

Part Number 812-8601-01
Issue 1986 - 1
14 April 1986

Technical Support Bulletin
April 1986

Software Technical Support

Technical Support Bulletins are distributed to customers with software/hardware or software only support contracts. Send comments or corrections to "Technical Support Bulletin" at Sun Microsystems, Inc., 2550 Garcia Ave., M/S 2-30, Mountain View, CA 94043 or by uucp mail to sun!tsfiler. Customers who have technical questions about topics in the Bulletin should call Sun Technical Support.

Sun Microsystems continually tries to improve the quality of its products and service. If you have any comments, concerns or questions regarding Sun's products or support services, please write or e-mail to the above addresses. Your ideas will be forwarded to the appropriate organization at Sun.

UNIX is a trademark of AT&T Bell Laboratories.

Sun-3, Sun-2, DVMA, SunWindows, NFS, SunView, SunGKS, SunCGI, SunGuide, SunLink, Sun Microsystems, and the Sun logo are trademarks of Sun Microsystems, Inc.

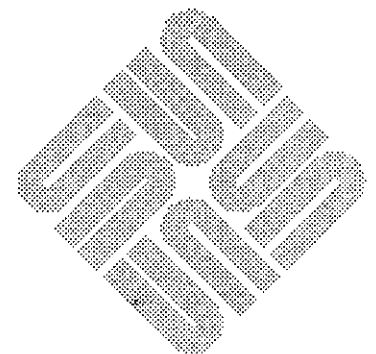
Sun Workstation and SunCore are registered trademarks of Sun Microsystems, Inc.

Copyright © 1986 by Sun Microsystems.

This publication is protected by Federal Copyright Law, with all rights reserved. No part of this publication may be reproduced, stored in a retrieval system, translated, transcribed, or transmitted, in any form, or by any means manual, electric, electronic, electro-magnetic, mechanical, chemical, optical, or otherwise, without prior explicit written permission from Sun Microsystems.

Contents

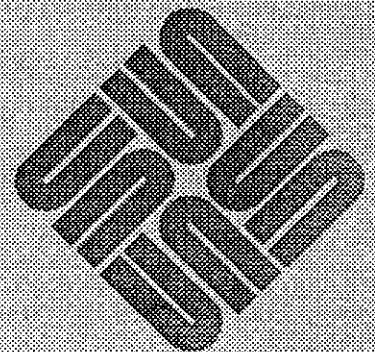
Section 1 NOTES & COMMENTS	3
1.1. Introduction	3
1.2. 3.0	4
Section 2 ARTICLES	7
SYSTEM ADMINISTRATION	7
2.1. Running <code>ifconfig</code> on Machines not Connected to the Ethernet	7
2.2. Filesystem Reorganization Under 3.0	8
2.3. How setup Accommodates 3Com Ethernet Controllers	10
2.4. Notes on 3.0 setup	11
2.5. Error in 3.0 <code>sendmail.cf</code> for Main Mail Hosts	13
2.6. NFS mount Intricacies	14
Section 3 IN DEPTH	19
3.1. How Diskless Sun-3 Machines Boot	19
3.2. 3.0 Boot and Standalone-Copy Problems with SMD Disks	26





NOTES & COMMENTS

NOTES & COMMENTS	3
1.1. Introduction	3
The spectrum of customer communication	3
Where the Technical Support Bulletin fits in	3
Layout of the Bulletin	4
Themes	4
1.2. 3.0	4





NOTES & COMMENTS

1.1. Introduction

Welcome to a new Technical Support Bulletin. Technical Support Bulletins are a service that helps Software Technical Support communicate with our customers on software support contracts.

Our aim is to provide useful timely information to aid you in using your Suns effectively and with minimal difficulty.

The spectrum of customer communication

This Bulletin fits into a continuum of resources that help fulfill that goal:

- Sun documentation is the first line of defense in resolving problems. Because of (and despite) its size most questions are answered somewhere in the documentation.
- The *Read This First* document that goes out with each software release details last-minute problems that did not make it into the manuals. They also summarize how customers may obtain software support from Sun.
- The Sun Users Group publishes a newsletter called **README** for its members that contains articles on Sun applications, how other users use their Suns, technical tips and news about the Sun Users Group.
- Local Users Group meetings provide a forum for the exchange of ideas and solutions between customers.

(For information about the Sun Users Group, call (415)960-7490 or send e-mail to sun!users.)

If the resources listed above do not provide the information you require then feel free to contact the Software Technical Support hotline, (415) 960-3500, with any question or problem you may have.

Where the Technical Support Bulletin fits in

The Technical Support Bulletin will be more timely than the documentation, but it cannot be up-to-the-minute in its coverage of problems. Some of the items in the Bulletin will be workarounds for problems that will be taken care of in the next release; others will have enduring interest. So, one of our priorities has been to index the Bulletins so old information remains accessible.

Layout of the Bulletin

Each Bulletin will have an introductory *Notes and Comments* section with general information in it. Then in the *Articles* section there will be items on particular bugs, tips or techniques of interest. In the *In Depth* section there will be one or two longer articles on a particular subject. The Bulletins will be indexed, and with each new issue the index will be revised to make it cumulative throughout the year.

We may be adding sections to the Bulletin in the future. For example, at some future date we plan to begin publishing a bug summary as part of, or as a supplement to, the Technical Support Bulletin.

Themes

In assembling the material for this Bulletin we noticed that many of articles deal with release 3.0, so that is the theme of this Bulletin. In the future we may have other themes, but we will be striving to get critical information into your hands as quickly as possible no matter what the subject is.

We welcome your comments.

Ray Jang sun!rjang (415)960-2702 M/S 2-30

the editor.

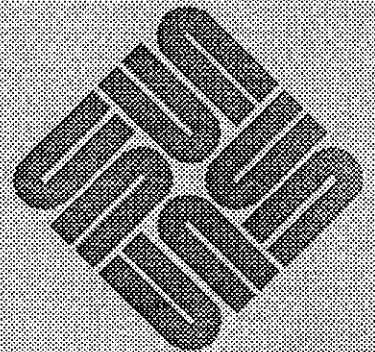
1.2. 3.0

If you have Sun-3 machines you will have received tapes of Sun Operating System Release 3.0 (68020) for the Sun-3 architecture. If you have Sun-2's you should soon receive the tapes of Release 3.0 (68010) for the Sun-2 architecture as part of your 2.0 to 3.0 upgrade. In this issue are articles covering the implications of upgrading to 3.0, some of the problems encountered in the release and some of the new features.

In a sense then, these articles are supplemental to the *Release 3.0 Change Notes* in the 3.0 documentation set and *Release 2.0 to 3.0 Upgrade Notes* that those customers getting the upgrade will receive. If you have not upgraded to 3.0 yet, it is vital that you read all three before upgrading.

ARTICLES

ARTICLES	7
SYSTEM ADMINISTRATION	7
2.1. Running ifconfig on Machines not Connected to the Ethernet	7
2.2. Filesystem Reorganization Under 3.0	8
Changes in the /etc directory	8
Changes in disk partitioning	9
Other changes	10
2.3. How setup Accommodates 3Com Ethernet Controllers	10
2.4. Notes on 3.0 setup	11
Redoing setup	11
Examining the setup.log file	12
The /etc/setup.info file	12
Repartitioning in diag vs. setup	12
Upgrade between major Sun UNIX releases	13
2.5. Error in 3.0 sendmail.cf for Main Mail Hosts	13
2.6. NFS mount Intricacies	14
All NFS mounts	14
Additional options in 2.3 and 3.0	14





ARTICLES

SYSTEM ADMINISTRATION**2.1. Running ifconfig on Machines not Connected to the Ethernet**

Applies to: Workstations with the ie or le Ethernet interface under release 3.0.

If you have a Sun that is not connected to the Ethernet, you may see the following messages periodically appear on the console:

either

ie0: no carrier

and

ie0: Ethernet jammed

or

le0: No carrier - transceiver cable problem?

and

le0: Transmit retried more than 16 times - net jammed

All Sun-3 workstations (except the Sun-3/50 and Sun-3/52) and Sun-2 VME models include an integral ie (4S) Ethernet interface on the CPU board, and most Sun workstations can have a separate Sun-2 Ethernet controller board attached that is driven by the same ie driver. The Sun-3/50 and Sun-3/52 have a built-in LANCE Ethernet le (4S) Ethernet interface that incorporates a thin Ethernet transceiver.

By default in release 3.0 the ifconfig(8C) command

```
/etc/ifconfig interface $hostname -trailers up
```

is run from /etc/rc.boot for all three Ethernet interfaces (ec0, ie0, le0). The commands for the interfaces that are not present fail but the interface that is present will be configured up and running at boot time. However if you run an ie or le equipped workstation standalone in a non-Ethernet environment or it is temporarily disconnected from the Ethernet, the interface driver detects the absence of the Ethernet carrier signal when it tries to transmit,

interprets it as a network problem, and displays these messages on the console.

NOTE *If you get these messages while connected to the Ethernet, consult the `ie(4S)` man page which explains the error messages in greater detail.*

To make the messages go away after the machine has booted, enter the following command as root:

```
polar# ifconfig ie0 down
```

or

```
polar# ifconfig le0 down
```

If the machine is not going to be connected to the Ethernet for some time, you can either remove the Sun-2 Ethernet board (if present), or change the line

```
/etc/ifconfig interface $hostname -trailers up
```

for your particular interface (`ie0` or `le0`) in `/etc/rc.boot` to

```
/etc/ifconfig interface down
```

The same procedure applies if your machine has a second `ie` interface (`ie1`). Refer to the *Configuring the Network in System Software* section in the *Communications* chapter in *System Administration for the Sun Workstation* for information on how to configure such gateway machines when they are connected to a network.

See Also:

`ifconfig(8)`, `ie(4S)`

Communications in *System Administration for the Sun Workstation*

2.2. Filesystem Reorganization Under 3.0

Applies to: Release 3.0.

The *Release 3.0 Change Notes* describes the major directory reorganizations in section 1.1., *System Software Changes and Upgrades*, but here are a few other changes you might be interested in.

Changes in the `/etc` directory

- `ypserv(8)` is now `/usr/etc/ypserv`.
- `/etc/ypbind` gets named `/etc/ypbind.orig` if you tell `setup(8S)` not to use the Yellow Pages.
- `setup` has been moved to `/usr/etc/setup`, but has a symbolic link from `/etc/setup` to maintain the old organization. This is the `setup` shell script that drives the actual `setup` programs in the `/usr/etc/setup.files` directory. The shell scripts `setup` uses are changed and have also moved to `/usr/etc/setup.files`. (Direct use of the binaries and scripts in `/usr/etc/setup.files` is not documented nor supported.)

- The Yellow Pages directory has also moved to `/usr/etc/yp`. It similarly has a symbolic link from `/etc/yp` to maintain the old appearance.

This reorganization has caused trouble for some customers and deserves a word of caution. The `/usr/etc/yp/domainname` directory now resides on the `/usr` filesystem, and if you work in an NFS environment, other machines will mount it. So far, everything sounds fine. But, `/etc/rc.local` assumes a system is a YP server if both the `/usr/etc/yp/domainname` directory and `/usr/etc/ypserv` exist, and starts `ypserv`. Since both are mounted from the same filesystem, `ypserv` gets started inappropriately.

Another problem arises when a diskfull NFS client is intended to be a YP slave server and mounts `/usr` from the YP master server in the same domain. Since both the YP master server and slave server both share the same `/usr/etc/yp/domainname` directory, the master and slave will try to update the same map files and possibly corrupt them. The same problem arises on diskless nodes mounting their `/usr` from a YP server, but you should never intentionally use a diskless client as a YP server.

You can prevent these problems by commenting out the group of lines in `/etc/rc.local` dealing with starting `ypserv`. We will publish additional information when it becomes available.

- A few items have been moved from `/etc/rc.local` to `/etc/rc.boot`. These include the `ifconfig` (8C) network configuration commands:

```
/etc/ifconfig ec0 $hostname -trailers up
/etc/ifconfig ie0 $hostname -trailers up
/etc/ifconfig le0 $hostname -trailers up
```

Note the addition of a third line for the `le` (4S) Ethernet interface. This interface is for the AMD LANCE (Local Area Network Controller for Ethernet) chip found on the Sun-3/50 workstation.

- Setting the `hostname` using `/bin/hostname` (1) has also been moved to `/etc/rc.boot` from `/etc/rc.local`.

Changes in disk partitioning

A 3.0 server may serve both Sun-2 and Sun-3 architectures, hence the assignment of partitions has changed substantially. It is documented in the *Setup Overview* chapter in *Installing UNIX on the Sun Workstation*. To serve both architectures, the server must have two `pub` partitions. Another change is that each `pub` is contained in its own separate hard partition, in contrast to the way a single hard partition was subdivided into `pub`, `client-root`, and `client-swap` ND partitions in previous releases.

Examining the first entries in `/etc/nd.local` on a heterogeneous server, you will find two lines for `pub`, for example:

```
user 0 1 /dev/xy0f 0 14720 -1
user 0 0 /dev/xy0e 0 14720 -1
```

`/dev/xy0e` is mounted on the server as `/pub.MC68010` and `/dev/xy0f` as `/pub.MC68020`. Each ND client mounts the appropriate `pub` as its own `/pub`.

- Another change is that ND client partitions are no longer in their own exclusive partition. This has changed because of a shortage of hard disk partitions. ND client partitions are now subpartitions of the `c` hard partition, the partition spanning the entire disk. This is best explained by examining an example `nd.local` file. These two `user` lines define the first ND client's root and swap partitions:

```
user ariel 0 /dev/xy0c 23000 11040 0
user ariel 1 /dev/xy0c 34040 23000 -1
```

Note that these ND partitions are subpartitions of `/dev/xy0c`, and the first ND partition begins with block 23000 in partition `c`. Partition `c` continues to span the whole disk as it did on previous releases, but ND only uses disk space not contained in other hard partitions for its client partitions. The offset starts the first ND partition past the server's root and swap spaces.

CAUTION Because of this change, even more extreme care must be taken when hand crafting ND client partitions because they are not restricted to a particular hard partition.

Other changes

- Standalone diagnostics that were previously in the `/stand` directory are no longer part of the release.
- The code generator for `pc(1)` is now `/usr/lib/f1`. It was previously `/lib/f1`, which was also used by `f77(1)`.
- The FORTRAN 77 compiler `f77` had been reorganized. The FORTRAN driver, `/usr/bin/f77`, calls the FORTRAN front-end, `/usr/lib/f77pass1`, followed by the new global optimizer and code generator, `/usr/lib/iropt` and `/usr/lib/cg`, respectively, and then `/lib/c2`, the peephole optimizer. If optimization is not invoked, `f77pass1` calls `cg` as a subroutine.
- `/usr/etc/extract_release` still exists, but is now obsolete. This script was used in previous releases for loading optional software. This script is no longer necessary because `setup` does this for you, and in the case of a heterogeneous server, installs the optional software for both architectures. The `extract_release` script can still be used, but is designed for a single architecture system.
- Netnews software, which has been available as a standard part of Sun releases, has been deleted from the 3.0 release due to size constraints. Netnews is in the public domain, and if you need it you can get it from other sources, for example the unsupported software tape from the Sun Users Group (if you are a member).

See Also:

`f77(1)`, `pc(1)`, `le(4)`, `hier(7)`, `config(8)`, `ifconfig(8)`, `nd(8)`, `setup(8S)`, `ybind(8)`
Filesystem Rearrangement in Release 3.0 Change Notes
Installing UNIX on the Sun Workstation

2.3. How setup Accommodates 3Com Ethernet Controllers

Applies to: 3.0 diskless clients using 3Com Ethernet controllers.

As the *Release 2.0 to 3.0 Upgrade Notes* states, a Sun-3 server can serve a diskless Sun-2 that is using a 3Com Ethernet board. When running `setup(8S)` to configure a server, the *client form* asks if the client has this Ethernet interface, and if you reply "yes," `setup` does two things differently:

1. Modifies the 3Com client's `/etc/fstab(5)` by adding the `rsize=2048` option for each NFS mount on this client. This reduces the NFS read-buffer size from the default of 8192 bytes to a size managed by the 3Com board.

An example of a 3Com client's `fstab` file is:

```
/dev/nd0 / 4.2 rw 1 1
/dev/ndp0 /pub 4.2 ro 0 0
fridge:/usr.MC68010 /usr nfs ro,hard,rsize=2048 0 0
fridge:/usr/fridge /usr/fridge nfs rw,hard,rsize=2048 0 0
```

2. Modifies `/etc/nd.local` on the server by adding a "2" at the end of each 3Com clients' user line. This reduces the maximum number of packets between acknowledgements from the default of 6 to 2. The optional `maxpacks` field is not documented in the 3.0 `nd(8C)` manual page, however it was described in the 1.x and 2.x manual pages as part of the `ether` command. (The essential ether information is now contained in `/etc/ethers`. See `ethers(5)`.)

A partial listing of the server's `/etc/nd.local` file is below:

```
user sun3 0 /dev/xy0c 49600 16000 0
user sun3 1 /dev/xy0c 65600 33600 -1
user sun2 0 /dev/xy0c 99200 16000 1 2
user sun2 1 /dev/xy0c 115200 33600 -1 2
```

Note the extra "2" at the end of each user line for the Sun-2 3Com client, "sun2."

See Also:

`fstab(5)`, `mount(8)`, `nd(8C)`, `setup(8S)`

Release 2.0 to 3.0 Upgrade Notes

Client Form section in *How To Use The Terminal Interface in Installing UNIX on the Sun Workstation*

Client Form section in *How To Use The Bit-Mapped Interface in Installing UNIX on the Sun Workstation*

2.4. Notes on 3.0 setup

Applies to: Release 3.0

Redoing setup

If `setup(8S)` reports any kind of error message or exit status, then the safest procedure is to contact Tech Support for assistance. Many customers have ignored seemingly innocuous messages like

```
Command command-string failed with exit status nn
```

only to find out later (for example) that their file system is full or damaged.

If you need to rerun `setup`, you should not do so by rebooting mini UNIX from the swap partition. The file system is not `fsck'd`, the exit from `setup` does not sync the disk, and the previous run of `setup` will have altered the filesystem.

The only way to insure that `setup` is run from the virgin state is to:

1. boot the bootstrap program off tape
2. load the standalone copy program

3. copy in mini UNIX
4. boot mini UNIX
5. rerun setup

This is time consuming, but is safest.

Examining the setup.log file

However, if setup fails for some unknown reason or you are interested in knowing (most of) what setup did on your system, it may be worth looking at the log file `/tmp/setup.log` that setup creates in mini UNIX. Some console error messages appear in this log that do not appear in setup's message region. Also, the commands executed by the scripts that setup calls are in here.

To see this file after a successful UNIX installation, click on **Exit** to exit setup, exit from the stripped-down `suntools` that setup runs in, and look at this file. Then abort and boot normally.

If setup had problems, you may still be able to look at `/tmp/setup.log`. Attempt to reboot mini UNIX from the swap partition by typing

```
Boot: disk(0,0,1)vmunix -as
```

as you did the first time. The mini UNIX filesystem will probably be in a corrupted state and will panic: if so, try the command again and run `fsck`.

CAUTION Again we stress that you should use this procedure only to investigate the files setup leaves behind, not as a short-cut to redoing setup.

The /etc/setup.info file

Another potentially useful undocumented file that setup creates is `/etc/setup.info`. This file is where setup saves some information for use if and when you next run setup in the Upgrade between major Sun Unix releases mode (described later). The information saved is the CPU-type of each of the clients and the optional software categories that have been extracted for each CPU-type.

Repartitioning in diag vs. setup

Users who are used to partitioning the disk from `diag` are sometimes confused when setup proceeds to modify their disk label.

You repartition the disk from within setup subject to the following limitations:

- you can't change the size of partition `disk0a` (root)
- you can't reduce the size of partition `disk0b` (swap)

If you remember that setup is running out of partition b (the swap partition) at the time, this makes sense.

setup does go in and change the size of partitions according to how much space you allocate for clients, how much optional software you load, which partition is the *free space hog*, etc. The default case for an ND server is that the free space hog is partition d, which is where the users' home directories are mounted. This means that as you request larger sizes for other partitions, the home directories partition shrinks.

The algorithm that setup uses makes sense but may not immediately be clear. setup initially sets the size of `/usr` for the space taken up by the bare bones `usr` files. As optional software categories are requested, setup increases the size of the `/usr` partition to accommodate them. As you adjust the size of clients' root and swap, the amount of partition c that is taken up by ND partitions changes.

It's a good idea to jump back and forth between the *Disks* form and the *Optional Software* or *Clients* forms as you experiment with different configurations in the latter forms in order to understand how *setup* adjusts the disk partition sizes.

Thus, in general, it is unnecessary and a waste of time to partition from within *diag*, which is why the section on relabeling the disk in *diag* is now an appendix to *Installing UNIX on the Sun Workstation*.

NOTE 3.0 root space requirements should be pretty compatible with the default label because files have been moved out of the root partition. See the article on 3.0 directories in this issue for more information. As in previous releases, large edits, compiles, window selections and so on require space in /tmp in the root partition.

NOTE The default swap partition size is much larger in 3.0 than in previous releases, reflecting the needs of most users. More customers will want to grow swap rather than shrink it.

Upgrade between major Sun UNIX releases

If you use the *Upgrade between major Sun Unix releases* mode of *setup*, it preserves the existing partitioning, and gets information like the hostname and current filesystem mount points from the disk. However, it will not preserve the contents of your disks during the upgrade.

See Also:

`setup(8S)`

Setup Overview in *Installing UNIX on the Sun Workstation*

2.5. Error in 3.0 `sendmail.cf` for Main Mail Hosts

Applies to: Release 3.0.

In release 3.0, `sendmail` determines whether or not it can connect to a machine in the local domain by checking if the other machine is in the *hosts* YP map (or `/etc/hosts(5)` if not running the Yellow Pages), instead of relying on the `/etc/hosts.equiv(5)` or `/usr/lib/mailhosts` files, as was done previously in 1.x and 2.x. This new feature works fine on the subsidiary mail machines, but fails on the main machine. The symptom is the subsidiary machines are able to mail any other machine on the network, but mailing from the main mail machine to any of the subsidiary ones results in the following message appearing on your screen and as part of the returned-mail message:

```
user@host... Never heard of host ; maybe you mean host.arpa ?
```

The reason this happens is as follows. The configuration file in 3.0 has a new pattern for the LHS (Left-Hand Side) rules: `$$x`, where `x` is a macro signifying a Yellow Page database. The rule `$$y` is predefined for `hosts.byname` when using YP, or `/etc/hosts` otherwise.

Use of `$$y` in `/usr/lib/sendmail*.cf` replaces the superseded rule `$$=S`. But in `/usr/lib/sendmail.main.cf`, `$$=S` has only been replaced in one of four instances.

The fix is to carry `$$y` through: replace the remaining three `$$=S` with `$$y` in the `/usr/lib/sendmail.cf` file on your main mail machine. Then kill and restart the `sendmail(8)` daemon. For completeness you can also update `/usr/lib/sendmail.main.cf`, which is the file the main machine's `/usr/lib/sendmail.cf` is derived from. (Note: the former file is `-r--r--r--` when delivered.)

If you do in fact wish to restrict the mail hosts you send mail to directly, you can return to the 2.x style of host checking. Just set up your `sendmail.cf` file in the normal way, then find the line

known hosts in this domain are obtained from `gethostbyname()` call
and add the line

```
FS/usr/lib/mailhosts
```

Then put the names (and aliases) of all the hosts that you want to send mail to directly in `/usr/lib/mailhosts`.

See Also:

`sendmail(8)`

Communications in System Administration for the Sun Workstation

2.6. NFS mount Intricacies

Applies to: Release 2.x and 3.0.

Sun's NFS is stateless and guarantees that in the event of a server failure the client does not need to do any special recovery. The flip side of this is that processes may hang indefinitely attempting I/O to NFS-mounted filesystems. However if you mount NFS filesystems soft, there is a danger of files getting corrupted and system calls failing that normally do not, such as `read(2)` and `write(2)`. To avoid problems please follow these guidelines for remote mounting.

All NFS mounts

NFS filesystems that you want to be able to write (create files or change existing files) should be mounted read-write and hard. The `fstab` entry should look like:

```
NFS_server:/usr/NFS_server /usr/NFS_server nfs rw,hard 0 0
```

If you don't need to modify files on a filesystem, for example `/usr/man`, you should mount it read-only and soft. The soft mount will time out if the server goes down so you can unmount it and keep going, and mounting it read-only keeps you from trying to write files. This `fstab` entry should look like:

```
NFS_server:/usr/man /usr/man nfs ro,soft 0 0
```

Notice that both entries have two zeros at the end. These are the orders in which to run `dump` and `fsck` on the filesystem. Since remote filesystems are checked and dumped on the server, these should both be zero.

NOTE *These example `fstab` lines apply to all NFS mounts except when mixing Sun-2 machines using the 3Com Ethernet interface with Sun-3 machines. See *Heterogeneous Networks in the Release 2.0 to 3.0 Upgrade Notes* for instructions on this configuration.*

Additional options in 2.3 and 3.0

The `mount` command now has many more options for tailoring your remote mounts. Below is a list of the new options and what they do.

Options that have an effect at mount time:

- bg If the first attempt at mounting fails, try some more in the background and let mount continue. This is now a configurable parameter under 3.0, and was the default when specifying `soft` under 2.x.
- fg If the first attempt at mounting fails, try some more in the foreground and keeps trying until it succeeds. This is a separate option under 3.0, but lumped together with the `hard` option under 2.x.

retry=*n* The number of times, *n*, to retry the mount if it fails the first time. The default is 1 if fg, and 10 if bg. This option is only available with 3.0.

Options that affect each NFS operation:

rsize=*n* The number of bytes, *n*, in each read request. This size is normally set by the kernel, but slow machines reading from fast ones, ones with bad Ethernet boards, or slow links can cause problems with the default size. Only in 2.3 and 3.0.

wsize=*n* The number of bytes, *n*, in each write request. Default set by the kernel.

NOTE Although rsize and wsize were documented in the 2.2 release, they are not totally effective. Release 2.3 permits correct handling of these parameters when NFS mounting files between Sun-2 machines using 3Com Ethernet interfaces running 2.x and Sun-3 machines running 3.0.

soft Give up on a request if *n* retransmissions time out, where *n* is set with the retrans option below.

hard Keep trying a request until it gets through.

timeo=*n* The initial NFS timeout. If a reply is not received in *n* tenths of a second, the timeout is multiplied by two and a retransmission is sent. The default is 7.

retrans=*n* The number of retransmissions, *n*, to do before giving up on a request. The default is 4.

Feel free to experiment with fg, bg, and retry to get mount to work the way you want, however, timeo, retrans, rsize, and wsize should not be randomly adjusted.

See Also:

mount(8)

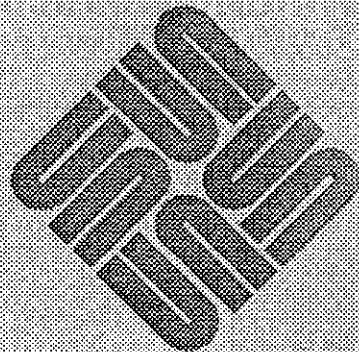
Heterogeneous Networks in Release 2.0 to 3.0 Update Notes

Sun Network Services in System Administration on the Sun Workstation



IN DEPTH

IN DEPTH	19
3.1. How Diskless Sun-3 Machines Boot	19
What you should see	19
As contrasted with a Sun-2 boot	19
So what happened in the Sun-3 boot?	19
In summary, what do you need to do?	20
OK, so what can go wrong?	21
Miscellaneous questions	23
3.2. 3.0 Boot and Standalone-Copy Problems with SMD Disks	26
Identifying and working around this problem	26
When loading the mini UNIX system	26
When booting the mini UNIX system	27
When booting the full UNIX system	28





3.1. How Diskless Sun-3 Machines Boot

Applies to: All Sun-3 machines with rev 1.1 or higher PROMs

What you should see

This example is a Sun-3/75; all Sun-3's will be similar.

```
>b
Boot: ie(0,0,0)
  [ pause, "-" character displayed, possibly get "|" character ]
Using IP Address 192.9.101.8 = C0096508
  [ short pause with "-" character ]
Booting from tftp server at 192.9.101.3 = C0096503
  [ get wild spinning character effect for a few seconds ]
Downloaded 28344 bytes from tftp server.

Boot: ie(0,0,0)vmunix
  [ pause, "-" character displayed, then get the normal "-" "=" flip-flop as vmunix comes in ]
Size: 391984+63300+70152 bytes
Sun UNIX 4.2 Release 3.0 (CLIENT020) #122: Wed Nov 6 11:15:04 PST 1985
  [ followed by the usual vmunix boot stuff, which falls out of the scope of this explanation ]
```

As contrasted with a Sun-2 boot

Booting a Sun-2 client under 3.0 looks the same as before, viz.:

```
>b
Boot: ec(0,0,0)vmunix
Load: ec(0,0,0)boot
Boot: ec(0,0,0)vmunix
Size: 410688+65164+63232 bytes
  [ Sun-2 client gets bootnd boot block from /pub, loads /pub/boot, then loads /pub/vmunix ]
```

So what happened in the Sun-3 boot?

- Client broadcasts "Hey, rarpd, my Ethernet address is 8:0:20:1:31:E6; anyone know my Internet address?".
- rarpd(8C) is a daemon that responds to Reverse Address Resolution Protocol requests. The RARP protocol lets machines discover their IP address given their Ethernet address.

- A machine running `rarpd(8C)` responds "It's 192.9.101.8", getting this Internet address from its `/etc/ethers(5)` and `/etc/hosts(5)` files, or the corresponding YP services if available.

- Client prints

Using IP Address 192.9.101.8 = C0096508

Actually, there's a wrinkle here. Machines running `rarpd` (typically only ND servers) check their `/tftpboot` directory to see if they have a tftp boot file for the machine broadcasting the RARP request, i.e. they see if they have the requestor as a client. If one does, it answers immediately. If it doesn't, it pipes its answer to a second copy of `rarpd`, which waits 3 seconds before replying. This reduces the network load due to all the ND servers on the net responding to RARP requests simultaneously, and means the client is more likely to get a response from its own server. Anyway, this is why there are two `rarpd` processes running on servers.

- Client broadcasts "Hey, tftpd, anyone got a file called C0096508?".

This is like an ordinary user's `tftp(1C)` request:

```
tftp> get C0096508
```

except the client broadcasts it instead of directing it to a particular machine.

Actually, there's another related wrinkle here. To save invoking the `tftpd(8C)` server, `/usr/etc/in.tftpd`, on all the machines on the local network, the client initially issues its TFTP request to the machine that answered its RARP request, assuming that that machine is probably its server. If that machine doesn't acknowledge or says "I don't have a C0096508", then the client broadcasts its TFTP request.

- Now, it just so happens that starting with 2.0 Sun has hacked `/usr/etc/in.tftpd`, so that if it gets an unqualified filename (no leading "/") it looks in `/tftpboot`. This is a new directory on 3.0 ND servers.

So, the server looks in its `/tftpboot` directory, and finds a file `/tftpboot/C0096508` which it feeds to the client.

- Client prints

Booting from tftp server at 192.9.101.3 = C0096503

In fact, `/tftpboot/C0096508` is a symbolic link to `ndboot.sun3.pub1`, which is a boot file 28344 bytes big (in this example).

(get wild spinning character effect for a few seconds)

Downloaded 28344 bytes from tftp server.

- By default `ndboot.sun3.pub1` boots `vmunix` from the 2nd pub partition on the server, which is usually `/pub.MC68020`, so the client now starts to boot `vmunix` just as a Sun-2 does:

```
Boot: ie(0,0,0)vmunix
```

```
Size: 391984+63300+70152 bytes
```

In summary, what do you need to do?

The diagnostics in the next section can be preempted by stating some simple rules. When a new ND client is added, in addition to setting up its ND partition and entry in `/etc/nd.local`, do the following on the server and the YP master, regardless of machine type:

1. Put its Ethernet address in `/etc/ethers`.
2. Put its IP address in `/etc/hosts`.
3. Create the appropriate link in the `/tftpboot` directory on the server depending on the client machine's type.
4. Rebuild the YP maps; make sure the server is using the updated maps.
5. Rerun ND on the server. You should not have to reboot the server.

Needless to say, `setup (8S)` does all this for you when installing a server and its diskless clients.

OK, so what can go wrong?

1.

```
>b ie()
Requesting Internet address for 8:0:20:1:31:E6
("!" character, infinite wait)
```

Usually the client just says `Using IP address...` but if there's a delay it prints out the above message.

The server (or some machine on the LOCAL network) has to explicitly run `/usr/etc/rarpd`. It doesn't matter which machine responds to the client's request to know its IP address. The 3.0 `/etc/rc.local` starts up `rarpd` only if the machine is an ND server, just after it starts up `nd`.

`rarpd` should show up as 2 processes in a `ps ax`. If there's no response, the client prints the `Requesting Internet address ...` message above and periodically rebroadcasts its RARP request. If `rarpd` isn't running on any machine on the local network, the client hangs.

If you start up `rarpd` on some machine on the net, the client should proceed with booting.

Or this could be because the client is not in `/etc/ethers` (network not running YP) or not in `ethers.by*` (if running YP), the client name returned from `ethers` is not in `hosts`, etc. Basically anything preventing the client from finding out its Internet address based on its Ethernet address.

If you make changes and the client still doesn't boot, try rebooting the server.

2.

```
Using IP Address 192.9.101.8 = C0096508
(pause, "\ " character, then)
tftp: time-out.
```

Perhaps `/usr/etc/in.tftpd` can't be invoked on the server because:

- `inetd` is not running
- `in.tftpd` has been renamed or is missing from `/etc/servers`.
- The entry for `tftp` is missing from `/etc/services`.

(You can check these three by trying `tftp servername`, since Sun-3 booting uses the ordinary TFTP protocol.)

- there is no `/tftpboot/C0096508` on any of the machines the client can talk to. If it has never worked, this is the most likely cause.

You can try forcing the client to make its TFTP request for C0096508 from a particular machine by typing **bie (0, n, 0)** where *n* is the hexadecimal Internet host number of its server.

3.

```
Using IP Address 192.9.101.8 = C0096508
Booting from tftp server at 192.9.101.3 = C0096503
Downloaded 28344 bytes from tftp server.
```

(machine hangs, gives wild panic message, etc.)

The client got something from /tftpboot, but when it ran it it didn't make any sense. The client has reported which machine it is booting from, so on that machine check to see what the client's IP address is a link to in /tftpboot:

```
server% ls -l /tftpboot
total 179
lrwxrwxrwx 1 root          16 Dec 20 12:02 C0096502 -> ndboot.sun2.pub0
lrwxrwxrwx 1 root          16 Dec 20 12:02 C0096505 -> ndboot.sun2.pub0
lrwxrwxrwx 1 root          16 Dec 21 10:30 C0096506 -> ndboot.sun3.pub1
lrwxrwxrwx 1 root          16 Dec 21 10:30 C0096508 -> ndboot.sun3.pub1
lrwxrwxrwx 1 root          16 Dec 20 12:02 C009650A -> ndboot.sun2.pub0
-rwxr-xr-x 1 root        29744 Dec 20 12:02 ndboot.sun2.private
-rwxr-xr-x 1 root        29744 Dec 20 12:02 ndboot.sun2.pub0
-rwxr-xr-x 1 root        29744 Dec 20 12:02 ndboot.sun2.pub1
-rwxr-xr-x 1 root        28344 Dec 20 12:02 ndboot.sun3.private
-rwxr-xr-x 1 root        28344 Dec 20 12:02 ndboot.sun3.pub0
-rwxr-xr-x 1 root        28344 Dec 20 12:02 ndboot.sun3.pub1
```

Obviously, each link must be to the appropriate machine architecture — if it is a link to `ndboot.sun2.*` then major problems will ensue.

It also matters which of the `ndboot.sun3.*`'s it is a link to. If it is a link to `ndboot.sun3.pub1` the client is booting from the 2nd pub partition on the server, on which `setup` mounts the `/pub.MC68020` partition by default. You can check `/etc/nd.local` and `/etc/fstab` on the server to make sure (this example is from a different server):

```

server% cat /etc/nd.local
#
# These lines added by the Sun Setup Program
#
clear
version 1
user 0 1 /dev/xy0f 0 32960 -1
user chani 0 /dev/xy0c 49600 20480 0
user chani 1 /dev/xy0c 70080 40960 -1
#user dummy 0 /dev/xy0c 111040 16000 1
#user dummy 1 /dev/xy0c 127040 33600 -1
son
#
# End of lines added by the Sun Setup Program
#
server% cat /etc/fstab
/dev/sd0a /usr.MC68020/fridge 4.2 rw 1 4
/dev/xy0a / 4.2 rw 1 1
/dev/xy0f /pub.MC68020 4.2 rw 1 2
/dev/xy0h /usr.MC68020 4.2 rw 1 3

```

The 0 1 in user 0 1 /dev/xy0f means that it is the 2nd public partition, and we see that /dev/xy0f is /pub.MC68020. So the client is loading the right ndboot code.

ndboot.sun3.pub0 is for booting from the 1st pub partition (user 0 0) but setup by default mounts /pub.MC68010 on that partition.

ndboot.sun3.private is for booting from the client's own ND partition rather than /pub.

4.

Requesting Ethernet address for 192.9.101.19

You should only get this message if you instructed the client to boot from a particular host, in this case by typing `b ie(0,13,0)`. Notice that the host number is in hexadecimal. The client needs to know the Ethernet address of that host so it can direct its tftpboot request to that particular machine. That machine is either down or does not speak the arp (4s) protocol.

Miscellaneous questions

What about the boot block?

The boot block is installed on the disk partition by `newfs(8)` or by `/usr/mdeci/installboot`. All it knows how to do is load /boot from the partition and device it is loaded on (e.g. /pub/boot for Sun-2 ND clients). The tftpboot procedure that Sun-3's use is able to load the boot program (e.g. ndboot.sun3.*) into memory in one step, so Sun-3's don't need a boot block to boot over the network (it is still needed to boot from disk).

What about /pub/boot?

/pub/boot is a general purpose boot program that knows how to boot from a variety of devices. Diskless Sun-3's boot a special ndboot program in /tftpboot that their Internet address is linked to, so they don't need /pub/boot to boot vmunix, but they can still boot it explicitly if you type

```
b -a
```

What are all these different ndboots in /tftpboot for?

They are boot programs like /pub/boot that can be directly loaded into memory in the tftpboot process and which know which program to load by default (vmunix) and which partition to load from. Here they are in detail:

```
ndboot.sun3.*      Sun-3 boot code.
ndboot.sun2.*      Sun-2 boot code, but currently irrelevant since Sun-2's don't use the new boot procedure.
ndboot.sun3.pub1   Sun-3 boot code to boot from 2nd pub partition (by default set up reserves
                   /dev/ndp0 for MC68010 clients).
ndboot.sun3.private Sun-3 boot code to boot from client's own ND partion, /dev/ndlx.
```

Specifically, what's ndboot.sun3.private for?

If you remake a client's link in /tftpboot so that it points to ndboot.sun3.private, then when the client boots it will boot a kernel in its own ND partition.

The usual restrictions apply — vmunix should not be a symbolic link, it needs to have ident GENERIC or options GENERIC or root on nd, etc. You don't need to install a boot block on the Sun3's ND partition.

So the old **bie(0,0,40)** hack to boot a vmunix in your own partition rather than /pub/vmunix is no longer required?

Yes. One of the intents of the new style of booting is to ensure that machines always boot the correct kernel when powered on. You can still go in and manually override the default boot device.

The boot code in the old ndboot and the new ndboot.sun?.* code has always XOR'd the partition number you give it with what it thinks your default partition is.

So, as an example: if you are booting from pub partition 1 and you want to boot from your own partition, type

```
bie(0,0,41)
```

This is XOR'd with your default partition (01) to get 40, which is still the hard-coded hack that points to your default partition.

Conversely, if you are booting from your private partition and you want to boot from pub partition 1, type

```
bie(0,0,41)
```

This is XOR'd with your default partition (40) to get 1, which is pub partition 1.

What's the significance of the new “\” and “/” characters during boot?

They don't mean anything in particular except “packet pumping in progress”. The other boot characters are documented under *Ethernet* in the *Communications* chapter in *System Administration on the Sun Workstation* manual, but it has not been updated for 3.0.

Why do it this way?

The main advantage of the Sun-3 boot procedure is that it is more flexible. With Sun-2 proms, the monitor patches in the string “vmunix” when no boot program is given, and ND clients always boot from /pub by default. With the Sun-3 it is up to a host-specific boot program to decide on the default program to boot and where to get it from.

The selection of this boot program itself is also more flexible. Each machine attempts to boot a file for its own Internet number in /tftpboot. These files are just UNIX files. By default they are symbolic links to the ndboot.sun?.{private, pub{0,1}} files, but they could be hard links, or standalone programs.

The new boot protocol is based on the standard protocols RARP and TFTP.

As mentioned earlier, when powered on or after a crash machines will boot from the correct partition.

See Also:

tftp(1C), ethers(5), hosts(5), nd(8C), rarpd(8C), setup(8S), tftpd(8C)

Sun Network Services in System Administration for the Sun Workstation

3.2. 3.0 Boot and Standalone-Copy Problems with SMD Disks

Applies to: Release 3.0 with xy0 disks.

Sites installing or running 3.0 on systems using Xylogics disk controllers with SMD disks may experience problems during the standalone copy step while installing 3.0, or when booting from disk.

These problems are caused by improper handling of mapped blocks by the standalone xy (4S) driver, characterized by two

xy: hard error bno *block-number*

messages, typically identifying the same *block-number*, but this is not always true. The UNIX xy driver handles mapped blocks correctly, so these problems are restricted to booting and while using the standalone copy during installation. Specifically, the problem only affects the following files on the 3.0 installation tape:

- Tape standalone copy - fifth file on the installation tape.
- /boot in mini UNIX and full UNIX.
- /stand/copy in mini UNIX and full UNIX.
- /usr/mdec/bootxy which gets installed as the *boot block* for the full UNIX root file system.

This article describes the circumstances in which these problems might arise, and the various workarounds. Some of the workarounds involve contacting Sun Technical Support for the appropriate *patch* tape. If this applies to your situation, Technical Support must know your machine architecture – Sun-2 (68010) or Sun-3 (68020) – and the type of tape media – ¼-inch cartridge or ½-inch reel – suitable for your machine.

Identifying and working around this problem

There are three stages during the installation process when the patches supplied on the patch tapes might be necessary.

- When loading the mini UNIX system.
- When booting the mini UNIX system.
- When booting the full UNIX system.

The last instance of this problem is not necessarily restricted to when initially booting the full UNIX system, but may occur whenever booting a new kernel from disk. A discussion of each individual situation follows below.

When loading the mini UNIX system

This instance of the problem surfaces while loading the mini UNIX system as instructed in section 2.2 The Mini UNIX System on page 20 of the *Installing UNIX* manual. Following is an example illustrating the error messages in context.

```

Boot: tape(0,0,4)
Size: 22528+4096+182836 bytes      [ numbers vary with system level ]
Standalone Copy
From: tape(0,0,5)
To: xy(0,0,1)
xy: hard error bno block-number
xy: hard error bno block-number
Write error
Copy completed - 460800 bytes      [ number varies ]
Boot:

```

Note there are two hard error messages. The *block-numbers* are either the same or very close. The Copy completed message indicates completion, although unsuccessful.

Sun-2 (68010)

If you have the 2.0 installation tape, you can use the standalone copy from it. Follow the instructions below, otherwise contact Sun Technical Support for the 3.0-68010 patch tape.

- Temporarily substitute the 2.0 tape for the 3.0 tape.
- Boot from the 2.0 tape:

```

>b tape()
Boot: tape(0,0,0)
Boot:

```

- Boot the standalone copy program from the 2.0 tape:

```

Boot: tape(0,0,1)
Size: 23328+5324+113452 bytes      [ numbers vary with system level ]
Standalone Copy
From:

```

- Before responding to the From: prompt, reload tape 1 of the 3.0 installation tapes, and continue by copying the mini UNIX system.

```

From: tape(0,0,5)
To: xy(0,0,1)

```

Now continue normal installation starting at the top of page 21 of the *Installing UNIX* manual.

Sun-3 (68020)

You must contact Sun Technical Support for the 3.0-68020 patch tape.

When booting the mini UNIX system

The following illustrates what might appear when attempting to boot the mini UNIX system while performing step 1 at the top of page 21 of the 3.0 *Installing UNIX* manual.

```

Boot: xy(0,0,1)vmunix -as
Size: 23328+xy: hard error bno block-number      [ numbers vary with system level ]
xy: hard error bno block-number
>

```

This indicates the `/vmunix` file in the mini UNIX file system is stored on disk in an area with a mapped block. The actual placement of the `xy: hard error bno` message varies depending on where the mapped block occurs within the `/vmunix` file.

Sun-2 (68010)

As with the previous scenario, you can use the 2.0 installation tape as a workaround if you have it. Otherwise, contact Sun Technical Support for the 3.0-68010 patch tape. If using the 2.0 installation-tape workaround, replace step 1 at the top of page 21 of the 3.0 *Installing UNIX* manual with the following.

- Load the 2.0 installation tape.

- Boot off the 2.0 tape:

```
>b tape ()
Boot: tape (0,0,0)
Boot:
```

- Boot mini UNIX from disk:

```
Boot: xy(0,0,1)vmunix -as
Size: 23328+5328+113452 bytes [ numbers vary with system level ]
Sun UNIX 4.2 Release 3.0 (GENERIC) #1: Tue Sept 11 20:35:13 PDT 1985
Copyright (c) 1985 by Sun Microsystems, Inc.
[ above messages vary with system level ]
[ ...various configuration messages... ]
root device?
```

- Reload tape 1 of the 3.0 installation tapes and continue installation starting with step 2 on page 21 of the 3.0 *Installing UNIX* manual.

Sun-3 (68020)

Contact Sun Technical Support for the 3.0-68020 patch tape.

When booting the full UNIX system

This instance of the problem may surface while booting full UNIX when initially booting the new system after completing Chapter 6: *Executing Setup*, or possibly when booting a newly built kernel. The following scenarios illustrate these situations.

```
>b
Boot: xy(0,0,0)vmunix
Load: xy(0,0,0)boot
xy: hard error bno block-number
xy: hard error bno block-number
>
```

Or


```

>b
Boot: xy(0,0,0)vmunix
Load: xy(0,0,0)boot
Boot: xy(0,0,0)vmunix
Size: 38796+xy: hard error bno block-number [ numbers vary with system level ]
xy: hard error bno block-number
>

```

The first example identifies the /boot file as containing a mapped block. If this problem occurs, it happens when initially booting a newly installed system.

The second scenario could happen whenever booting a new kernel, that is, when booting a new software installation, or possibly when booting a newly built kernel. Occurrence of this problem reflects the presence of a mapped block within the /vmunix file. Where the xy: hard error bno messages appear with respect to the other messages varies with the location of the mapped block.

Both Sun-2 (68010) and Sun-3 (68020)

The workaround technique is the same for both cases — /boot or /vmunix. It involves copying the affected file to an area on disk without mapped sectors. The following steps provide a workaround for both situations.

- Boot from the 3.0 distribution tape:

```

>b tape()
Boot: tape(0,0,0)
Boot:

```

- Boot the standalone copy from the 3.0 distribution tape and load mini UNIX:

```

Boot: tape(0,0,4)
Size: 22528+4096+182836 bytes [ numbers vary with system level ]
Standalone Copy
From: tape(0,0,5)
To: xy(0,0,1)
Copy completed 4718592 bytes [ numbers vary with system level ]
Boot:

```

- Boot mini UNIX:

```

Boot: xy(0,0,1)vmunix -as
Size: 23328+5328+113452 bytes [ numbers vary with system level ]
Sun UNIX 4.2 Release 3.0 (GENERIC) #1: Tue Sept 11 20:35:13 PDT 1985
Copyright (c) 1985 by Sun Microsystems, Inc.
[ above messages vary with system level ]
[ ...various configuration messages... ]
root device? xy0*
using 100 buffers containing 366592 bytes... [ numbers vary with system configuration ]
#

```

- Mount the full UNIX xy0a file system and change to that directory:

```

# /etc/mount /dev/xy0a /a
# cd /a
#

```

- Copy and rename the offending file (replace *file* with either `boot` or `vmunix`, as appropriate):

```
# mv file file.old
# cp file.old file
# rm file.old
#
```

- Now, unmount the `xy0a` partition:

```
# cd /
# /etc/umount /a
#
```

- Finally, abort mini UNIX and boot up full UNIX.
- Contact Sun Technical Support for the appropriate patch tape.

Although these last workarounds allow you to sufficiently avoid the problem, you may want to take measures for eliminating possible problems in the future by installing correctly working `/boot`, `/stand/copy`, and `/usr/mdec/bootxy` files from the patch tape.

See Also:

Installing UNIX on the Sun Workstation

Index

3

- 3.0
 - filesystem reorganization in 3.0, 8
 - hard errors on SMD disks, 26
 - ND partitions, 9
 - NFS mount options, 14
 - partition changes, 9
 - setup notes, 11
- 3Com Ethernet controller
 - compatibility with Sun-3 machines, 10

B

- booting
 - from a different partition, 24
 - hard errors on SMD disks in 3.0, 26
 - how diskless sun-3's boot, 19
 - troubleshooting Sun-3 diskless boot, 21

D

- diag
 - 3.0 partition use changes, 9
 - repartitioning in diag vs. setup, 12
- disk
 - 3.0 root requirements, 13
 - 3.0 swap size, 13

E

- Ethernet
 - 3Com controller on clients from Sun-3 servers, 10
 - systems without, 7
- ethers
 - used in diskless boot, 20

F

- filesystem
 - reorganization in 3.0, 8
- fstab
 - Sun-2 3Com clients with Sun-3 servers, 10

H

- hostname
 - filesystem reorganization in 3.0, 9

I

- ie
 - error messages, 7
- ifconfig
 - filesystem reorganization in 3.0, 9

- ifconfig, *continued*
 - on standalone machines, 7
- installing
 - 3.0 on SMD disks, 26

L

- le
 - error messages, 7
 - filesystem reorganization in 3.0, 9

M

- mail
 - main mail machine sendmail problem in 3.0, 13
- mount
 - options for NFS, 14

N

- ND
 - 3.0 partition changes, 9
 - boot files, 23 *thru* 24
 - how diskless sun-3's boot, 19
 - pub partition in 3.0, 22
- nd
 - Sun-2 3Com clients with Sun-3 servers, 11
- netnews
 - not in 3.0, 10
- NFS
 - mount options, 14

R

- RARP
 - used in diskless boot, 19
- rc files
 - ifconfig on standalone machines, 7

S

- sendmail
 - main mail machine problem in 3.0, 13
- setup
 - /etc/setup.info, 12
 - 3.0 root requirements, 13
 - 3.0 swap size, 13
 - disk partition changes, 9
 - filesystem reorganization in 3.0, 8
 - notes on 3.0 version, 11
 - recovering from errors, 11
 - repartitioning in diag vs. setup, 12
 - setup.log, 12

setup, *continued*
 upgrade path, 13
standalone copy
 hard errors on SMD disks in 3.0, 26
standalone diagnostics
 not in 3.0, 10
standalone machine
 ifconfig, 7
standards
 tftpboot protocol, 25

T

tftp
 changes to support booting, 20
 used in diskless boot, 20
/tftpboot
 use in Sun-3 booting, 20

Y

YP
 filesystem reorganization in 3.0, 9
ypbind
 filesystem reorganization in 3.0, 8
ypserv
 filesystem reorganization in 3.0, 8



