See the example for ttym1 below.

```
# @(#)ttytab    1.10 SMI ECD
#
# To allow root to log in, rather than forcing users to log in as
# themselves and SU, add the "secure" comment to the end of each
# line below; this means that your site is physically secure.
#
# name  getty                              type            status  comments
#
console "/usr/etc/getty -n -s -l std.9600" sun             on      secure
ttym1   "/usr/etc/getty D2400"             dialup          on
```

As root, restart init with the command shown below.

```
{spain:SUPERUSER:1} kill -1 1
```

/etc/remote Changes

Modify the dv parameter to show the alternate device of the port to which the modem is attached. Continuing with the above example, since /dev/cum1 is defined by /etc/modules/ats.script as the alternate device for /dev/ttym1, the /etc/remote appears as shown below.

Note the `dialers` entry in the below example.

```
{spain:8} cat /etc/remote
# @(#)etc.remote 1.8 88/05/25 Copyr 1987 Sun Micro
#
# This file is automatically adminstered by SNAP.
# You may make additions, but all comments will be stripped when
# this file is updated by SNAP.
# The 'UNIX' entry below is needed by SNAP.

tip0:tc=UNIX-2400:
UNIX:\
        :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#2400:tc=dialers:
UNIX-2400:\
       :el=^D^U^C^S^Q^O@:du:at=hayes:ie=#$%:oe=^D:br#2400:tc=dialers:
dialers:\
        :dv=/dev/cum1:
varsun:\
        :pn=14089422777:tc=UNIX-2400:
{spain:9}
```

Make sure to follow the AT serial card installation instructions in the *SunOS 4.0.1 Sun386i Administrator's & Developer's Notes*, pages 9-10.

Serial ports `/dev/ttya`, `/dev/ttym0`, and `/dev/ttym1` have been tested for dial-in and dial-out capability and work fine.

MS-DOS Communications Software

You can also configure the DOS windows on the Sun386*i* machine so that you can also use MS-DOS communications software to access the same port and modem that we defined above for UNIX.

First, DOS communications software has the capability to set dial-out and dial-in (autoanswer) modes within the DOS software. There is no need to set up one device for dialing out and another device for dial-up with a `getty` running on that device for logins. Only one device is required.

Second, the COM port you wish to be available for the DOS window must have permissions of at least 666. The COM port will now appear on the window's name stripe.

When a `getty` is running on a device, it sets the permissions to 622. If we were to have COM2 use `/dev/ttym1` as configured in the above examples, then COM2 would not be available for the DOS window due to permissions problems. A long listing of `cua0`, `cum0`, and `cum1` would show that these alternate devices have permissions of at least 666, with `cum0` and `cum1` having permissions of 777.

The AT serial card installation instructions in the *SunOS 4.0.1 Sun386i Administrator's & Developer's Notes* have you create a symbolic link in `/etc/dos/defaults/com2` to `/dev/ttym1` . Since we have a `getty`

running there for UNIX, we now know that defining COM2 this way while we are using that device for a UNIX `getty` would prevent DOS access for that device.

Instead, we can take advantage of the fact that the permissions on `/dev/cum1` are 777. Rather than creating a link to `/dev/ttym1`, we can set the link to `/dev/cum1` by using the command shown below.

```
% ln -s /dev/cum1 /etc/dos/defaults/com2
```

Configured this way, the DOS window comes up with COM2 available. However, once DOS has control of the port and the modem, UNIX cannot use them. In order to use the modem for UNIX, either detach the COM port that has the modem attached or quit the DOS window.

SCSI Configurations

SCSI Bus and Device Configuration Tables

This article presents a series of reference tables to facilitate putting together valid configurations of embedded Small Computer System Interface (SCSI) storage devices for the following new Sun systems:

- Sun-3/80
- SPARCstation 1
- SPARCserver 330

A supported Sun system should have *either* embedded SCSI devices *or* the ESDI/SCSI shoebox(es), but not both. Mixing the ESDI/SCSI shoebox on the same SCSI bus with embedded SCSI devices, either internally or externally, is not a Sun-supported configuration.

Sun-3/80 Systems

Device Description	Address Selection Method	SCSI Target ID	UNIX Device ID	Boot Device ID	External SCSI Termination	
					Type	Location
3/80 Base Unit's 1st Internal Disk	Preset Jumper	3	sd6	sd(0,18,0)	N/A	N/A
3/80 Base Unit's 2nd Internal Disk	Preset Jumper	1	sd2	sd(0,8,0)	N/A	N/A
Base Unit's Floppy Disk Drive	Preset Jumper	N/A	fd0	N/A	N/A	N/A
1st Desktop Disk Pack	Selector Switch	0	sd0	sd(0,0,0)	mini-SCSI	unused port of any Storage Pack
2rd Desktop Disk Pack	Selector Switch	2	sd4	sd(0,10,0)	mini-SCSI	unused port of any Storage Pack
3nd Desktop Disk Pack	Selector Switch	1	sd2	sd(0,8,0)	mini-SCSI	unused port of any Storage Pack
4th Desktop Disk Pack	Selector Switch	3	sd6	sd(0,18,0)	mini-SCSI	unused port of any Storage Pack
1st Desktop Backup Pack	Selector Switch	4	st0	st(0,0,0)	mini-SCSI	unused port of any Storage Pack
2nd Desktop Backup Pack	Selector Switch	5	st1	st(0,28,0)	mini-SCSI	unused port of any Storage Pack
ESM's 1st Disk Drive	Preset Jumper	0	sd0	sd(0,0,0)	D-SCSI	ESM's bottom SCSI port
ESM's 2nd Disk Drive	Preset Jumper	2	sd4	sd(0,10,0)	D-SCSI	ESM's bottom SCSI port
ESM's 1/4-inch Tape Drive	Preset Jumper	4	st0	st(0,0,0)	D-SCSI	ESM's bottom SCSI port
EEM's 1st Disk Drive	Preset Jumper	1	sd2	sd(0,8,0)	D-SCSI	ESM's bottom SCSI port

Key to Codes Used

ESM	External Storage Module (526G, 530G, 539G)
EEM	External Expansion Module (527G)
N/A	Not Applicable

Notes on the Sun-3/80 System

1. Two 'SCSI Target ID' values do not appear in the preceding table. The value '6' is reserved for future use by Sun. The SCSI host interface on the Sun-3/80 CPU board uses the value '7'.

2. The maximum number of Sun SCSI devices that the Sun-3/80 can support is six. The devices supported are up to two external tape drives and up to four disk drives (internal and external).

3. Options with the 'preset jumper' address selection method are not re-configurable by the user. As such, options with identical 'SCSI Target ID' values that use the 'preset jumper' scheme are mutually-exclusive options.

4. When Desktop Storage Packs (disk or tape) are connected to a system with internal disks and/or Expansion Storage Modules, a conflict of 'SCSI Target ID' values may occur. When a conflict exists, the SCSI Target ID value of the Desktop Storage Pack(s) should be changed to resolve the conflict.

5. Even though address selection schemes may seem to allow up to six external devices, the resulting physical length of the SCSI cables may prevent such a configuration from working. The next section provides the necessary data to verify that your configuration's overall cable length does not exceed six meters.

SCSI Subsystem Cable Lengths for the Sun-3/80 System

The physical cable length of a SCSI subsystem cannot exceed 6.0 meters. The following table specifies recommended cable lengths for each subsystem component.

Length (meters)	Device/Cable Description
0.3	Sun-3/80 CPU traces and Base Unit's internal cables
0.0	Desktop Storage Pack internal cable
0.8	Desktop Storage Pack external cable
1.0	External Storage Module's internal cable
2.0	ESM/EEM external cable
1.0	External Expansion Module's internal cable
0.4	ESM daisy-chain cable

Note: When both Desktop Storage Packs and External Storage Modules are connected to the same system, the Storage Packs are daisy-chained from the system's SCSI port and then the External Storage Modules are daisy-chained from the unused SCSI port of the last Storage Pack.

SPARCstation 1

Device Description	Address Selection Method	SCSI Target ID	UNIX Device ID	Boot Device ID	External SCSI Termination	
					Type	Location
SPARCstation 1's 1st Internal Disk	Preset Jumper	3	sd0	sd(0,0,0)	N/A	N/A
SPARCstation 1's 2nd Internal Disk	Preset Jumper	1	sd1	sd(0,1,0)	N/A	N/A
Base Unit's Floppy Disk Drive	Preset Jumper	N/A	fd0	N/A	N/A	N/A
1st Desktop Disk Pack	Selector Switch	0	sd3	sd(0,3,0)	mini-SCSI	unused port of any Storage Pack
2nd Desktop Disk Pack	Selector Switch	2	sd2	sd(0,2,0)	mini-SCSI	unused port of any Storage Pack
3rd Desktop Disk Pack	Selector Switch	1	sd1	sd(0,1,0)	mini-SCSI	unused port of any Storage Pack
4th Desktop Disk Pack	Selector Switch	3	sd0	sd(0,0,0)	mini-SCSI	unused port of any Storage Pack
1st Desktop Backup Pack	Selector Switch	4	st0	st(0,0,0)	mini-SCSI	unused port of any Storage Pack
2nd Desktop Backup Pack	Selector Switch	5	st1	st(0,1,0)	mini-SCSI	unused port of any Storage Pack
ESM's 1st Disk Drive	Preset Jumper	0	sd3	sd(0,3,0)	D-SCSI	ESM's bottom SCSI port
ESM's 2nd Disk Drive	Preset Jumper	2	sd2	sd(0,2,0)	D-SCSI	ESM's bottom SCSI port
ESM's 1/4-inch Tape Drive	Preset Jumper	4	st0	st(0,0,0)	D-SCSI	ESM's bottom SCSI port
EEM's 1st Disk Drive	Preset Jumper	1	sd1	sd(0,1,0)	D-SCSI	ESM's bottom SCSI port

Key to Codes Used

ESM	External Storage Module (526H, 530H, 539H)
EEM	External Expansion Module (527H)
N/A	Not Applicable

Notes on the SPARCstation 1 System

1. Two 'SCSI Target ID' values do not appear in the preceding table. The value '6' is reserved for future use by Sun. The SCSI host interface on the SPARCstation 1 CPU board uses the value '7'.

2. The maximum number of Sun SCSI devices that the SPARCstation 1 can support is six. The devices supported are up to two external tape drives and up to four disk drives (internal and external).

3. Options with the 'preset jumper' address selection method are not re-configurable by the user. As such, options with identical 'SCSI Target ID' values that use the 'preset jumper' scheme are mutually exclusive options.

4. When Desktop Storage Packs (disk or tape) are connected to a system with internal disks and/or Expansion Storage Modules, a conflict of 'SCSI Target ID' values may occur. When a conflict exists, the SCSI Target ID value of the Desktop Storage Pack(s) should be changed to resolve the conflict.

5. Even though address selection schemes may seem to allow up to six external devices, the resulting physical length of the SCSI cables may prevent such a configuration from working. The next section provides the necessary data to verify that your configuration's overall cable length does not exceed six meters.

SCSI Subsystem Cable Lengths for the SPARCstation 1

The physical cable length of a SCSI subsystem may not exceed 6.0 meters. The following table specifies recommended cable lengths for each subsystem component.

Length (meters)	Device/Cable Description
0.3	SPARCstation 1 CPU traces and Base Unit's internal cables
0.0	Desktop Storage Pack internal cable
0.8	Desktop Storage Pack external cable
1.0	External Storage Module's internal cable
2.0	ESM/EEM external cable
1.0	External Expansion Module's internal cable
0.4	ESM daisy-chain cable

Note: When both Desktop Storage Packs and External Storage Modules are connected to the same system, the Storage Packs are daisy-chained from the system's SCSI port and then the External Storage Modules are daisy-chained from the unused SCSI port of the last Storage Pack.

SPARCserver 330

Device Description	Address Selection Method	SCSI Target ID	UNIX Device ID	Boot Device ID	External SCSI Termination	
					Type	Location
SPARCsystem 330's 1st Internal Disk	Preset Jumper	3	sd6	sd(0,18,0)	mini-SCSI	330 Unit's SCSI port
SPARCsystem 330's 2nd Internal Disk	Preset Jumper	1	sd2	sd(0,8,0)	mini-SCSI	330 Unit's SCSI port
SPARCsystem 330's Internal Tape	Preset Jumper	4	st0	st(0,0,0)	mini-SCSI	330 Unit's SCSI port
1st Desktop Backup Pack	Selector Switch	4	st0	st(0,0,0)	mini-SCSI	unused port of any Storage Pack
2nd Desktop Backup Pack	Selector Switch	5	st1	st(0,28,0)	mini-SCSI	unused port of any Storage Pack
ESM's 1st Disk Drive	Preset Jumper	0	sd0	sd(0,0,0)	D-SCSI	ESM's bottom SCSI port
ESM's 2nd Disk Drive	Preset Jumper	2	sd4	sd(0,10,0)	D-SCSI	ESM's bottom SCSI port
ESM's 1/4-inch Tape Drive	Preset Jumper	4	st0	st(0,0,0)	D-SCSI	ESM's bottom SCSI port
EEM's 1st Disk Drive	Preset Jumper	1	sd2	sd(0,8,0)	D-SCSI	ESM's bottom SCSI port

Key to Codes Used

ESM	External Storage Module (526A, 530A, 539A)
EEM	External Expansion Module (527A)

Notes on the SPARCserver 330 System

1. Two 'SCSI Target ID' values do not appear in the preceding table. The value '6' is reserved for future use by Sun. The SCSI host interface on the SPARCstation 330 CPU board uses the value '7'.

2. The maximum number of Sun SCSI devices that the SPARCstation 330 can support is six. The devices supported are up to two external tape drives and up to four disk drives (internal and external).

3. Options with the 'preset jumper' address selection method are not re-configurable by the user. As such, options with identical 'SCSI Target ID' values that use the 'preset jumper' scheme are mutually exclusive options.

4. When a Desktop Backup Pack is connected to a system containing an internal 1/4" tape drive or an Expansion Storage Module with a 1/4" tape drive, a conflict of 'SCSI Target ID' values will occur. When a conflict exists, change the 'SCSI Target ID' value of the Desktop Backup Pack to '5'.

5. Even though address selection schemes may seem to allow up to six external devices, the resulting physical length of the SCSI cables may prevent such a configuration from working. The next section provides the necessary data to verify that your configuration's overall cable length does not exceed six meters.

SCSI Subsystem Cable Lengths for the SPARCserver 330

The physical cable length of a SCSI subsystem may not exceed 6.0 meters. The following table specifies recommended cable lengths for each subsystem component. If no internal SCSI options are present, the cable for the external option(s) connects directly to the CPU SCSI port and not to the chassis' SCSI 'OUT TO EXTERNAL' port. Therefore, the Pedestal Unit's Internal Cable length is used for calculations only when internal options are present.

Length (meters)	Device/Cable Description
0.0	SPARCstation 330 CPU's trace length
2.0	Pedestal Unit's internal cables (calculate only when internal option(s) are installed)
0.0	Desktop Backup Pack internal cable
0.8	Desktop Backup Pack external cable
1.0	External Storage Module's internal cable
2.0	ESM/EEM external cable
1.0	External Expansion Module's internal cable
0.4	ESM daisy-chain cable

Note: When both Desktop Storage Packs and External Storage Modules are connected to the same system, the Storage Packs are daisy-chained from the system's SCSI port and then the External Storage Modules are daisy-chained from the unused SCSI port of the last Storage Pack.

SunOS 4.1 Filesystems

New Directory Layout Policies in SunOS 4.1

This document outlines the current directory layout policies adopted in SunOS 4.1.

IMPORTANT NOTE:

Please note that the material presented in this article is *for informational purposes only* at this time, and is subject to change without notice.

Technical support will be available only after product availability. Therefore, please withhold your calls on SunOS 4.1 to either USAC or to your local service center until that time.

The actual date for SunOS 4.1 availability and technical support will be announced in an upcoming STB issue.

Scope

The major topics include architecture naming conventions, file sharing strategies, SunOS release naming conventions, directory structures, multi-release heterogeneous server configuration and multi-architecture kernel configuration.[2]

Architecture Naming Conventions

Recently, Sun has introduced a number of new hardware platforms which may be viewed as being largely compatible with existing Sun hardware and software products. The key word here is *largely*, since significant differences exist in these new hardware platforms which must be specifically addressed by the system software. Some examples of these new platforms are the Sun-3x and the Sun-4c.

Both of these new platforms are generally capable of executing Sun-3 and Sun-4 applications programs without adjustment or modification. The main differences between these new platforms and their related predecessors is that they have different kernel and virtual memory (`kvm`) access methods that require new software interfaces. In practice this means that the kernels and the programs which access the kernel data structures (e.g. `ps`) must take these differences into account. This hybrid model of compatibility separates the system software into two generic categories: applications software and hardware, or implementation-specific software.

The differences mentioned above give rise to two new architectural nomenclatures which are applied to each individual Sun hardware platform and similarly, to each copy of SunOS. With this naming convention, any given platform will have both an *application architecture* that conveys which family of

[2] This article is contributed by Fred Lloyd, Sun Microsystems, Inc., 2550 Garcia Ave., Mountain View, California 94043.

user-level programs that it may run and an *implementation architecture* which conveys the type of `kvm`-specific programs which are required by its particular hardware implementation. As an example, the Sun-3 and Sun-3x machines both have the same type of application architecture, which for reference purposes is called Sun-3.[3]

On the other hand, their respective implementation architectures are distinctly different. For naming purposes, implementation architectures are simply referred to by the machine's basic model type, such as Sun-3 or Sun-3x. The term used to denote a particular combination of application architecture and implementation architecture is *architecture pair*. Architecture pairs are to be used to categorize both the attributes of a particular platform as well as those of a particular OS release tape. For the purposes of definition, when two architecture pairs have identical application architectures they are said to have *compatible application architectures*.

SunOS Release Naming Conventions

As of SunOS release 4.1, it will become possible for a server to provide more than one version of SunOS to its different clients. In order for this to take place, a set of unambiguous naming conventions must be employed so that systems administration software can distinguish between the various versions of the SunOS that are or may be installed on the server.

The naming scheme which is used, combines the architecture used, the word `sunos` and the numerical release level identifier into a single string. In practice, this will consist of the application architecture followed by the word `sunos`, followed by a dot, and then the release number. For example, a 4.1 version of the SunOS for the Sun-4 denoted by the system software is shown below.

```
sun4.sunos.4.1.0
```

Server Layout Strategy

To provide a consistent environment for the various architectures to be served, it is necessary for server file systems to segregate binary executable code which is architecture-specific. From the system management point of view, it is most convenient if all files which are architecture-specific are located in directories which have the same names as the architectures themselves.

In accordance with existing SunOS conventions, such directories reside under the `/export/exec` point in the server's filesystem. To illustrate, a server separates Sun-3 and Sun-4 binaries in the following manner:

Example 1:

```
/export/exec/sun3
/export/exec/sun4
```

[3] Application architecture names such as Sun-3 and Sun-4 are presently being used for historical purposes to maintain compatibility with existing programs. This is apt to change, however, at a future date.

While the layout shown above could suffice when all of the server's clients were running the same version of the OS, it could not address the needs of a multiple-release heterogeneous server. To address this need, it becomes necessary to append the architecture name with an identifying tag which consists of a properly formatted release and realization name as discussed above.

By extending the logic shown in example 1, we then arrive at a suitable naming convention which a server could use to support both SunOS releases 4.0.3 and 4.1 for the same two architectures:

Example 2:

```
/export/exec/sun3
/export/exec/sun4
/export/exec/sun3.sunos.4.0.3
/export/exec/sun4.sunos.4.0.3
/export/exec/sun3.sunos.4.1
/export/exec/sun4.sunos.4.1
```

Note that in this example, the basic unqualified architecture names were retained. This is done to provide for the concept of a *default release*. The default release will be the release used by the system administration software whenever the system administrator adds a client or makes other changes without specifically indicating which OS release is to be used.

Also note that not all of the directories shown above actually exist, as several of them are symbolic links. This is made possible by the fact that much of what is used can be shared between compatible architectures.

Sharing Application Architecture Binaries

When two architecture pairs have a common application architecture, the user-level binary executable files may be freely shared between them. In practice, this means that Sun-3 and Sun-3x clients running the same release will normally mount identical copies of `/usr` filesystem images from their server. This file sharing can amount to a significant space savings on the server's disks.

To illustrate, let us use the same layout as used in example 2 but this time show how symbolic links are arranged to facilitate sharing as described above. As in the previous example, let us assume that the server being discussed is a Sun-4 which is running release 4.1 of SunOS.

Example 3:

```
/export/exec/sun3 -> sun3.sunos.4.1
/export/exec/sun4 -> /usr
/export/exec/sun3.sunos.4.0.3
/export/exec/sun4.sunos.4.0.3
/export/exec/sun3.sunos.4.1
/export/exec/sun4.sunos.4.1 -> /usr
```

Note how example 3 shows that `/export/exec/sun4.sunos.4.1` is a symbolic link to the server's own `/usr` filesystem. This is done because 4.1 is the release that the server itself is running and clients which need access to these files may simply share copies of them with the server.

In actual practice, compatible clients should never mount `server:/usr` but instead follow the `/export/exec/`*application_arch* model. Also pay particular attention to the fact that there is no corresponding `/export/exec/sun3x` directory. This is because the paths above represent *application architecture* files and Sun-3x machines have an application architecture of Sun-3.

Also, the example above shows how another release of the OS (in this case version 4.0.3) is supported. Since 4.0.3 is not the OS release that the server is running, whole copies (not symbolic links) for both Sun-3 and Sun-4 application architectures are required.

Implementation Architecture Files

While example 3 demonstrates how the application architecture files are shared, this next example will show how the implementation architecture files are handled. These files, which are packaged on the release tape in a module called 'Kvm', contain those programs and files which are implementation architecture-specific. On a running system, all of the `kvm` files may be found under the `/usr/kvm` mount point on a given system.

Starting with SunOS 4.0.3, the relative locations of several directories on target filesystems have been changed to better facilitate sharing. The list below details the relocations:

Example 4:

Old Location	New Location
/usr/kvm	(unchanged)
/usr/boot	/usr/kvm/boot
/usr/stand	/usr/kvm/stand
/usr/mdec	/usr/kvm/mdec
/usr/share/sys	/usr/kvm/sys

Note that the `/usr/share/sys` directory tree did not appear under `/usr/kvm` until release 4.1.

The relocation of the directories shown above make it possible for clients to use a single mount point for all of their implementation-architecture files. To aid in providing a smooth transition for users and software developers, the original `/usr/boot`, `/usr/mdec`, `/usr/stand`, and `/usr/share/sys` directories have been replaced with symbolic links which redirect both users and applications to the new `/usr/kvm` location.

Please be aware that none of the relocated directories are normally used by diskless clients during runtime. Instead, they are provided for the benefit of the server. The server must have access to the files in these directories in order to setup and configure diskless clients which will boot and run different versions of the OS in accordance with the ways that those versions are supposed to run. It is important to note that when a diskless client boots and has a different implementation architecture than its server, the client must explicitly mount a kvm tree in addition to its normal `root` and `/usr` trees.

Implementation-architecture files are arranged on a server in a manner which is similar to that used for application architectures. The base node in the server's file system for implementation architecture files is `/export/exec/kvm`. On the client, they are mounted on `/usr/kvm`. The example below represents a typical layout for a Sun-4 server's `/export/exec/kvm` area:

Example 5:

```
/export/exec/kvm/sun3 -> sun3.sunos.4.1
/export/exec/kvm/sun3x -> sun3x.sunos.4.1
/export/exec/kvm/sun4 -> /usr/kvm
/export/exec/kvm/sun3.sunos.4.0.3
/export/exec/kvm/sun3x.sunos.4.0.3
/export/exec/kvm/sun4.sunos.4.0.3
/export/exec/kvm/sun3.sunos.4.1
/export/exec/kvm/sun3x.sunos.4.1
/export/exec/kvm/sun4.sunos.4.1 -> /usr/kvm
```

The major difference between the layout of these files and the applications architecture files as shown in example 3 is that no sharing between types exists. The only possible sharing occurs when the client has the identical architecture pair as the server. This is illustrated by the symbolic links to `/usr/kvm`.

Client Mounting Parameters

Making use of these new layouts is particularly evident when viewed from the client's perspective. Below are a couple of examples of `/etc/fstab` from some typical clients. Note that not all entries in the below `/etc/fstab` example files are shown.

Example 6:
A Typical `/etc/fstab` *for a Sun-3 Client Running SunOS 4.1*

```
server:/export/root/client / nfs rw 0 0
server:/export/exec/sun3 /usr nfs ro 0 0
server:/export/exec/kvm/sun3 /usr/kvm nfs ro 0 0
```

Example 7:
A Typical `/etc/fstab` *for a Sun-4 Client Running SunOS 4.0.3*

```
server:/export/root/client / nfs rw 0 0
server:/export/exec/sun4.sunos.4.0.3 /usr nfs ro 0 0
server:/export/exec/kvm/sun4.sunos.4.0.3 /usr/kvm nfs ro 0 0
```

share Files

Another issue for a multi-release heterogeneous server is the architecture-independent file sharing area known as `/usr/share` . The items which fall into this category are those which are sharable between all architecture types supported by a given release. The architecture-independence of these files is because they are primarily ordinary ASCII text files. The specific modules involved include the man pages, the sharable portions of `/usr/lib` (mostly, but not limited to text processing macros), and the `/usr/share/src` areas.

Although the disposition of these files can be made without regard to machine architecture, significant portions of the files in these categories are subject to change from release to release. Clients will find the `/usr/share` tree appropriate to the release which they are running under the server's `/export/share` directory. For example, the `/export/share` directory of our example server would look like the next example when all modules are loaded.

Example 8:

```
/export/share/sunos.4.1 -> /usr/share
/export/share/sunos.4.0.3
```

Kernel Configuration on Heterogeneous Servers

For the purposes of realizing multi-architecture/multi-release support as outlined in this article, kernels and kernel configuration files are classified as being implementation-architecture specific. Under the provisions of this plan, all implementation architecture files (and therefore kernel config files) must reside under the mount point `/usr/kvm` .

In previous releases of SunOS 4.0, all kernel configuration took place in the `/usr/sys/`*arch*`/conf` directory. The new layout, however, has necessitated the relocation of these files to `/usr/kvm/sys` . Symbolic links will remain in the old locations to provide a migration path. The basic paths to the system configuration files, `/sys` and `/usr/sys` (which had both been symbolic links to `/usr/share/sys`) have been rearranged such that they now point to `/usr/kvm/sys` for the same reasons as before. Therefore, to configure a kernel, a user will simply change directories to either `/sys/`*arch*`/conf`, `/usr/sys/`*arch*`/conf` or `/usr/kvm/sys/`*arch*`/conf` and follow the regular method of configuring a kernel for their system.

It should be noted that client architecture subdirectories will no longer exist in the `/usr/sys` area of the server's directory layout space. Since these directories have been moved to `./kvm/sys`, they are now visible only to machines which mount the corresponding `/export/exec/kvm/`*client_arch* tree.

The operator of a server may, however, continue to edit the kernel configuration files of the clients by changing directories to `/export/exec/kvm/`*client_arch*`/sys` and making the appropriate changes. The actual kernel build, however, must take place on the client itself since cross-compilation is not currently supported. Because of this, a client wishing to

compile and link its own kernel must have read-write access permissions set up temporarily on the server's `/export/exec/kvm/`*client_arch*`/sys` directory, which is normally exported read-only.

Ongoing development may see the need for this temporary read-write access change. One may simply copy this directory to another (writable) place on the diskless client's filespace and then successfully build a kernel. In addition, some cleverly placed symbolic links might resolve the problem as well, without making it necessary to compromise the server by temporarily exporting its `/export/exec/kvm/`*client_arch* filesystem read-write. Incidentally, this problem has not been introduced in 4.1, as it has always been an area of concern on heterogeneous servers.

3

STB SHORT SUBJECTS

3

STB SHORT SUBJECTS

Window Errors

Windows and SunView Errors Defined

This short subject contains definitions for window device and SunView error conditions.

Window Devices Error

The error condition shown below results with problems with window devices.

```
ioctl #1C error
```

This error results from either the permissions being wrong on one or more of the `/dev/win*`, or an insufficient number of window devices in `/dev`.

As root, use the two below commands as one way to fix this error condition.

```
# chmod 666 /dev/win*
# MAKEDEV win0 win1 win2 win3
```

SunView Error Condition

You may see the below error message when `suntools` starts up.

```
open:  no such file or directory
```

This error condition may result from a missing shared font. Shared fonts are stored in `/usr/tmp` for those running SunOS 4.0. It may be that `/usr/tmp` is missing on the system. Note that `suntools` starts up successfully without the shared fonts by using one of its own.

SunOS 4.0.3 and SunLink

Upcoming SunOS 4.0.3 Upgrades and SunLink Products

Many customers will be upgrading to SunOS 4.0.3 during the upcoming weeks and months. They may be using *sunupgrade(8)* to do this.

Customers running SunLink products may see error messages resulting from a known bug, reference id number 1022502. `sunupgrade` may erroneously attempt to install `sunlink` as though it were a client implementation architecture.

sunupgrade(8) and SunLink Products: The Problem

The *sunupgrade(8)* utility treats all filenames, with three minor exceptions, of the form `/export/exec/*sun*` as though they are 'implementation architecture types'.

Servers running some of the SunLink products will typically have `/export/exec/sunlink` installed on them. This results (for most types of server upgrades) in the user being asked to do the following near the end of the upgrade.

```
Load tape volume 1 for sunlink and <RETURN>
```

Users attempting to follow the above instructions then receive several errors regarding the tape format.

The Workaround

Customers running SunLink products can temporarily rename all `/export/exec/*sun*` file or directory names *before* running *sunupgrade(8)*. Of course, this does not include the files for `sun2`, `sun3`, `sun3x`, `sun4`, or `sun4c`.

The temporary name must be chosen such that it no longer contains the string '`sun`'. Upper case '`SUN`' will work. When the upgrade has completed, restore the original file or directory names.

Aborting *sunupgrade(8)*

If the files were not renamed prior to beginning the upgrade, `sunupgrade` may be aborted with a Control-C at the time the bogus prompt for a tape is issued.

You then need to run the 'mop-up' script (`/usr/etc/upgrade/mop_up`). This should work in most cases and does seem to work in the case of `sunlink` . Note, however, if the bogus 'architecture type' name sorts out before any of the legitimate 'architecture types', upgrades for those architecture types will still need to be performed.

Distributed Databases

Distributed Databases

There has been repeated request for distributed database capabilities. This article defines distributed databases and outlines a few of the benefits and known drawbacks.

Distributed Databases Access

Distribute database access allows a customer to split the processing between machines to improve the bandwidth of database applications while preserving the centralization of data on the database server. Full access and update capabilities exist in this environment.

Distributed Databases Access Benefits

The reasons for distributing the database (for 95% of the environments) are as follows:

1. Remote sites share the same schema but have site-dependent data. This allows for widely used databases (e.g., corporate) to have a global view of all information, while each site can concern itself only with maintaining local information.

2. Redundancy of the database is critical. That is, if the machine on which the database resides goes down, the other machine can continue with its copy.

3. The implementation of schema design is most efficiently divided among different machines based on conditions such as I/O capacity and speed, machine CPU power, and network speed.

Distributed Databases Access Drawbacks

Possible reasons for not getting into a distributed database are as follows:

1. Not all update operations are supported under distributed DBMS' (Oracle included.) Full *Two phase commit* support for update consistency between DBMSs is not yet available. Oracle can be a participant in a two-phase commit operation, but cannot initiate it. Also, Oracle does not support dissecting a relation column-wise (where table R has columns C1, C2 C3 and C4; machine M1 has Columns C1 and C3 of R, and machine M2 has columns C2 and C4 of R).

2. Distributed databases access is difficult to design properly and maintain.

3. Performance of a distributed database is inherently slower than a centralized database. There are many variables that affect it (CPU speed, I/O bandwidth and speed, network speed and congestion, number of nodes into which the database is split, and so forth).

Using checkconfig

Using checkconfig

Those installing a system on large, heterogeneous networks may find problems from checkconfig when changing hostnames to a fully-qualified name and then rebooting. You may note that checkconfig may remove the directory /var/yp/*domainname* and then cause another reboot.

It is run in /etc/rc as /usr/etc/checkconfig -asr . Running this command on the command line yields the output shown below.

```
{triplering:SUPERUSER:22} checkconfig -asr
checkconfig: illegal option -- s
usage:  checkconfig [-afrvy] [-c category]
Legal categories for '-c' are:
ypservers:      Ensuring that this system is a YP server iff it is supposed to be one
(null): (null)
{triplering:SUPERUSER:22}
```

checkconfig and YP Security

You may see checkconfig remove directories and require another reboot as it helps enforce YP security. checkconfig finds out which machine is the YP master and which machines it thinks are its slaves. Then it checks what the booting system is trying to be. For example, if the booting system is trying to be a YP slave and the YP master says the booting system name is not authorized to be a slave, checkconfig assumes that the booting machine is someone trying to break into YP.

When checkconfig detects an apparent attempt to violate security, it takes immediate and drastic action to delete every type of YP information it can find on the local disk and then forces a reboot.

Renaming Machines

Those rebooting a system when only partially done renaming it may get caught by checkconfig . If this is inconvenient and you are sure that you are not worried about security, you can rename /usr/etc/checkconfig to something like /usr/etc/checkconfig- . This way checkconfig will not be executed.

Please note, again, that this workaround is for use only at installations where security is not of as much importance as at more sensitive installations.

Sun386*i* SNAP Restores

Sun386*i* SNAP and Restoring Files

Customers using the Sun386*i* SNAP `Select Files` function for restoring files may have difficulty when the number of files is about 200 files or more.

The problem is found when running Sun386*i* SunOS 4.0.1, when you are logged on as yourself, and you are the owner of the files. If you use the `Select Files` button, SNAP brings up the organizer and you are told that root (/) is an empty directory. If you look at the disk using the *bar(1)* utility, you will see the files.

The Problem Defined

When the user uses the `Select Files` button in SNAP's restore category, the organizer that is brought up lookings at the user's backup catalog, not the files on the backup media. This is a known bug, SDR reference ID numbers 5059 and 5060.

This problem occurs with SNAP backups containing a large number of files, typically about 200 files. The backup catalog becomes corrupted. The user cannot use SNAP `restore` to selectively restore files or recreate the backup catalog from that backup by using the [ADD ENTRY] button in SNAP's restore category.

The problem is caused by a bug in the `/usr/etc/backup` program and in the SNAP program. These programs store the names of the files in the backup in a memory buffer which is of a fixed size. When the list of filenames exceeds this limit, the programs fail to allocate more memory.

The Workarounds

Customers finding that SNAP's restore category shows an empty backup catalog can restore the backup using the [RESTORE ALL] button in SNAP's restore category or by using *bar(1)*.

Customers can either limit the number of files in each backup, or use the [RESTORE ALL] button in SNAP's restore category to restore all the files from the backup. Customers can also use the *bar(1)* command to selectively restore files.

Use the command shown below to selectively restore files from a SNAP backup on diskette.

```
bar xvfZp /dev/rfd0a filename1 filename2 ...
```

The below command is used to selectively restore files from a SNAP backup on tape.

```
bar xvfZp /dev/rst8 filename1 filename2 ...
```

FORTRAN Link Errors

FORTRAN Link Editor Error and Patch

Those customers with any architecture running SunOS 4.0 and Sun FORTRAN 1.1 may see the error shown below.

```
ld: Undefined _units in FORTRAN?
```

The Workaround

This is a known bug, reference ID number 1011308. The patch is named `system_libU77` and is available by calling the USAC or your local service office.

Bad `fseek`: An Error Definition

Customers have inquired regarding the meaning of the error message appearing below.

```
Bad fseek in es_file_read to position 835
```

This error comes from the `textsw` package. The error is generated when the `textsw` is trying to read characters from a file and the `fseek` call fails. The 'file' in this case is either the temporary file in `/tmp/*%` or the file associated with the `textsw` for `save`.

Those receiving this error should ensure that the `/tmp/*%` file is not removed. The original file may not be intact.

`cmdtool` Errors

`cmdtool` Graphics Problems and Solutions

This short subject contains information on two `cmdtool` problems and solutions.

The Disappearing `cmdtool` Problem

Some customers have observed a `cmdtool` disappear or dump core in SunOS 4.0 when running on a Sun-3/50.

The Solution

This is a known hardware bug. See bug 1010624. The fix is in Engineering Change Order (ECO) 3051. You need to log a hardware call.

The Dying `cmdtool` Problem

Other customers have seen a `cmdtool` die with the below message sent to the console window.

```
Child of cmdtool died due to signal 1
```

This error results from bug 1004140. A bug in the `pty` driver can cause a `cmdtool` to die with the console message `child of cmdtool died due to signal 1`. The problem occurs when a `cmdtool` is exited from the frame menu, but its associated `csh` is still running. When a new `cmdtool` is brought up and anything is typed in, the `cmdtool` dies because there are now two instances of `csh` running on the same `pty`.

The Workaround

The problem seems to occur most frequently when subprocess(es) of the `csh` in the first `cmdtool` are not terminated before choosing `Quit` on the frame menu. Thus, make sure all subprocesses have terminated before destroying the `cmdtool`.

`textedit` Shifting

`textedit` and Shifting Left

Customers using `textedit` to shift left may receive the error message shown below in the console window.

```
Argument value must be an integer!
```

This is a known bug when using the `shift_lines -t` option, BugTraq id number 1015959, and SDR id number 4946. If you try to shift left a section of code, the code will be *deleted*, but can be regained using the Undo key.

The Problem Defined

`textedit extras=>shiftlines` uses an undocumented option to `/usr/bin/shift_lines`. See *textedit_filters(1)*. The `-t` *number* option should shift each textline by *number* tabs. Note that the size of a tab is either the value specified in `~/.indent.pro` or eight spaces.

In theory, *number* can be either positive or negative. In practice, the code has a bug and for a negative number prints the message `Argument value must be an integer!` on the console and exits. The original text is deleted and `shift_lines` provides no new text, so it appears that all `shift_lines` did was delete the selection.

The Workaround

You can use the following work around. Copy the system-wide extras menu `/usr/lib/text_extras_menu` to `$home/.text_extras_menu`. Edit `$home/.text_extras_menu` by changing the below line as shown.

```
ORIGINAL LINE:  "Left"  shift_lines -t -1

NEW LINE:       "Left"  shift_lines -8
```

Use `defaultsedit` to change `Text/Extras_menu_filename` to `~/.text_extras_menu`. Then exit and restart `sunview`

If you are the machines administrator, you might prefer to edit the system-wide extras menu and tell people who like a tabsize of four spaces to create their own local version.

SPARCstation 1 Modems

SPARCstation 1 Modem/UUCP Configuration

This short subject explains how to turn on hardware carrier detect when modifying the kernel configuration file when configuring a SPARCstation 1 running SunOS 4.0.3 for modem and UUCP usage.

When the SPARCstation 1 is shipped, the EEPROM is programmed to have the serial ports configured to software carrier detect enabled. When modems are set up for dialin and dialout (for example, when using `tip` and `uucp`), the serial ports will need to sense hardware carrier detect from pin 8 directly.

When configuring a Sun-4 system (with `sun4` implementation architecture) running SunOS 4.0.3, the user sets the `flags` bit in the kernel configuration file to turn on hardware carrier detect, as follows:

```
device          zs0 at obio ? csr 0xf1000000 flags 3 priority
```

With the SPARCstation 1 (with `sun4c` implementation architecture) running SunOS 4.0.3, however, the user runs `/usr/kvm/eeprom` to first check to see what the current settings are, then changes the settings, if necessary.

To check the current settings, first enter `/usr/kvm/eeprom` as follows:

```
% /usr/kvm/eeprom
```

If the following appears, the serial ports `a` and `b` are set for software carrier detect, and will have to be changed to enable proper carrier detect for modems:

```
      .
      .
      .
ttyb-ignore-cd=true
ttya-ignore-cd=true
      .
      .
      .
```

To change these settings, enter the following:

```
% /usr/kvm/eeprom ttya-ignore-cd=false
% /usr/kvm/eeprom ttyb-ignore-cd=false
```

This sets up both ports to turn off the sensing carrier at pin 8. A separate kernel need not be rebuilt; just power-cycle the SPARCstation 1 system.

SunOS 4.1 User Guides

SunOS 4.1 and the Beginner's Guides

The Beginner's Guides will become the User's Guides for SunOS release 4.1 and subsequent releases. Please note that the new titles shown below do not contain any major content changes.

- *Getting Started*, part number 800-3830-01
- *SunView User's Guide*, part number 800-3831-01
- *Using the Network*, part number 800-3832-01
- *Doing More with SunOS*, part number 800-3833-01
- *Basic Troubleshooting*, part number 800-3834-01
- *Customizing SunOS*, part number 800-3835-01
- *Mail and Messages*, part number 800-3836-01

4

IN DEPTH

4

IN DEPTH

SunVideo Overview

SunVideo Overview

This article is an overview of the SunVideo hardware and software.

IMPORTANT NOTE:

Please note that the material presented in this article is *for informational purposes only* at this time, and is subject to change without notice.

Technical support will be available only after product availability (planned for Fall, 1989). Therefore, please withhold your SunVideo calls to either USAC or to your local service center until that time.

The actual date for SunVideo availability and technical support will be announced in an upcoming STB issue.

Supported Systems

SunVideo is supported on the following systems:

Sun-3 Systems:

- Sun-3/110
- Sun-3/150
- Sun-3/160
- Sun-3/180
- Sun-3/260
- Sun-3/280

- Sun-3/470

Sun-4 Systems:

- Sun-4/110
- Sun-4/150

SPARCservers:

- SPARCserver 330
- SPARCserver 370

Introduction

The SunVideo hardware and software products extends Sun's graphics offerings to deliver live motion, 24-bit true color video in a Sun window. SunVideo's non-interlaced high resolution output produces a clear video picture with no text or graphics flicker. Video sources can include videodisc players, Video Cassette Recorders (VCRs), cameras, or off-the-air broadcast materials. The SunVideo input/output board can also produce 525-line video suitable for display on a television monitor, recording on a VCR, or printing on a hard copy device, such as a video printer. SunVideo has a bank of input and output connectors to accept most of the popular video formats.

SunVideo offers an extensive set of menu-driven library functions which can be called from the host to manipulate every aspect of real-time video images, such as size, position, brightness, contrast, and color. SunVideo software allows users to easily create image databases and select video sources. Additionally, users can gain access to a control plane to overlay text and computer-generated graphics onto a video image. The software tightly integrates video into Sun's windowing system.

SunVideo Features

SunVideo contains the features listed below.

- 24-bit, live-motion video in a Sun window
- Windows can be scaled from 1/16x (icon size) to 2x (full screen)
- Flexible 525-line video input/output formats
- On triple-height VME board to minimize slot requirements
- Real-time brightness, contrast, and color controls
- Realtime frame capturing and freeze frame capabilities
- Genlock output to allow users to combine and record two video signals

SunVideo Operation

SunVideo manages the display of two video sources on the screen: high-resolution video originating from the native frame buffer, and a 525-line external source, such as a camera, VCR, or videodisc player. The 525-line video is digitized and held in memory on the SunVideo board. The digitized frames are read from memory and encoded into Sun high-resolution RGB video rates through a process called *upconversion.* SunVideo uses a pixel-accurate analog switch to mix the Sun high-resolution video coming from the native frame buffer with an 'upconverted' version of the low-resolution 525-line source. Thus, SunVideo intercepts the analog RGB signals going to the color monitor from the standard frame buffer, then inserts the live video image into a window boundary defined by the window system software at the appropriate time. The shape and size of the window and the visual characteristics of the image are controlled by the SunVideo software. The SunVideo software interprets window system parameters for the hardware, so the video window follows the same rules as text or graphic windows.

Additionally, digitized frames can be read out of memory and encoded into 525-line video formats for display on a television monitor, or for recording onto a video tape.

Hardware

The SunVideo board accepts four different live video formats: NTSC, Y/C (S-Video), YUV, and Red-Green-Blue (RGB). PAL and SECAM formats are not supported. NTSC, a television standard in the United States, Japan, and several other countries, is by far the most popular of these formats today. Y/C (S-Video) is a new video tape play standard, offering far higher resolution than conventional VHS technology. RGB component video is used extensively in graphic arts and animation application. YUV is used extensively in professional broadcasts.

Input Decoding

The NTSC signal is decoded into its luminance (black and white) component, Y, and its modulated color component, C, by a comb filter Y/C separator. Variable gain amplifiers are provided to adjust the luminance gain (contrast) and chroma gain (saturation). In the case of Y/C (S-Video) inputs, the luminance and modulated color components are switched into the video signal path following the comb filter. The color component is then demodulated into R-Y (defined notationally as V) and B-Y (defined notationally as U) color difference signals. In the case of YUV inputs, the luminance and color difference signals are switched in following the demodulator.

The luminance and color difference (YUV) signals are then matrixed into RGB. In the case of RGB inputs, the external RBG signals are switched in after the YUV-to-RGB matrix. The RGB signals are clamped to restore the black level, and buffered into three analog-to-digital converters. Software control of black level (brightness) and gain (contrast) is provided for each of the red, green, and blue signals. Non-volatile memory is provided for saving calibration settings.

Input Phase Locked Loop

The input phase locked loop should lock to signals with a wide range of stabilities. The phase locked loop processes a sync-stripped version of the user-provided 525-line source. The output of this phase locked loop is a clock that

controls the digitization speed of the A/D converters. This also informs the memory control unit of when and where to address the digitized video data.

The phase locked loop is designed to reliably handle signals from stable sources, such as broadcast video tape recorders; noisy sources, such as off-the-air signals; and unstable inputs from consumer video cassette recorders and laser disc players. The input phase locked loop produces the 12.1 MHz clock which the analog-to-digital converters use to digitize the RGB signals. The phase locked loop contains circuitry to limit errors introduced into the loop by noisy signals and missing sync pulses. The phase locked loop locks to the sync pulse of the input signal. The color demodulator has a separate phase locked loop, operating at 3.58 MHz, to produce the R-Y and B-Y color difference signals.

The High-Resolution Phase Locked Loop

To provide a display of live video on the Sun high-resolution monitor, SunVideo has a 92 MHz phase locked loop which locks to the output of the host color frame buffer located in the workstation. The output of this phase locked loop is a clock used to 'inform' the digital-to-analog converters and memory control units of when to encode pixel into 'Sun workstation' RGB video. Noise immunity is not necessary, since the native color frame buffer clock is provided by a crystal oscillator.

The Low-Resolution Phase Locked Loop

To provide 525-line 30Hz video, SunVideo has a 12.1 MHz phase locked loop which locks to the output of the native color frame buffer located in the workstation. The output of this phase locked loop is a clock used to 'tell' the digital-to-analog converters and memory control units when to encode pixels into RGB 525-line 30Hz video.

SunVideo Frame Buffer and Control Plane

SunVideo contains one and a half frames of image storage. The extra half frame (or one field) is provided to allow sychronizing the image to Sun frame rates during the vertical interval of the user-supplied 525-line video, thus avoiding a temporal mismatch during the active picture.

The memory is organized as three parallel banks of 8-bit-deep R, G, and B channels. The image size is 640 pixels in the horizontal direction by 512 lines in the vertical direction. The active picture is 640 by 480. The extra memory is provided to digitize test and control signals located in the vertical interval before the active picture start. Image data from the three analog-to-digital converters is clocked in at 12.1 MHz. The data is clocked out at one of two rates: 92 MHz when the video is displayed on the Sun monitor, and 12.1 MHz when the video is displayed on an external television monitor. Memory control state machines provide synchronization and scan conversion of the low speed input data to the high speed output rate. They also control size increase and decrease operations by performing pixel replication and decimation, respectively. Image display is independent of the host, except for window management and video picture parameter adjustments. The memory is accessible to the VME bus for both reads and writes.

A one-bit deep, 1152-pixel by 900-line memory (one bit for each visible pixel on a Sun color monitor), called the control plane, is provided for the window

software to define the viewable video window. The software writes the video window shape to the control plane. A fast analog switch inserts the live upconverted video into the Sun monitor signal path, based on the bit pattern written into the control plane.

SunVideo Output

SunVideo provides NTSC, Y/C, YUV, and RGB video output. The digital data from the memory is converted into R, G, and B by three digital-to-analog converters. As mentioned above, the digital-to-analog process operates at either 92 or 12.1 MHz. If the user chooses to display the video on the Sun monitor, the clock rate is 92 MHz. This video passes to a pixel accurate analog switch which mixes the native frame buffer video with the upconverted 525-line video. The position of this single pole, double throw switch is determined by the contents of a one bit deep 1152-pixel by 900-line control plane. For each pixel on the Sun workstation monitor, the contents of this control plane determines whether the switch is in the up (525-line video) or down (native frame buffer video) display mode.

If the user wishes to record the video on a VCR or display it on a television monitor, the three digital-to-analog converters run at 12.1 MHz. The resulting RGB signals are buffered and provided to the BNC connectors. Matrix amplifiers produce YUV, which are switched to the connectors under software control. NTSC and S-Video is produced in a one-chip modulator from RGB. SunVideo offers simple genlocking capability; the user can lock the 525-line video output to an externally-supplied video source. Refer to the section entitled 'Genlock', below.

SunVideo Software

SunVideo software includes the following:

- `videotool`
- SunView modifications
- Pixrect modifications
- Device Driver

Source is available for `videotool`; the remainder of the software is binary. `videotool` and SunView modifications are discussed briefly below. Refer to the *SunVideo Programmer's Manual* for further information.

`videotool`

`videotool` allows the user to exercise SunVideo through four pop-up windows: `video control`, `video adjust`, `video configuration`, and `image load/save`.

`video control`

The `video control` window consists of three buttons: `+zoom-`, `freeze`, and `on screen/video out`.

The `+zoom-` button changes the size of SunVideo frame buffer image from twice normal size (full screen) to 1/16 size (icon size) in power-of-2 increments.

The `freeze` button contains a `live/still` button, allowing the user to freeze the incoming video.

The `on screen/video out` button is used to toggle the window display to either a 525-line Sun workstation monitor (on screen), or on a separate 525-line monitor (video out).

`video adjust`

The `video adjust` panel consists of slider controls to adjust the picture quality: `luma gain`, `chroma gain`, `RGB gain`, and `RGB black level`. Luma gain controls the contrast of the black and white part of the NTSC and S-Video signals. Chroma gain controls the color intensity of the NTSE and S-Video signals. RGB gain controls the contrast for all inputs. RGB black level adjusts the brightness for all inputs.

`video configuration`

The `video configuration` panel contains the following selections: `input format`, `sync source`, `external video lock source`, and `chroma demodulation control`. `chroma demodulation control` has two modes: automatic, based on burst presence, or monochrome (color off). This panel also allows the user to customize a configuration, then save the configuration in a file.

`image load/save`

The `image load/save` panel contains buttons to load an image from a Sun raster file, or to save an image currently residing in the SunVideo frame buffer to a Sun raster file. Stored video images are kept in Sun's raster file format using the following parameters:

Image Size:	640 x 512
Image Depth:	32 bits, O-G-B-R format
Color Map:	None

SunView Modifications

Modifications made to SunView allow the programmer to create his or her own video windows and have full control over the board's configuration. All controls available through `videotool` are available to the SunView programmer. SunVideo requires pixrect modifications to provide full SunView 1.x compatibility. The usual `close`, `move`, `resize`, `expose`, `hide`, `redisplay`, `full size`, and `quit` options are provided. The video image is cropped to fit when the window is resized.

SunVideo Applications

SunVideo hardware and software has the flexibility needed to handle a wide range of multimedia applications. Some of the most common applications, including any necessary third-party equipment, are discussed below.

Command and Control: C31

SunVideo is well suited for a range of electronic surveillance applications. Video windows can be used to display real-time radar or sensor data. Live video can be used to give the viewer the ability to survey situations as they develop, from the Sun workstation.

Image Database

SunVideo is an integral part of an image database system. The user's choice of system components will depend on the size of the database, and whether still frames or motion video sequences are to be archived. If a still frame archive is being built, digital frame storage is preferable.

Because of disk and bus bandwidth constraints, uncompressed video cannot be retrieved digitally at realtime frame rates. Uncompressed 24-bit video images require nearly 1 MB per frame of storage. For motion video sequences, storage in analog on a write-once videodisc is recommended; one 12" videodisc platter can store 54,000 video frames.

Color Image Processing

SunVideo has a number of applications in image processing. SunVideo can be used for realtime manipulations of color, brightness, and contrast. SunVideo can also be used to perform format conversion between NTSC, S-Video, RGB, or YUV. The board can also be used as a general purpose frame grabber and display system, allowing material to be previewed and selected on the Sun workstation monitor. The image can then be processed on the host CPU, or on Sun's Transcript Application Accelerator (TAAC-1).

Videoteleconferencing

SunVideo allows the user to connect a Sun workstation to a third party videoteleconferencing network. A number of manufacturers offer desktop videoteleconferencing equipment, including Datapoint of San Antonio, Texas; Videotelecom of Austin, Texas; and Compression Laboratories of San Jose, California.

Supercomputer Image Distribution

It is estimated that an NTSE-quality video signal has 90 Mbits/sec of information. Full-motion digital video transmission is not practical in an Ethernet or Fiber Distributed Data Interface (FDDI) networking environment. SunVideo can be used to break the digital network bottleneck by distributing simulation images in analog over inexpensive coaxial cables.

Video Post-Production

SunVideo's input/output capabilities are ideal for video post-production. SunVideo can be combined with a paint package and an animation controller (used to control a videotape player) to create a personal post-production studio. A number of third-party packages exist, such as Artisan by Medialogic of Santa Monica, California, and MiniVas by Lyon Lamb Video Animation Systems, Incorporated, of Burbank, California.

Electronic Publishing

Video frames can be captured and downloaded into a document. Still frames are represented as a standard Sun 32-bit RGB raster file. Depending on the application, the user may need to convert the file into a different format.

SunVideo Application Tips

The following tips will streamline SunVideo application development.

Video Peripheral Control

To fully use the SunVideo card, the user must be able to interface and control video peripherals from the Sun workstation.

Videotape Recorder

To perform single frame recording, the user must be able to gain access to the controls of a videotape recorder from the Sun workstation. Because there are no standards for videotape recorder control, an animation controller is used to convert generic commands, such as 'play', 'stop', and so on into the required hex commands and handshaking for each machine. If the MiniVas from Lyon Lamb Video Animation Systems, Inc. is used, operations are performed over an RS-232 interface, allowing for automated recording, grabbing, searching, and other videotape recorder functions. Note that the videotape recorder must be supported by the selected animation controller.

Videodisc Players

Videodisc player control is important for computer-aided instruction and image databases. The following considerations apply to videodisc player control.

Video Connectors

SunVideo's 525-line video input/output connectors are of two types, as follows:

Input:	NTSC, RGB, YUV--BNC S-Video--4-pin DIN
Output:	NTSC, RGB, YUV--BNC S-Video-4-pin DIN

Users must provide the proper adapters to connect video sources with different connectors. These adapters are available through regular electronics sources.

Videodisc Mastering

Users who wish to produce their own interactive videodiscs need to work with a mastering house. Users can expect to pay from $300 to $500 for one-time mastering. Press masters typically cost around $5,000, and copies cost around $12 to $15 each (in quantities of 1-20 copies). These prices are in U.S. dollars.

Image Compression

When archiving images, the user may need to use compression to reduce the 1.3 MB of space required for each 32-bit 640 by 512 frame. It is important to keep in mind that compression can either be destructive or non-destructive. Destructive compression/decompression will not restore the original image; there is always some loss or distortion of information. Non-destructive compression/decompression will always restore the data exactly as it was. In general, higher compressions can be obtained with destructive techniques. The choice of technique will depend on the application. Users should keep in mind the tradeoff between compression and image quality.

Audio

For some applications, most notably computer-aided instruction, audio must be consider. Most video source materials, including videotapes, videodiscs, and off-the-air broadcasts have sound Frequency Modulated (FM) into the 6 MHz video channel at 4.5 MHz. Accordingly, VCRs, videodisc players and television tuners contain FM demodulators to strip off the audio. The baseband audio, available from a separate connector, can be used to drive a pair of miniature speakers. Miniature speakers are available from vendors such as Sony and Bose.

For users desiring digital audio capability, third party products are available. For example, Micro Technology Unlimited of Raleigh, North Carolina makes the

Digisound-16, a 16-bit audio conversion system with a Sun-3/Sun-4 system interface.

Genlock

Video recording applications may require SunVideo's 525-line output to match other 525-line sources at a switcher or mixer. This is accomplished by feeding a reference signal into the SunVideo input, selecting the external lock software button present on the Video Configuration panel, then adjusting the reference signal as required. The SunVideo output will follow the reference input signal timing. Subcarrier is maintained at a constant phase relative to sync, so a sync reference is sufficient for system timing; burst is not required.

Input/Output Formats

SunVideo input and output formats are shown below.

NSTC	1 volt p-p into 75 ohms BNC connector
S-Video	1 volt p-p luminance into 75 ohms 0.286 volt p-p burst into 75 ohms 4 pin mini-DIN connector
YUV	1 volt p-p luminance into 75 ohms 0.7 volt p-p U, V (75% color bars) BNC connector
RGB	0.714 volt p-p R,G,B into 75 ohms (noncomposite) or 1 volt p-p composite BNC connector

Input Only

Input-only information appears below.

External Sync	0.286 to 2 volts p-p into 75 ohms, into NTSC, Y (of Y/C), Y (of YUV), or G(of RGB) connector

Time Base Correction

Inputs may be from a stable (camera) source, videodisc player, or from a heterodyne video tape player. Vertical lock is not required. Phase locked loop will track input frequency variation of up to +/- 2%. The image will freeze on loss of horizontal or vertical sync, or improper field sequence.

Video Processing User Adjustments

Video processing user adjustments are described below.

Red, Green, and Blue gain
Red, Green, and Blue setup
Luminance gain (NTSC, S-Video only)
Chroma gain (NTSC, S-Video only)

Image Storage

The following table summarizes image storage information.

Spatial Resolution	640 pixel horizontally 512 lines vertically
Digitizing Start Point	line 525
Amplitude Resolution	256 levels for each R,G,B

On-Board Memory: 1.5 MB

Power Consumption: 18 A @ +5V, 2 A @ +12V, 1 A @ -12V

5

HINTS AND TIPS

5

HINTS AND TIPS

`printcap` Tips

Printing Tips: Your `printcap` File

This article contains some tips on printing and using your `/etc/printcap` file.

Printing Defaults

When you want to print a hardcopy of a file, the SunOS first looks to the `-P` option of the `lpr` command. Using `lpr -Ptrixie` specifies that the printer named `trixie` defined in your `/etc/printcap` file is the printer to be used.

If no printer name appears in the `lpr` command, the SunOS then looks to see if an environmental variable is set in your `.login` file in your home directory. For example, you can add the following line in your `~/.login` file to specify that the printer named `guttenberg` (again defined in your `/etc/printcap` file) is the printer to be used.

```
% setenv PRINTER trixie
```

Please note that you must now use the command shown below in order to have the above environmental variable take effect without having to first log out and then log back in.

```
% source ~/.login
```

Finally, if you have neither specified a printer name in the `lpr` command nor set the `PRINTER` environmental variable, the SunOS now defaults to looking for a printer named `lp` in your `/etc/printcap` file.

An Example printcap Entry

A typical /etc/printcap entry for a LaserWriter is shown below.

```
guttenberg|lp|trixie|eb2.laser|laserwriter in EB2 copy room:\
        :lp=:\
        :rm=scribe:\
        :rp=guttenberg:\
        :sd=/usr/spool/guttenberg:\
        :lf=/usr/spool/guttenberg/log:
```

In this example, there are four alias names for this printer, one of which is lp. The alias names are separated by the pipe symbol (|). Also note that the backslash (\ ) is used to continue the line (escape the non-printing line-feed character) so that the single-line file entry appears in a more readable form.

See the example again, this time with annotations.

Alias names for the printer seperated by pipe symbol
Backslash continues the line

```
   guttenberg|lp|trixie|eb2.laser|laserwriter in EB2 copy room:\
```

Device name for local printer (null for remote printer)

```
        :lp=:\
```

Name of host supporting the remote printer

```
        :rm=scribe:\
```

Name of the printer as defined on the remote host

```
        :rp=guttenberg:\
```

Local spool directory (mkdir /usr/spool/guttenberg)

```
        :sd=/usr/spool/guttenberg:\
```

Log file (touch /usr/spool/guttenberg/log)

```
        :lf=/usr/spool/guttenberg/log:
```

Note that the mkdir and touch commands shown in the above annotations are needed when setting up a new printer in your printcap file.

Example Printing Error Messages

You must have a spool directory that contains a log file and other files in order to print. The directory in /usr/spool has the same name as the printer and the log file is named log.

See the example below. As root, you change directories to /usr/spool and list the files and directories. Note that we are trying to use the printer named treebane in this example.

In order to show the error messages received, we change the name of the directory treebane to a temporary directory named fred.

We then use the lpq -P command to display the printing queue for printer treebane and see an error message from lpq when it cannot find the spooling directory which we just named fred.

We try to print a hardcopy of the `/etc/passwd` file, and see the error message from `lpr` when it cannot create a file named `.seq`, since that file is now in `fred`.

Finally, we move the temporary directory `fred` and its files back to `treebane` and successfully print a copy of the `passwd` file for one example. We then look again at the printer queue to see the new print job.

```
# cd /usr/spool
# ls
at           lpd         mqueue      rainfall    techprint
guttenberg   lpd.lock    news        rwho        treebane
lithograph   mail        obadiah     secretmail  uucp
log          micah       papertiger  sunwriter   uucppublic
# mv treebane fred
# lpq -Ptreebane
treebane is ready and printing
Rank   Owner      Job  Files                                 Total Size
active prc        71   standard input                        78207 bytes
lpq: treebane: cannot chdir to spooling directory
# lpr -Ptreebane /etc/passwd
lpr: cannot create /usr/spool/treebane/.seq
# mv fred treebane
# lpr -Ptreebane /etc/passwd
# lpq -Ptreebane
treebane is ready and printing
Rank   Owner      Job  Files                                 Total Size
active prc        71   standard input                        78207 bytes
1st    root       72   /etc/passwd                           327 bytes
```

6

THE HACKERS' CORNER

6

THE HACKERS' CORNER

Email Flags Up

Incoming Email: Is Your Mail Flag Up?

Some of those running `suntools` have had problems getting their mail flag up on the `mailtool` incon. This is a useful feature for those wanting to know if they have received any incoming email.

This month's **Hackers' Corner** contains a tool that will tell you not only when you get new email, but how many new email items you have, and how many unread items you have. It is called `checkmail` and can be run as a window tool in iconic form. You can run it automatically every 120 seconds as part of your `~/.suntools` file with a very large font.[4]

Please consult your local shell script or programming expert regarding any script or code problems. The example programs are not offered as a supported Sun product, but as items of interest to enthusiasts wanting to try out something for themselves. Note that **Hackers' Corner** code may not work in all cases, and may not be compatible with future SunOS releases.

Your `~/.suntools` Entry

Place the entry shown below into your `~/.suntools` file to run `checkmail` all of the time. Note that the below entry is shown on three lines for presentation purposes only. It should all be on one line in your file (without the backslashes).

```
shelltool -Wp 2 153 -Ws 481 224 -WP 600 0 -Wi \
-Wt /usr/lib/fonts/fixedwidthfonts/gallant.r.19 \
-Wl "suntool: checkmail" -WL "suntool: checkmail" checkmail -i 120
```

To run `checkmail` on the command line in the background with a 120 second cycle, type **`shelltool checkmail -i 120 &`**.

---

[4] The code contained in this **Hackers' Corner** is submitted by Yoon Kim, Manager, System Administration, Software Products Division; and Bill Petro, GSG Marketing, Sun Microsystems, Inc.

Installing the `checkmail` Shell Script

Save all of the code into a file named `neato`. Then type **`sh neato`**.

Three files will be created, `checkmail`, `mail.icon`, and `nomail.icon`. For those not wanting the two icons, you can uncomment the two lines shown below in the `checkmail` file.

```
#Havemail=/usr/include/images/mail.icon
#Nomail=/usr/include/images/nomail.icon
```

You must now type **`chmod +x checkmail`** and you are set.

The `checkmail` Shell Script

The `checkmail` shell script appears on the following pages. Note that it is in 'shar' format.

```
#! /bin/sh
# This is a shell archive.  Remove anything before this line, then unpack
# it by saving it into a file and typing "sh file".  To overwrite existing
# files, type "sh file -c".  You can also feed this as standard input via
# unshar, or by typing "sh <file", e.g..  If this archive is complete, you
# will see the following message at the end:
#       "End of shell archive."
# Contents:  checkmail checkmail.shar mail.icon nomail.icon
# Wrapped by rock@warp on Fri Jun 23 12:09:14 1989
PATH=/bin:/usr/bin:/usr/ucb ; export PATH
if test -f 'checkmail' -a "${1}" != "-c" ; then
  echo shar: Will not clobber existing file \"'checkmail'\"
else
echo shar: Extracting \"'checkmail'\" \(1462 characters\)
sed "s/^X//" >'checkmail' <<'END_OF_FILE'
X#! /bin/sh
X#
X# checkmail - checks for new mail and updates icon label
X#      with "new" or "unread" or both messages.
X#
X# syntax:  checkmail [ -i seconds ]
X#
X# 16.nov.87        Original code by yOOnkim@Sun.COM
X#               (Yoon Kim - SPD System Administration)
X# various       expansion, escape code cleanups, etc. by rock@Sun.COM
X#               (Bill Petro - GSG Marketing)
X# tip-of-the-hat   to kevin@Sun.COM
X#               (Kevin Sheehan - Consulting)
X#
X
Interval=60     # In seconds
X
case $1 in
X-i*)
X   if [ $# = 2 ]
X   then
X       Interval=$2
X   fi
X   ;;
esac
X
Havemail=/home/lala/rock/images/mail.icon#My arrow icon
Nomail=/home/lala/rock/images/nomail.icon#Dan Heller's MUSH icon
X#Havemail=/usr/include/images/mail.icon#or use the generic one
X#Nomail=/usr/include/images/nomail.icon#or use the generic one
X
XEsc=`echo blah | awk '{printf "%c", 27}'`
Close="echo -n ${Esc}[2t"
X
newLabel()
X{
X   echo -n "${Esc}]L$1${Esc}\\"
```

```
X}
X
newicon()
X{
X   echo -n "${Esc}]I$1${Esc}\\"
X}
X
newicon $Nomail
newLabel ""
X$Close
while true
do
X   Top=`echo x | mail | head -2 | egrep "new$|unread$"`
X   case "$Top" in
X   *new*unread)
X       label=`echo $Top | \
X           awk '{printf("%d/%d", $(NF-3), $(NF-1));}'`
X       newLabel "${label}"
X       newicon $Havemail
X       ;;
X   *new)
X       label=`echo $Top | awk '{printf("  %d", $(NF-1));}'`
X       newLabel "${label}"
X       newicon $Havemail
X       ;;
X   *unread)
X       label=`echo $Top | awk '{printf("%d_U", $(NF-1));}'`
X       newLabel "${label}"
X       newicon $Havemail
X       ;;
X   *)
X       newLabel ""
X       newicon $Nomail
X       ;;
X   esac
X   sleep $Interval
done
END_OF_FILE
if test 1462 -ne `wc -c <'checkmail'`; then
    echo shar: \"'checkmail'\" unpacked with wrong size!
fi
chmod +x 'checkmail'
# end of 'checkmail'
fi
if test -f 'checkmail.shar' -a "${1}" != "-c" ; then
  echo shar: Will not clobber existing file \"'checkmail.shar'\"
else
echo shar: Extracting \"'checkmail.shar'\" \(0 characters\)
sed "s/^X//" >'checkmail.shar' <<'END_OF_FILE'
END_OF_FILE
if test 0 -ne `wc -c <'checkmail.shar'`; then
    echo shar: \"'checkmail.shar'\" unpacked with wrong size!
fi
```

```
# end of 'checkmail.shar'
fi
if test -f 'mail.icon' -a "${1}" != "-c" ; then
  echo shar: Will not clobber existing file \"'mail.icon'\"
else
echo shar: Extracting \"'mail.icon'\" \(1933 characters\)
sed "s/^X//" >'mail.icon' <<'END_OF_FILE'
X/* Format_version=1, Width=64, Height=64, Depth=1, Valid_bits_per_item=16
X */
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0008,0x0000,0x0000,0x0000,0x000C,0x0000,
X   0x0000,0x0000,0x000E,0x0000,0x0000,0x0000,0x000F,0x0000,
X   0x0000,0x0000,0x000F,0x8000,0x0000,0x0000,0x000F,0xC000,
X   0x0000,0x0000,0x000F,0xE000,0x0000,0x0000,0x000F,0xF000,
X   0x0000,0x0000,0x000F,0xF800,0x0000,0x0000,0x000F,0xFC00,
X   0x0000,0x0000,0x000F,0xFE00,0x0000,0x0000,0x000F,0xFF00,
X   0x7FFF,0xFFFF,0xFFFF,0xFF80,0x7FFF,0xFFFF,0xFFFF,0xFFC0,
X   0x7FFF,0xFFFF,0xFFFF,0xFFE0,0x7E7C,0xF1E6,0x7FFF,0xFFF0,
X   0x7E38,0xE4E6,0x7FFF,0xFFF8,0x7E10,0xCE66,0x7FFF,0xFFFC,
X   0x7E44,0xCE66,0x7FFF,0xFFFE,0x7E6C,0xCE66,0x7FFF,0xFFFF,
X   0x7E7C,0xC066,0x7FFF,0xFFFF,0x7E7C,0xCE66,0x7FFF,0xFFFE,
X   0x7E7C,0xCE66,0x7FFF,0xFFFC,0x7E7C,0xCE66,0x7FFF,0xFFF8,
X   0x7E7C,0xCE66,0x03FF,0xFFF0,0x7E7C,0xCE66,0x03FF,0xFFE0,
X   0x7FFF,0xFFFF,0xFFFF,0xFFC0,0x7FFF,0xFFFF,0xFFFF,0xFF80,
X   0x0000,0x0000,0x000F,0xFF00,0x0000,0x0000,0x000F,0xFE00,
X   0x0000,0x0000,0x000F,0xFC00,0x0000,0x0000,0x000F,0xF800,
X   0x0000,0x0000,0x000F,0xF000,0x0000,0x0000,0x000F,0xE000,
X   0x0000,0x0000,0x000F,0xC000,0x0000,0x0000,0x000F,0x8000,
X   0x0000,0x0000,0x000F,0x0000,0x0000,0x0000,0x000E,0x0000,
X   0x0000,0x0000,0x000C,0x0000,0x0000,0x0000,0x0008,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,
X   0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000,0x0000
END_OF_FILE
if test 1933 -ne `wc -c <'mail.icon'`; then
    echo shar: \"'mail.icon'\" unpacked with wrong size!
fi
# end of 'mail.icon'
fi
if test -f 'nomail.icon' -a "${1}" != "-c" ; then
  echo shar: Will not clobber existing file \"'nomail.icon'\"
else
echo shar: Extracting \"'nomail.icon'\" \(1933 characters\)
sed "s/^X//" >'nomail.icon' <<'END_OF_FILE'
X/* Format_version=1, Width=64, Height=64, Depth=1, Valid_bits_per_item=16
```

```
X */
X   0x0000,0x0000,0x0000,0x0000,0x7FFF,0xFFFF,0xFFFF,0xFFFE,
X   0x7FFF,0xFFFF,0xFFFF,0xFFFE,0x7FFF,0xFFFF,0xFFFF,0xFFFE,
X   0x7FFF,0xFFFF,0xFFFF,0xFFFE,0x7FFF,0xFFFF,0xFFFF,0xFFFE,
X   0x7FFF,0xFFFF,0xFFFF,0xFFFE,0x7FFF,0xFFFF,0xFFFF,0xFFFE,
X   0x7FFF,0xFFFF,0xFFFF,0xFFFE,0x7FFF,0xE000,0x0000,0x01FE,
X   0x7FFF,0x01FF,0xFFFF,0xFE7E,0x7FF8,0xFE3F,0xFFFF,0xFF9E,
X   0x7FE7,0xFFCF,0xFFFF,0xFFEE,0x7FDF,0xFFF7,0xFFFF,0xFFEE,
X   0x7F3F,0x8FF9,0xFFFF,0xFFF6,0x7EFF,0x07FE,0xFFFF,0xFFF6,
X   0x7EFE,0x03FE,0xFFFF,0xFFFA,0x7DFE,0x03FF,0x7FFF,0xFFFA,
X   0x7BFE,0x03FF,0xBFFF,0xFFFA,0x7BFF,0x07FF,0xBFFF,0xFFFA,
X   0x77FF,0x8FFF,0xDFFF,0xFFFA,0x77FF,0xFFFF,0xDFFF,0xFFFA,
X   0x77FF,0xFFFF,0xCFFF,0xFFFA,0x6FFF,0xFFFF,0xEFFF,0xFFFA,
X   0x6FFF,0xFFFF,0xEFFF,0xFFFA,0x6C00,0x0000,0x6FFF,0xFFDA,
X   0x5DFF,0xFFFF,0x6FFF,0xFF1A,0x5DFF,0xFFFF,0x6FFF,0xFC5A,
X   0x5DFF,0xFFFF,0x6FFF,0xF9DA,0x5C00,0x0000,0x6FFF,0xE3DA,
X   0x5FFF,0xFFFF,0xEFFF,0xCBDA,0x5FFF,0xFFFF,0xEFFF,0x3BDA,
X   0x5FFF,0xFFFF,0xEFFC,0x7BDA,0x5FFF,0xFFFF,0xEFF9,0xFBDA,
X   0x5FFF,0xFFFF,0xEFE3,0xFBDA,0x5FFF,0xFFFF,0xEE0F,0xFBDA,
X   0x5FFF,0xFFFF,0xEC1F,0xFBBA,0x5FFF,0xFFFF,0xEC1F,0xFA7A,
X   0x5FFF,0xFFFF,0xEC1F,0xF9FA,0x5FFF,0xFFFF,0xEE3F,0xFBFA,
X   0x5FFF,0xFFFF,0xEFFF,0xFFF2,0x5BDE,0x7BD8,0x2FFF,0xFFEE,
X   0x59DD,0xB9DB,0xEFFF,0xFFDE,0x59DB,0xD9DB,0xEFFF,0xFF3E,
X   0x5ADB,0xDADB,0xEFFF,0xFEFE,0x5ADB,0xDADB,0xEFFF,0xF9FE,
X   0x5B5B,0xDB58,0x6FFF,0xF7FE,0x5B5B,0xDB5B,0xEFFF,0xC7FE,
X   0x5B9B,0xDB9B,0xEFFF,0xB7FE,0x5B9B,0xDB9B,0xEFFF,0x77FE,
X   0x5BDD,0xBBDB,0xEFFC,0xF7FE,0x5BDE,0x7BD8,0x2FFB,0xF7FE,
X   0x5FFF,0xFFFF,0xEFE3,0xF7FE,0x5FFF,0xFFFF,0xEFDB,0xF7FE,
X   0x5FFF,0xFFFF,0xEFBB,0xF7FE,0x5FFF,0xFFFF,0xEE7B,0xF7FE,
X   0x5FFF,0xFFFF,0xEDFB,0xF7FE,0x5FFF,0xFFFF,0xEBFB,0xF7FE,
X   0x4000,0x0000,0x07FB,0xF7FE,0x7FFF,0xFFF7,0xFFFB,0xF7FE,
X   0x7FFF,0xFFF7,0xFFFB,0xF7FE,0x7FFF,0xFFF7,0xFFFB,0xF7FE,
X   0x7FFF,0xFFF7,0xFFFB,0xF7FE,0x0000,0x0000,0x0000,0x0000
END_OF_FILE
if test 1933 -ne `wc -c <'nomail.icon'`; then
    echo shar: \"'nomail.icon'\" unpacked with wrong size!
fi
# end of 'nomail.icon'
fi
echo shar: End of shell archive.
exit 0
```

7

HARDWARE, CONFIGURATIONS, & UPGRADES

7

HARDWARE, CONFIGURATIONS, & UPGRADES

Software Release Levels

As of June 23, 1989

Operating Systems

Product Name	Current Release
SunOS	4.0.3
SunOS 386*i*	4.0.1

Communications Products

Product Name	Current Release
SunLink BSC3270 (SunOS 3.x)	3.0
SunLink BSC3270 (SunOS 4.x)	6.1
SunLink SCP	6.0
SunLink TE100	6.0
SunLink BSCRJE	6.0
SunLink Local 3270	6.1
SunLink SNA3270	6.1
SunLink Peer-to-Peer	6.0
SunLink IR	6.0
SunLink DDN	5.0
SunLink DNI	6.0
SunLink OSI	5.2
SunLink MCP	6.0
SunLink X.25	6.0
SunLink Channel Adapter SCA	6.0
SunLink CG3270	6.0
SunLink MHS	5.2
Notes: SunLink release 5.x products are only compatible with SunOS release 3.x. SunLink release 6.x products are only compatible with SunOS release 4.0.	

Unbundled Languages

Product Name	Current Release
Sun Modula-2 (Sun-2,3 and SunOS 3.x)	2.0
Sun Modula-2 (Sun-3,4,386*i* and SunOS 4.x)	2.1
Sun FORTRAN* (Sun-2,3)	1.0
Sun FORTRAN* (Sun-4 and Sys4-3.2)	1.05
Sun FORTRAN* (Sun-2 and SunOS 4.0)	1.1
Sun FORTRAN* (Sun 386*i* and SunOS 4.0)	1.1R
Sun FORTRAN* (Sun-3,4 and SunOS 4.0)	1.2
SPE for SCLisp 2.1	1.0
Sun Common Lisp-E	1.1
Sun Common Lisp-D	2.1
Sun Common Lisp-D (Sun-3, Sun-4)**	3.0
Cross Compilers (SunOS 3.x, Sys4-3.2)	2.0
Pascal*** (Sun-4 and Sys4-3.2)	1.05
Pascal*** (Sun-2,3,4,386*i* and SunOS 4.0)	1.1

Notes:

* The `f77` compiler is automatically included with SunOS Release 3.x, which includes SunOS Releases 3.2, 3.4, and 3.5. Sun FORTRAN 1.0 (for Sun-2,3 systems and SunOS 3.x), Sun FORTRAN 1.05 (for Sun-4 systems running Sys4-3.2), Sun FORTRAN 1.1 (for Sun-2,Sun386*i* systems and SunOS 4.0), and SunFORTRAN 1.2 (for Sun-3,4 and SunOS 4.0) are value-added products that support VMS extensions to the `f77` compiler, and must be purchased separately from the SunOS.
There is no bundled FORTRAN or Pascal for Sys4-3.2 or SunOS 4.0.

** Sun Common Lisp-D release 3.0 does not obsolete Sun Common Lisp release 2.1 at this time.

*** The `pc` (Pascal) compiler is automatically included with SunOs Release 3.x, which includes Release 3.2, 3.4, and 3.5. Sun Pascal 1.05 (for Sun-4 systems) and Sun Pascal 1.1 (for Sun-2, Sun-3, Sun-4 and Sun386*i* systems running SunOS 4.0) are value-added products that support many extensions to the `pc` compiler, and must be purchased separately from the SunOS.

Unbundled Graphics

Product Name	Current Release
SunGKS	2.2.1
SunPHIGS	1.0
Sun58TE	1.0

Unbundled Applications

Product Name	Current Release
SunSimplify	1.1
SunTrac (Sun-2 and Sun-3)	1.2
SunTrac (Sun-4)	1.0/3.2
SunIPC	1.1
Transcript	2.1
SunUNIFY	3.0
PC-NFS	3.0
SunAlis	2.1
SunINGRES (Sun-2 and Sun-3)	5.1

Other Products

Product Name	Current Release
NeWS	1.1
NSE	1.1

TOPS Network Products

Product Name	Current Release
TOPS for the PC	2.1
TOPS for the Sun Workstation (Sun-3, SunOS 3.5)	2.1
TOPS for the Sun Workstation (Sun-3, Sun-4, Sun386i, SunOS 4.X)	2.2
TOPS for the Macintosh	2.1
TOPS NetPrint	2.0

Current Sun Software Products and Release Levels

The preceding tables contain lists of current Sun software products and their respective current release levels.

You will note that the Software Technical Bulletin (STB) contains articles from time to time that detail technical changes in a given software product's next available release.

Please contact your sales representative if you decide that you would like to update the release level of a Sun software product you already use, or wish to purchase another product. Use the tables to determine whether your release is the current release level.

These tables appear monthly in the STB for your convenience.

Sun386*i* Serial I/O

Sun386*i* Serial I/O Software and Hardware

Customers running SunOS releases 4.x and Sun386*i* SunOS 4.x will find information in this article helpful regarding existing and upcoming Serial Input/Output (SIO) software and hardware capabilities.[5]

In one example application, a customer running a Sun386*i*/250C bought an Aurora Technologies 16-port serial port terminal MUX card, and installed it along with the UNIX device drivers. No `gettys` were running on any of the ports.

The customer tests to see if they can get output between the ports, so they hook up a serial cable between the Sun386*i* `ttya` port, and port `ttyb2` from the MUX card. Seeing nothing using `cat > /dev/ttya` and `cat < /dev/ttyb2` in another window, the customer adds a null modem link into the chain. Now they see output from one port with many characters dropped.

Cabling Complications

No problem was caused when using a standard 25-pin cable. However, a telephone modular RS-232 cable caused complications. This cable was a 4-wire cable, with two wires to ground and the other two wires going to pins 2 and 3, and with the blue DB-25 at each end containing pins 2, 3, and 7 only.

Using this cable with the null modem in the link, the customer used `stty -echo > /dev/ttyb2` and `stty -echo > /dev/ttya` . The system and CPU LEDs froze within a couple of seconds after the second command. This was followed by the machine completely resetting itself, as if an 'L1-A + K2' sequence had been issued.

Though a two-wire cable should not be used, the severity of the problem suggested that there was some other difficulty involved.

The SunOS, Sun386*i* Machines, and High-Speed Serial I/O

SunOS releases 4.0 and 4.0.1 do not contain certain features and bug fixes found in SunOS 4.0.3 for the Sun-2,3,4 and SPARCstation 1 machines. Some of the fixes found in SunOS 4.0.3 solve problems with high-speed SIO.

Sun386*i* SunOS 4.0 and Sun386*i* SunOS 4.0.1 likewise do not contain these bug fixes found in SunOS 4.0.3.

---

[5] This article is submitted by Guruprem Singh ("GS") Khalsa, Software Support Engineer, UNIX Systems Support Group, U. S. Answer Center.

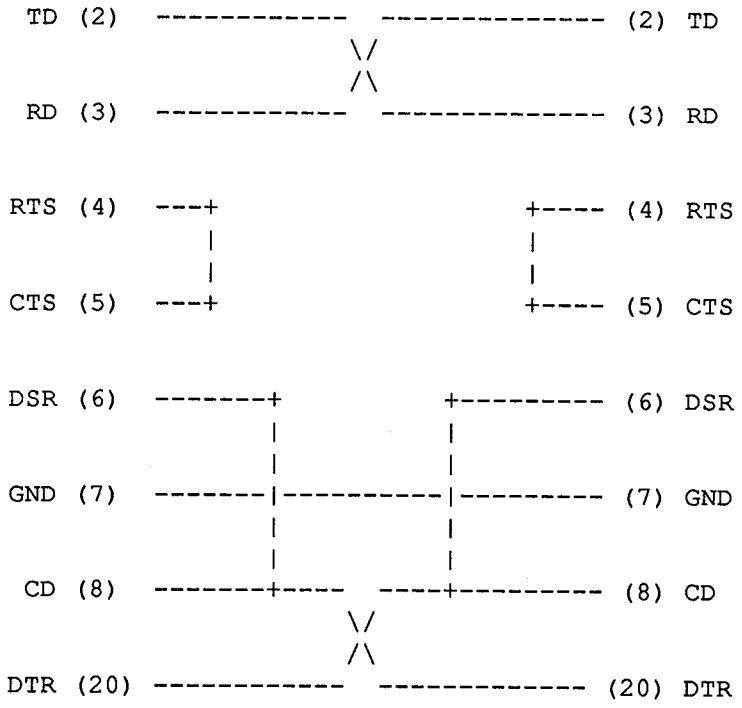
Over 25 bugs in the `termio` subsystem and the `STREAMS` layer beneath it have been found and fixed since the initial SunOS 4.0 release. Any one of several software flow-control bugs can cause a system to crash if an attached serial device tries to stop the Sun machine from sending too much serial information too fast.

SIO will work properly when the `serial_port` patches will be included in a future Sun386*i* SunOS release. The only short term workaround is to use a baud rate low enough that no software handshake (`CNTRL-S` and `CNTRL-Q`) is attempted. In most practical situations, this means limiting serial I/O to 1200 baud or less.

Customers can expect reliable performance at 9600 baud on a Sun386*i* running a future release of the Sun386*i* SunOS, after the `serial_port` patches have been included and software flow-control is working correctly.

Recommended Asynchronous Null-Modem Cabling Diagram

The recommended asynchronous null-modem cable in such situations would be wired as shown below.



The above wiring generally eliminates concerns about ground loops between the computer and the peripheral. To do this use only Signal Ground, pin 7, and *not* Chassis Ground, pin 1.

The SunOS and Real Time Events

Please note that software flow control *cannot* guarantee perfectly reliable handshaking between machines running a non-real time operating system. It remains possible that, under extreme load conditions, the OS may not be able to detect and respond to a `CNTRL-S ( XOFF )` character quickly enough to stop output data transmission to a peripheral device before that device's input buffer is overrun. Data will be lost in this case.

The only way to ensure reliable software flow control in systems running non-real time operating systems such as the SunOS is to limit the baud rate to 'what works' under all expected load conditions.

For Further Information

Please see the article entitled 'SunOS and Real Time' appearing on page 49 of the January 1989 STB. This article details differences between single- and multi-threaded kernels and how they relate to real time processing.

Monitor Sizes

Changing Monitor Base Sizes

Due to Sun's new product design, the current large-base monitors are being phased out and are being replaced by reduced-based versions. Over the past 45 days Sun has been shipping both versions. Please note, this does not affect the monitor performance. New part numbers for all of the new smaller-based versions as shown in the list below.

The following is the matrix of monitors from large-base units currently used to the corresponding reduced-base replacement monitor.

FROM (Large-Base)		TO (Reduced-Base)
365-1000-01	115V Hitachi 19" Color	365-1056-01
365-1006-01	240V Hitachi 19" Color	365-1054-01
365-1013-01	115V Philips OCLI	365-1044-01
365-1011-01	240V Philips OCLI	365-1045-01
365-1016-01	115V Elston/Philips ECL	365-1051-01
365-1014-01	240V Elston/Philips ECL	365-1043-01
365-1005-01	115V Displaytek Hi Res	365-1047-01
365-1007-01	240V Displaytek Hi Res	365-1050-01
365-1010-01	115V Philips Greyscale	365-1053-01
365-1009-01	240V Philips Greyscale	365-1046-01

Sun-3/80 Storage Systems

System Upgrades: the Mass Storage Subsystem

An officially-supported alternative for customers wishing to upgrade systems while retaining existing peripherals and data is now available.

The Mass Storage Subsystem (Shoebox) is now an officially supported option on Sun-3/80, SPARCstation 1, and SPARCstation 330 systems. Customers currently using the ESDI Shoebox will be able to use these same devices with the new systems. This alternative may save the expense of purchasing new storage devices, as well as the time and effort to transfer existing data onto new devices.

Anticipated Questions

The following questions may answer customer questions regarding this upgrade path.

1. Can we daisy chain the existing ESDI shoebox with the new embedded SCSI External Storage Module or Desktop Storage Pack for increased storage capacity?

   No. A single system can have *either* the ESDI shoeboxes *or* the new embedded SCSI External Storage Module/Desktop Storage Pack but no 'mixing' of the two.

2. Does this mean that if my system has internal embedded SCSI disks, I cannot connect an ESDI shoebox to it?

   That is correct. All internal drives shipped with Sun-3/80, SPARCstation 1, or SPARCstation 330 systems are embedded SCSI devices and therefore should not be mixed on the same bus with the ESDI shoebox. So, you should only plug an ESDI shoebox into a *diskless* system. Note that diskless SPARCstation 330 configurations do not yet exist on the price list, but all SPARCstation 330 upgrades are diskless.

3. Is there any additional hardware required to connect an ESDI shoebox to a new Sun-3/80 or SPARCstation system?

   Yes. The new systems use the smaller SCSI connector known as a 50-pin micro-miniature connector, while the ESDI shoebox uses the larger 50-pin sub-D connector. In order to connect the two, a cable with the proper connector on each end is necessary. This cable is available as part number X931. See the Accessories section of the new Price List for ordering information.

4. Where can I get more detailed information on the new embedded SCSI devices and how they compare to the devices currently available in the ESDI shoebox?

   See the *Embedded SCSI Drives Sales Guide* of April, 1989.

OPEN LOOK Source

OPEN LOOK Source Availability, Ordering, and Licensing

Source for AT&T's OPEN LOOK™ End User System and X Toolkit is orderable and shipping from AT&T now. OPEN LOOK software consists of the following:

- The OPEN LOOK XT+ Toolkit
- OPEN LOOK Window Manager
- OPEN LOOK Workspace Manager
- OPEN LOOK File Manager
- OPEN LOOK Terminal Emulator (based on `xterm`)

OPEN LOOK Source, Binary, and Runtime Pricing

The OPEN LOOK source license is $1000. The binary royalty and runtime fees are as shown below.

Units shipped	Binary Royalty (per copy)	Runtime Fee
1 - 5,000	$ 36	$ 8
5,001 - 10,000	29	7
10,001 - 20,000	19	6
20,001 - 50,000	17	5
50,001 +	16	4

Note that the binary royalty applies to OEM source licensees who port OPEN LOOK to their platforms and then furnish binaries derived from that port to their customers for general use. The runtime fee applies instead of the binary royalty to independent software vendors who ship OPEN LOOK code for use with a single application.

As a special introductory offer from AT&T, application developers who introduce an OPEN LOOK trademarked product by February 1990 will have all runtime payments waived until July, 1990.

The OPEN LOOK license is a site license which allows unlimited copies for systems connected via LAN up to a mile. You do not have to be a UNIX licensee to license OPEN LOOK. The porting base is the AT&T Intel 80386-based 6386 WorkGroup System.

Available separately (or packaged with OPEN LOOK at a different package price) is a source license for XWIN, AT&T's version of the MIT X Window System™. Redistribution licenses for OPEN LOOK and XWIN are available, too. Call 1-800-828-UNIX to get more details or to order the source products.

Binary Availability

A binary version of OPEN LOOK for the AT&T 6386 WGS workstation or compatible 386-based systems such as the COMPAQ 386 will be generally available on June 5 and orderable from your AT&T representative. The binary version will include the OPEN LOOK End User System, OPEN LOOK X Toolkit and XWIN.

60-Day Evaluation Program

Effective immediately, AT&T is offering a special OPEN LOOK binary 60-day evaluation program to qualified external Value Added Resellers (VARs) and independent software vendors. To participate, you must just execute a two-page agreement and submit a conditional purchase order for the production version of the OPEN LOOK End User System and X Toolkit ($995 list). There is no charge if you return the evaluation materials within 30 days after the end of the evaluation period. Otherwise AT&T will also ship you the regular production version and will invoice for it. Call Bill Stanger at 201-898-6076 (or via email *attunix!attmail!ntower!bstanger*) for the details.

OPEN LOOK is a trademark of AT&T. X Window System is a trademark of MIT

SunOS Release Tapes

SunOS Release Tapes and Ordering Information

This article contains information for those ordering SunOS release tapes. Following is a list of part numbers for the FCS product.

SPARCstation 1 U.S. Price List Part Numbers

These prices are applicable for SunOS binaries. Note that a SPARCstation 1 Owner's Set is included with every country kit.

OMAR	Description	Price
SS2-07	1/4" Tape No Manuals	$250
SS2-08	1/2" Tape No Manuals	$250
SS2-13	Diskettes No Manuals	$450
SS2-E-05	1/4" Encryption Kit	$50
SS2-E-06	1/2" Encryption Kit	$50
SS2-E-11	Diskettes Encryption	$50
SS-9B	SPARCstation 1 Owners' Set	$100

SPARCstation 1 SunOS Source Part Numbers

These part numbers are applicable for SunOS source for university, nonprofit, commercial, and federal users.

University Accounts

OMAR	Description
UN-SUNSRC01-403-4c	1/4" SUNOS Univ
UN-SUNSRC02-403-4c	1/2" SUNOS Univ

Nonprofit Accounts

OMAR	Description
NP-SUNSRC01-403-4c	1/4" SUNOS NONP
NP-SUNSRC02-403-4c	1/2" SUNOS NONP

Commercial and Federal Accounts

OMAR	Description
INTSRC01-403-4c	1/4" INTERNAL OS SRC
INTSRC02-403-4c	1/2" INTERNAL OS SRC
SUNSRC01-403-4c	1/4" SUNOS SRC
SUNSRC01-403-4c	1/4" SUNOS SRC
WINSRC01-403-4c	1/4" WINDOW SRC
WINSRC02-403-4c	1/2" WINDOW SRC

Sun386*i*s and PC-LANs

Sun386*i* and PC LANs

Sun386*i*™ workstations now connect to PC Local Area Networks (LANs). By supporting Novell NetWare and 3Com 3+ LANs, the Sun386*i* can be linked to more than 60 percent of existing PC networks.

The Sun386*i* is the first UNIX® workstation to provide full networking compatibility with existing PC LANs. Connection to these networks extends the Sun386*i*'s ability to link with a wide variety of computers through the Network File System (NFS™). NFS is the industry standard for heterogeneous transparent file sharing.

PC LAN connectivity enables users to preserve their investments in existing PCs and networking hardware and software, while taking advantage of a UNIX workstation. With the new PC LAN support, the Sun386*i* can also access PC files, share peripherals, and use other network resources such as email.

Support of Novell NetWare 2.1 running over Ethernet or Token Ring networks, as well as 3Com 3+ networks running over Ethernet is to be included in Sun386*i* SunOS™ 4.0.2. Users can easily install a standard LAN board in one of the Sun386*i*'s AT bus slots. PC LAN resources are accessed through a DOS window on the Sun386*i* machine.

Workstation or PC vendors of DOS-under-UNIX systems cannot provide PC LAN compatibility because either they lack an AT bus or they can not access the AT bus from DOS due to insufficient DOS integration.

Faster Speed

The workstation's new level of connectivity is due to an overall 20-fold improvement in AT bus interrupt response, allowing effective communication with PC LANs. Besides LAN support, this has increased the Sun386*i*'s interactive DOS performance, thereby improving DOS keyboard, mouse, and screen response. Calculation rates for DOS remain at native 80386 speed.

Lower Cost

In addition to offering faster DOS response and PC LAN connectivity, Sun has cut prices 10 to 15 percent on complete workstation configurations, which now begin at $8,990, including a high-resolution monitor and disks. Sun expects these enhancements to increase penetration of the Sun386*i* into non-technical accounts, such as financial services. During the past six months, the Sun386*i* workstations have been taking sales in these areas from high-end PCs.

The lower prices on the Sun386*i* are competitive with 80386-class PCs. Unlike most PCs, the Sun workstation comes standard with Ethernet, SCSI controllers and an 80387 floating point chip. A 20-MHz Sun386*i*/150 with 4 Mbytes of memory, 15-inch 1024x768 monochrome monitor, and 91-Mbyte disk drive costs $8,990, down 14 percent from its previous price. A 25-MHz Sun386*i*/250, with 8 megabytes of memory, 16-inch 1152x900 color monitor and 155-Mbyte disk

drive is $18,490, a reduction of 12 percent from its original price. Standard software includes the Sun386*i* SunOS UNIX operating system, SunView™ windowing system, DOS Windows, and ONC/NFS™ networking software.

8

CUMULATIVE INDEX: 1989

8

CUMULATIVE INDEX: 1989

Index

D

E

F

L

M

N

Q

R

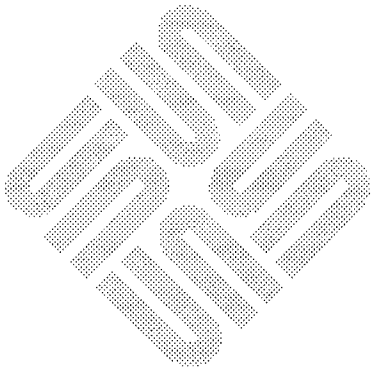
S

T

Z

Revision History

Revision	*Date*	*Comments*
FINAL	August 1989	Eighth issue of the 1989 Software Technical Bulletin, developed by Technical Information Services (TIS), WWFO Technical Support Services (TSS), Sun Microsystems, Inc.



Systems for Open Computing™

Corporate Headquarters
Sun Microsystems, Inc.
2550 Garcia Avenue
Mountain View, CA 94043
415 960-1300
TLX 37-29639

For U.S. Sales Office locations, call:
800 821-4643
In CA: 800 821-4642

European Headquarters
Sun Microsystems Europe, Inc.
Bagshot Manor, Green Lane
Bagshot, Surrey GU19 5NL
England
0276 51440
TLX 859017

Australia: (02) 413 2666
Canada: 416 477-6745
France: (1) 40 94 80 00
Germany: (089) 95094-0
Hong Kong: 852 5-8651688
Italy: (39) 6056337
Japan: (03) 221-7021
Korea: 2-7802255
Nordic Countries: + 46 (0)8 7647810
PRC: 1-8315568
Singapore: 224 3388
Spain: (1) 2532003
Switzerland: (1) 8289555
The Netherlands: 3133501234
Taiwan: 2-7213257
UK: 0276 62111

Europe, Middle East, and Africa, call European Headquarters:
0276 51440

Elsewhere in the world, call Corporate Headquarters:
415 960-1300
Intercontinental Sales