sun microsystems

Software Technical Bulletin
December 1989

Technical Information Services

Part Number 812-8912-01
Issue 1989–12
December 1989

Software Technical Bulletin
December 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 December 1989 issue include the items of interest listed below.

- In Depth Special Features: The Sun386*i* Administration Cookbook
- Hints and Tips: Sun386*i* YP Masters and Slaves
- The Hackers' Corner: `cleandisk`
- Configurations: updated software release level tables, effective October 27, 1989

In Depth Special Features: The Sun386*i* Administration Cookbook

This month continues a three-month series of In Depth features which will include the *Sun386i Administration Cookbook*. This second month includes the title page, trademarks, table of contents, and chapters 1-5.

The next STB issue, January 1990, will finish this three-part series and will include chapters 6-8, 10, appendix A (`automounter`), and the cookbook index.

STB readers should note that the pagination in these In Depth features is the same as in the cookbook. Simply remove the cookbook pages from the November 1989, December 1989, and January 1990 STBs and insert them in a separate Sun386*i* Administration Cookbook binder. The remaining STB pages are paginated cumulatively as reflected in the STB Cumulative Index.

Hints and Tips

This month's hints and tips section contains an item of interest to those setting up Sun386*i* YP master and slave servers. Following the hints contained in the procedure allows you to avoid `ypserver not responsing` messages and `fsck` inconsistencies.

The Hackers' Corner

This month's **Hackers' Corner** contains a short script named `cleandisk` of use to those wishing to routinely delete large, forgotten files that are of no further use. This script allows you a convenient way to automatically free disk space.

For those with email access and wishing an online copy of **Hackers' Corner** code samples, please email *sun!stb-editor* or *stb-editor@sun* with your request. Please include the program title, and the STB issue month and year with your request.

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

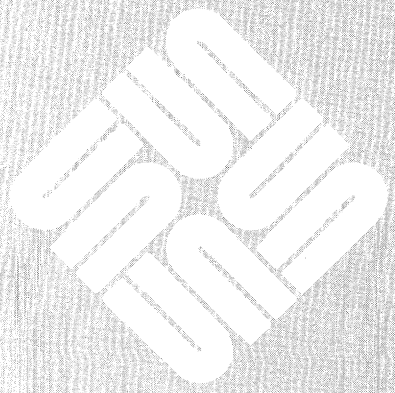
The seven tables showing current Sun software product release levels appear monthly. These tables show release levels for operating systems, communications products, unbundled languages, unbundled applications, unbundled graphics, other products, and TOPS networking products. The tables in this issue are updated through October 27, 1989.

Thanks.

The STB Editor

2

ARTICLES



2

ARTICLES

IEEE Floating Point

IEEE Floating Point and Sun FORTRAN

This article discusses the IEEE floating point standard from the Sun FORTRAN perspective and addresses the following commonly-asked questions:

What do the IEEE error messages mean?
Refer to "`ieee_retrospective` and IEEE Warning Messages," below.

If I get an exception, how do I find out where it occurred?
Refer to "`ieee_handler` and UNIX FPE Signals" and "`dbx` and IEEE," below.

How do I find out all the exceptions that have occurred at any time in my program? Refer to "`ieee_flags`" and "`ieee_retrospective` and IEEE Warning Messages," below.

I know that my program works correctly. How do I turn off the warning messages? The exceptions can be cleared using the information described in "`ieee_flags`" below, or by calling a dummy `ieee_retrospective`, as described in "`ieee_retrospective` and IEEE Warning Messages," below.

How can I get better performance with IEEE floating point?
Refer to "Underflow Exceptions and Performance," below.

Introduction to the IEEE 754 Standard

IEEE 754 standard floating point arithmetic offers the user greater control over computation than is possible in any other type of floating point. Sun is one of many vendors who support this standard. The IEEE standard gives the user control over functions such as rounding precision and rounding direction. In addition, the IEEE standard allows the user to decide for himself or herself whether the program should abort or continue.

The IEEE 754 standard is implemented by Sun by a combination of both hardware and software: on Sun-3 and Sun-3x (Motorola 680X0-based) kernel architectures; Sun-4 and Sun-4c (SPARC-based) kernel architectures; and Sun-386*i* (Intel 80386-based) kernel architecture. Currently, several hardware floating point options are provided by Sun, and all conform to the IEEE standard.

On a Sun-3 system, the possible combinations are as follows:

- Motorola 68881, with or without the Weitek 1165/5 chip set floating point accelerator (FPA, sold separately)
- Motorola 68882, with or without TI 8847 floating point accelerator (FPA, sold separately)

On Sun-4 systems, the following possible combinations are available:

- Older Sun-4 systems: Weitek 1164/5 (FPU)
- Newer Sun-4 systems and SPARCstations: Weitek 3170 or TI 8847 (FPU2)

The Sun386*i* system supports Intel 80387 and optional Weitek 3167 floating point hardware (FPX).

A Note on the Examples Used in this Article

All examples used in this article are for a Sun-4 system with a TI 8847 Floating Point processor (FPU2), running SunFORTRAN release 1.2 and SunOS release 4.0.3. The examples have also been run on the following systems/configurations:

- Sun-4/Weitek 1164/5 (FPU)
- Sun386*i* with standard 80387 (FPX)
- Sun-3/68881/Weitek 1164/5 (FPA)
- Sun-3/Motorola 68881
- SPARCstation 1 (Sun-4c kernel)/Weitek 3170 or TI 8847 (FPU)
- Sun-3 (Sun-3x kernel)/68882/Weitek 1164/5

There are some differences for Sun-3x kernel (Motorola 68030-68882) and Sun386*i* (Intel 80386-80387) from the rest of the machines used for exception handling. The differences are noted in "`ieee_handler` and UNIX FPE Signals" and "Debugging Floating Point Exceptions," below.

To run on a Sun-3 or a Sun-3x, it is necessary to specify `-ffpa` or `-f68881` in order to get the IEEE exception handling. On Sun-4 and Sun386*i* systems, this capability is the default.

Sun FORTRAN implements IEEE 754 with several library routines which may or may not access UNIX kernel trap handling routines to perform user-specified operations.

IEEE Numbers

The IEEE standard specifies different types of floating point numbers, single and double precision, sub-normal, positive and negative infinity, and NaN (Not a Number).

For more information, refer to the `libm_double(3f)`, `libm_single(3f)`, and `ieee_values(3m)` online `man` pages.

The following table specifies how to get a desired precision in IEEE value.

Desired IEEE Value	Double Precision	Single Precision
`infinity`	`x=d_infinity()`	`r = r_infinity()`
`quiet NaN`	`x=d_quiet_nan()`	`r = r_quiet_nan()`
`signaling NaN`	`x = d_signaling_nan()`	`r = r_signaling_nan()`
`min_normal`	`x = d_min_normal()`	`r = r_min_normal()`
`min_subnormal`	`x = d_min_subnormal()`	`r = r_min_subnormal()`
`max_subnormal`	`x = d_max_subnormal()`	`r = r_max_subnormal()`
`max_normal`	`x = d_max_normal()`	`r = r_max_normal()`

Example 1 next illustrates some common values for single precision from FORTRAN on a Sun-4, which are also valid for a Sun-3 and and a Sun-386*i*. A bug has been filed on the output format on the Sun-386*i* for NaN, but the routines still work.

Example 1: Common Values for Single Precision

```
c   example program to generate IEEE special values.
c   The special values are: infinity, quiet NaN, signaling NaN,
c   min_subnormal, max_subnormal, min_normal, max_normal.
c
c   Refer to ieee_values(3m) and <f77/f77_floatingpoint.h>


#include <f77/f77_floatingpoint.h>

      program print_ieee_values
c the next 2 implicit statements are necessary so that the
c f77_floatingpoint
c pseudo-intrinsic functions are declared with the correct type
c single precision only in this example for the sake of brevity
      implicit double precision (d)
      implicit real (r)

      real r

      r = r_infinity()
      print *, 'r = r_infinity() : ', r
      write (*, 27) r
 27   format ('in hex, r = ', Z8.8)

      r = r_quiet_nan()
      print *, 'r = r_quiet_nan() : ', r
      write (*, 28) r
 28   format ('in hex, r = ', Z8.8)

      r = r_signaling_nan()
      print *, 'r = r_signaling_nan() : ', r
      write (*, 29) r
 29   format ('in hex, r = ', Z8.8)

      r = r_min_subnormal()
      print *, 'r = r_min_subnormal() : ', r
      write (*, 30) r
 30   format ('in hex, r = ', Z8.8)

      r = r_max_subnormal()
      print *, 'r = r_max_subnormal() : ', r
      write (*, 31) r
 31   format ('in hex, r = ', Z8.8)

      r = r_min_normal()
      print *, 'r = r_min_normal() : ', r
      write (*, 32) r
 32   format ('in hex, r = ', Z8.8)

      r = r_max_normal()
      print *, 'r = r_min_subnormal() : ', r
```

```
      write (*, 33) r
 33   format ('in hex, r = ', Z8.8)

      end
```

```
system 191: f77 f.f
f.f:
 MAIN print_ieee_values:
system 192: a.out
r = r_infinity() :  Inf
in hex, r = 7f800000
r = r_quiet_nan() :  NaN
in hex, r = 7fffffff
r = r_signaling_nan() :  NaN
in hex, r = 7f800001
r = r_min_subnormal() :     1.40130E-45
in hex, r = 00000001
r = r_max_subnormal() :     1.17549E-38
in hex, r = 007fffff
r = r_min_normal() :     1.17549E-38
in hex, r = 00800000
r = r_min_subnormal() :     3.40282E+38
in hex, r = 7f7fffff
 Warning: the following IEEE floating-point arithmetic exceptions
 occurred in this program and were never cleared:
 Invalid Operand;
```

ieee_flags

The subroutine `ieee_flags` takes the following form:

```
int ieee_flags(action,mode,in,out)
char *action, *mode, *in, **out;
```

The subroutine `ieee_flags` is used for two classes of functions, as follows:

- To control precision, rounding modes, and so on
- To return or clear exceptions status

There are four types of actions: `'get'`, `'set'`, `'clear'`, and `'clearall'`. `'set'` and `'clearall'` are primarily used to control precision, rounding modes, and so on. `'get'` and `'clear'` are used to get exceptions status, and to clear the exception status after an exception has occurred.

The `'set'` action can be used to set precision or rounding mode, while `'clearall'` is used to restore the default as existed before the user used `'set'`. `'clear'`, when used with the mode `'exception'`, clears the exception flags for any exceptions that have occurred up until that time. `'get'` `'exception'` tells the user if an exception of a particular type has occurred.

Other combinations exist, but these are the most commonly used.

Mode can be set to `'direction'`, `'precision'`, or `'exception'`, where `'direction'` sets the direction of rounding, `'precision'` sets the precision of an operation, and `'exception'` returns the status or clears the status, depending on the action specified. Note that `'precision'` affects only the precision of the intermediate results in extended registers on the Motorola 68881 and 68882, and the Intel 80387 chips.

The `'in'` parameter controls the type of exception status that is returned. Possible condition flags are `'inexact'`, `'division'`, `'underflow'`, `'overflow'`, and `'invalid'`. `'common'` will get only `'overflow'`, `'invalid'` and `'division'` exceptions. If `''` is used for `'in'`, the the highest priority exception status will be returned. (Priority order is `'invalid'`, `'overflow'`, `'division'`, `'underflow'`, and `'inexact'`.)

Example 2 next uses `ieee_flags` to get exceptions status and to clear the exceptions. To control computation precision or rounding, `ieee_flags` would be called with the appropriate parameters.

Example 2: `ieee_flags` *Used to Get Status and Clear Exceptions*

```
      Program  flags
      real*4  a,b

          print *, ' UNDERFLOW'
      a =  2.0**(1-127) * 0.001
          call pflag

          print *, ' OVERFLOW'
      b = 1.0/a
          call pflag

         print *, 'INVALID'
      a = b/(b+1.0)
          call pflag

          print *, 'INEXACT'
         a=10.0
          b=3.0
      i = a/b
          call pflag

          print *, 'DIVISION'
          a=0.0
      a = b/a
          call pflag
      end
c
      subroutine pflag
c         The purpose of this subroutine is to
c         print out the current ieee exception
c         using a 'C' wrapper around ieee_flags,
c         and to clear the exceptions using
c         ieee_flags from fortran.
c         In Fortran 1.3, it will not be necessary to
c         call ieee_flags from 'C' in order to
c         print out the exceptions.  This will be
c         able to be done directly from Fortran.

c     print and clear ieee
      integer*4 ieee,ieee_flags
          external flags_c
          parameter (l=16)
          character*(l) out

c         In Fortran 1.2 or earlier, it is necessary to call
c         a "C" routine to print out exceptions from ieee_flags.
c         This will not be necessary in Fortran 1.3.
c         It will then be possible to call ieee_flags to get
c         exceptions status just as done below to clear the ieee flags.

          ieee = flags_c('get','exception','all',out)
```

```
c

c       Clear ieee flags so no warning messages will appear
c       and so can test for new exceptions.

      ieee = ieee_flags('clear','exception','all',out)
      return
      end


int flags_c_(action,mode,in,out)
     char *action, *mode, *in, **out;
{
int i,j,ieee_flags();
float x;
char *pt;
    i = ieee_flags(action,mode,in,out);
    printf(" i = %3i  out = %s\n",i,*out);
}


system 223: cc -c c.c
system 224: f77 f.f c.o
f.f:
MAIN flags:
    pflag:
system 225: a.out
 UNDERFLOW
 i =   5  out = underflow
 OVERFLOW
 i =   9  out = overflow
INVALID
 i =  16  out = invalid
INEXACT
 i =   1  out = inexact
DIVISION
 i =   2  out = division
```

ieee_handler and UNIX FPE Signals

`ieee_handler` is a math library routine which enables the user to set up his or her own exception handler. This is where actions such as aborting execution, decoding of the exception trap, or getting the address of the instruction that caused the exception, and others can be specified.

One of the most common uses of an exception handler is to determine the section of program code in which the exception occurred. For this purpose, it is important to realize that the UNIX signal `SIGFPE` must be signaled. `SIGFPE` is not signaled by default, but only when a signal handler is established for the exception. This is especially important when using the `dbx 'catch FPE'` command to locate exceptions.

It is not useful to call `ieee_flags` from within an exception handler, because the `ieee_flags` will not reflect the status of the exception that caused the trap. This makes it necessary to examine the parameter "code" in the handler (see Example 3 next).

Also interesting is that FPE signals are not generated by default in all cases unless the user sets up an exception handler of some kind. This is especially important when using the `dbx 'catch FPE'` command to locate exceptions. `dbx` uses UNIX FPE signals to determine where an exception has occurred. Therefore, in most cases if one wants to use `catch FPE`, then a user exception handler must be established.

Only one signal is generated at any one time and the highest priority signal code is returned to the user-defined handler in the parameter "code" (as in Example 3 next). There are two methods of decoding the signal to determine what kind of exception occurred. One method is presented below. The other method, using the FPE signals specified in `sys/system.h`, can be implemented in a similar way, except that the decode routine must be written in C. The parameter to the user exception handler "code" must be tested against the values in `signal.h`.

Example 3 next shows how to use an exception handler to determine the type and location of an exception. Please note that this example works differently on a Sun386*i*. It is necessary to put in an `exit(1)` call in the handler or the program will loop. This is due to the hardware and is not a bug. It is possible to catch only the first exception. Additionally, this example will not work on Sun-3x (Motorola 68030-68882 processor) systems.

Example 3: Exception Handlers to Determine Exception Type and Location

```
#include <values.h>
#include <f77/f77_floatingpoint.h>
c  Ensure that this file is a .F file so C preprocessor invoked
c generate the 5 IEEE exceptions:
c invalid, division by zero, overflow, underflow and inexact

      program generate_ieee_exceptions
      external handler
      integer ieeer
      double precision a,b

c use ieee_handler to establish the function "handler" as the
c signal handler to use whenever any floating point exception occurs

       ieeer=ieee_handler('set', 'all', handler)
       if (ieeer.ne.0) print *, 'ieee_handler cannot set "handler" '
c If the user does not want to trap inexact, for example,
c then call
c      ieeer=ieee_handler('clear', 'inexact', handler)
c      if (ieeer.ne.0) print *, 'ieee_handler cannot set "handler" '
c This will leave "invalid", "division", "underflow" and
c "overflow" exception handling established.



       print *, 'INVALID'
        a = log(-37.4)

       print *, ' OVERFLOW'
       b = MAXDOUBLE
        a = MAXDOUBLE
        a = a + b

        print *, 'DIVISION'
        a=0.0
       a = b/a

       print *, ' UNDERFLOW'
       b = MINDOUBLE
       a = b/2.0

       print *, 'INEXACT'
        a=10.0
        b=3.0
        a = a/b

        end

      integer function handler (sig, code, sigcontext)
      integer sig,code,sigcontext(5)
      character label*16
```

```
      if (loc(code).eq.208) label='invalid'
      if (loc(code).eq.200) label = 'division'
     if (loc(code).eq.212) label = 'overflow'
     if (loc(code).eq.204) label = 'underflow'
     if (loc(code).eq.196) label = 'inexact'
      write (6,77) loc(code), label, sigcontext(4)
 77   format ('ieee exception code ', i3, ',',
     * a17, ',', ' occurred at pc ', i5 )
     end

# Note that this file is f.F (uppercase F) in order to use
# the 'C' preprocessor.

system 291: f77 f.F
/tmp/cpp.05437.0.f:
 MAIN generate_ieee_exceptions:
    handler:
system 292: a.out
INVALID
ieee exception code 208, invalid          , occurred at pc  9124
 OVERFLOW
ieee exception code 212, overflow         , occurred at pc  9284
DIVISION
ieee exception code 200, division         , occurred at pc  9420
 UNDERFLOW
ieee exception code 204, underflow        , occurred at pc  9564
INEXACT
ieee exception code 196, inexact          , occurred at pc  9724
```

ieee_retrospective and IEEE Warning Messages

The following are examples of the IEEE floating point messages.

```
Warning: the following IEEE floating-point arithmetic exceptions
occurred in this program and were never cleared:
Inexact;  Division by Zero;  Underflow;  Overflow;  Invalid Operand;
```

`ieee_retrospective` is the FORTRAN library routine that puts out these messages and it is by default called when a FORTRAN program exits. However, `ieee_retrospective` can be called from anywhere in the user program, anywhere the user wishes to see which exceptions have occurred. This may be more convenient than to call `ieee_flags` as described in "`ieee_flags,`" above.

What Do the Error Messages Mean?

Division by zero, underflow, and overflow are exactly what they say they are. An inexact exception occurs whenever the result of a floating point operation cannot be represented exactly by a binary number, which is to say, most of the time. For example, 2.0 is an exact binary number, as is 0.5, but 3.0 or 1/3.0 are not. Invalid exceptions arise when no numerical result makes sense, as in the cases of infinity-infinity, infinity*0, 0/0, sqrt(-1), log(-1), and so on. These can be represented as a NaN (Not a Number), positive infinity, or negative infinity.

What Do the Warning Messages Mean?

Warning messages are generated by the library routine `ieee_retrospective`. If the user does not wish to see the messages for one reason or another, then an empty subroutine such as the following can be linked with the user code:

```
subroutine ieee_retrospective
end
```

This dummy routine will be called instead of the library routine, and the messages will not appear. This should only be done if the user is certain that all the exceptions that occur are harmless, or have been taken care of.

The other way to get rid of the warning messages is, of course to call `ieee_flags` with `'clear', 'exceptions'` as in Example 3 above. Add the following statement to your FORTRAN program at any location you want the flags to be cleared:

```
ieeer=ieee_flags('clear','exception','all',out)
```

Debugging Floating Point Exceptions

Users often need to debug a floating point exception. There are several ways to use `dbx` in this instance. `dbx` must be able to catch an FPE signal in order to work. It is necessary for the user to set up an exception handler in order for FPE signals to be generated. For performance reasons, the default situation in many cases is to NOT set up an exception handler. Therefore, it is necessary to set up an exception handler which will cause exception FPE signals to be generated and then `dbx` can catch these. This is not possible with the `-fsoft` option on the Sun-3 system.

To use `dbx`, one must compile the source with the `-g` option, issue the command `dbx a.out`. The first command to `dbx` should be `catch FPE` and then `run`. This should give the user the source line where the exception occurred. To continue with the program, type `cont`, as in example 4 next.

A second approach is to set up an exception handler which traps SIGFPE code (as in example 3 above), give the command `'stop in sample_handler'` then run the code. Use `'where'` to find the location of the exception. The user can also call `abort()` in the exception handler, run with `dbx`, then type `'where'` to see where the exception occurred.

Note that it is only possible to catch the first exception on a Sun-386*i* system, because the hardware does not increment the program counter and as a consequence, the same instruction that caused the exception is re-executed. This causes the program to stay at the same place.

Using example 3 above, but compiling with `-g`, the code shown in example 4 next is returned. For a Sun-3 system with a floating point accelerator, it is necessary to compile with `f77 -g -f68881` in order to catch the exceptions under `dbx`. Please note that this example will not work a Sun-3 system with Sun-3x kernel architecture (Motorola 68030-68882 coprocessor).

Example 4: Code Returned when Compiled with -g

```
system 366: f77 -g f.F
/tmp/cpp.05587.0.f:
 MAIN generate_ieee_exceptions:
    handler:
```

Example 4-1: catch FPE

```
system 367: dbx a.out
Reading symbolic information...
Read 326 symbols
(dbx) catch FPE
(dbx) run
Running: a.out
INVALID
signal FPE (floating point exception) in MAIN at line 19 in file "f.F"
   19           a = log(-37.4)
(dbx) cont
 OVERFLOW
signal FPE (floating point exception) in MAIN at line 24 in file "f.F"
   24           a = a + b
(dbx) cont
DIVISION
signal FPE (floating point exception) in MAIN at line 28 in file "f.F"
   28           a = b/a
(dbx) cont
 UNDERFLOW
signal FPE (floating point exception) in MAIN at line 32 in file "f.F"
   32           a = b/2.0
(dbx) cont
INEXACT
signal FPE (floating point exception) in MAIN at line 37 in file "f.F"
  37            a = a/b
(dbx) cont
program exited with 0
(dbx) quit
```

Example 4-2: Stopping in the exception handler

```
system 369: dbx a.out
Reading symbolic information...
Read 326 symbols
(dbx) stop in handler
(2) stop in handler
(dbx) run
Running: a.out
INVALID
stopped in handler at line 44 in file "f.F"
   44         if (loc(code).eq.208) label='invalid'
(dbx) nexti
```

```
stopped in handler at 0x2644
handler+0x28:    cmp     %o0, 208
(dbx) cont
ieee exception code 208, invalid          , occurred at pc  9124
 OVERFLOW
stopped in handler at line 44 in file "f.F"
   44          if (loc(code).eq.208) label='invalid'
(dbx) nexti
stopped in handler at 0x2644
handler+0x28:    cmp     %o0, 208
(dbx) cont
ieee exception code 212, overflow         , occurred at pc  9284
DIVISION
stopped in handler at line 44 in file "f.F"
   44          if (loc(code).eq.208) label='invalid'
(dbx) nexti
stopped in handler at 0x2644
handler+0x28:    cmp     %o0, 208
(dbx) cont
ieee exception code 200, division         , occurred at pc  9420
 UNDERFLOW
stopped in handler at line 44 in file "f.F"
   44          if (loc(code).eq.208) label='invalid'
(dbx) nexti
stopped in handler at 0x2644
handler+0x28:    cmp     %o0, 208
(dbx) cont
ieee exception code 204, underflow        , occurred at pc  9564
INEXACT
stopped in handler at line 44 in file "f.F"
   44          if (loc(code).eq.208) label='invalid'
(dbx) nexti
stopped in handler at 0x2644
handler+0x28:    cmp     %o0, 208
            (dbx) cont
ieee exception code 196, inexact          , occurred at pc  9724

execution completed, exit code is 0
program exited with 0
(dbx) quit
```

Underflow Exceptions and Performance

Performance is always a very important issue. With the Sun-3/Sun FPA, Sun-4/FPU, Sun-386*i*/FPX, and Sun-4-SPARCstation/FPU2, it is possible to get increased performance from FORTRAN by calling `abrupt_underflow`. This takes advantage of what is called the "fast mode" of the Weitek and TI floating point co-processors in order to increase floating point computation speed with subnormal numbers. When `abrupt_underflow` is called from a FORTRAN program, all subnormal operands are flushed to zero from that point on, thereby preventing expensive recomputation and underflow traps.

Note that this does not mean that subnormal results are flushed to zero, as shown in Example 5 next. Since this routine does not conform to the IEEE floating point standard, another routine, `gradual_underflow`, is provided to return the floating point to normal floating point mode. These routines can be called from anywhere in a program, and can be called as often as is necessary.

In order to enhance performance, it is also necessary to **not** establish an exception handler as this requires UNIX kernel overhead. Example 5 next demonstrates the use of this routine. The first run uses an exception handler to count the number of underflows; the second run disenables the exception handler. The performance benefit comes when subnormal operands are flushed to zero as in the fourth case.

Example 5: Using not *to Establish an Exception Handler*

```
#include <values.h>

#ifdef SUB
#define VAL MINDOUBLE
#else
#define VAL 2.2e-305
#endif


/* MINDOUBLE is defined in /usr/include/values.h to be the
   minimum subnormal number IEEE double precision can hold -
  namely, 4.9406564584124654e-324.

   The IEEE minimum for normal numbers, double precision, is
   around e-308.  So the value 2.2e-305 is a tiny number
   near the underflow threshold.    */


      program under
      common /counters/ underflow_counter
      integer ieeer, i, underflow_counter
      double precision x, y
      external underflow_handler

      underflow_counter = 0

#ifdef ABRUPT
      call abrupt_underflow()
#endif

      ieeer=ieee_handler('set', 'underflow', underflow_handler)
      if (ieeer.ne.0) print *, 'Could not set underflow handler'

      do 10 i= 1, 10000
         x = VAL
         y = x * 0.01d-4
 10      continue

    print *, 'underflow counter = ' , underflow_counter

         end


       integer function underflow_handler(sig,code,sigcontext)
       integer sig, code, sigcontext(5)
        common /counters/ underflow_counter
    integer underflow_counter

    underflow_counter = underflow_counter + 1
c        write (0, 77) loc(code), sigcontext(4)
77     format ('exception # ', i3,' occurred at pc ',i5)
```

```
        end

# This command is used to initiate the runs below.

echo "case 1: normal operands"
        f77 -o under.case1 f.F
time         under.case1

echo "case 2: normal operands, call abrupt_underflow"
        f77 -o under.case2 -DABRUPT f.F
time       under.case2

echo "case 3: subnormal operands"
        f77 -o under.case3 -DSUB f.F
time  under.case3

echo "case 4: subnormal operands, call abrupt_underflow"
        f77 -o under.case4 -DSUB -DABRUPT f.F
time  under.case4

system 435: cmd
case 1: normal operands
/tmp/cpp.05890.0.f:
 MAIN under:
    underflow_handler:
underflow counter =   10000
        7.0 real         0.8 user         6.1 sys
case 2: normal operands, call abrupt_underflow
/tmp/cpp.05897.0.f:
 MAIN under:
    underflow_handler:
underflow counter =   10000
        7.1 real         0.8 user         6.2 sys
case 3: subnormal operands
/tmp/cpp.05904.0.f:
 MAIN under:
    underflow_handler:
underflow counter =   10000
        7.3 real         0.8 user         6.4 sys
case 4: subnormal operands, call abrupt_underflow
/tmp/cpp.05911.0.f:
 MAIN under:
    underflow_handler:
underflow counter =   0
        0.1 real         0.0 user         0.0 sys
```

Testing Floating Point Hardware

To test floating point hardware, use the following commands. For further information, consult the appropriate `man` page(s). In most cases, it necessary to be superuser (`su`) to use these commands.

- Sun-4 systems:

  `/usr/diag/sundiag/sundiag` (tests more than `fpu`)
  `/usr/diag/sundiag/fputest re`
  `/usr/diag/fpurel -v`

- Sun-3 systems:

  `/usr/etc/fpa/fparel -v`
  `/usr/diag/sundiag/fpatest`

  Additional tools in `/usr/etc/fpa`

- Sun-386*i* systems:

  `/usr/sysex/sysex` (tests more than `fpx`)

3

STB SHORT SUBJECTS

3

STB SHORT SUBJECTS

Applications and *mmap(2)*

Applications Use *mmap(2)*

Those wanting to map memory may use the *mmap(2)* system call in an application program only. This call cannot be used by device drivers, since they use lower level kernel interfaces.

`mmap()` Overview

`mmap()` establishes a mapping between the process's address space at a specified address and length to the memory object.

The process address space is an implementation-dependent function, and a successful `mmap()` call returns the process address space as its result. A failing `mmap()` returns '-1'.

For Further Information

For details, see the *mmap(2)* system call man page, plus the following additional pages:

- *fork(2)*
- *getpagesize(2)*
- *munmap(2)*
- *mprotect(2)*

`rlogin` and `cmdtool`

Using `rlogin` within `cmdtool` Windows

Customers using `rlogin` from a command tool window may see the below message just before the window disappears.

```
reset tty pgrp from xxx to yyyy
```

`cmdtool` receives the above message because the window is dying for one of several reasons. If you are sending escape or control-key sequences to your `cmdtool` from a `.login`, `.cshrc`, or any application, the `cmdtool` may die.

If it dies, then it *always* gives the above message which basically means that the `cmdtool` is no longer running in a certain `pty`, since it just died.

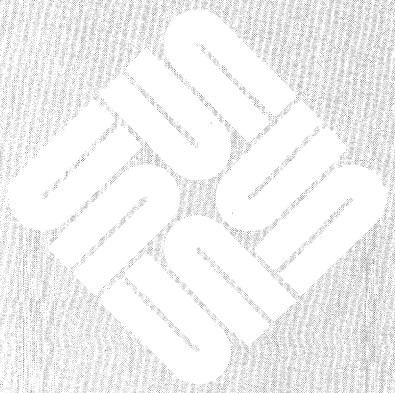
`cmdtool` Bugs

The bugs listed below summarize problems relating to `cmdtool` and escape and control-key sequences.

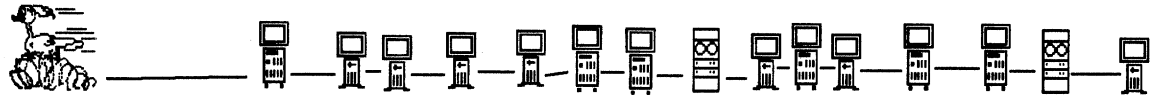
- **1004159**. You are echoing a large number of characters into a `cmdtool` without any carriage returns within the long strings.
- **1002445** and **1002446**. Using `browse`, `more` or typing Control-s key sequence causes `cmdtool` to die at times with the above message.
- **1002452**. `rlogin` hangs especially when `dbx`-ing a process, and the user 'kills' the `dbx` process to stop it. The error messages `window input queue overflow` and `window input queue flushed` may appear. In some cases the window just dies.
- **1003138**. You cannot use `csh` file completion (set `filec`), since the ESC key is caught by the `cmdtool` and not passed onto the application.
- **1004196**. The Control-c sequence is not sent to the child process in the `cmdtool`, but is caught by the `cmdtool`. The `cmdtool` puts out spurious characters or it core dumps.
- **1004456**. Audible or visible do not work in a `cmdtool`. Echoing these produces spurious results.
- **1005090**. Pressing keys on the right keypad causes the `cmdtool` to die.
- **1005308**. Type-ahead does not always work in a `cmdtool`, especially if many characters are being sent to the `cmdtool`, causing it to die.

4

IN DEPTH



Sun386i Administration Cookbook



sun microsystems

Sun Microsystems, Inc.
East Coast Division
Two Federal Street
Billerica, MA 01821

October 1989

Trademarks

SunOS, PC-NFS, NFS, DOS Windows, Sun Organizer, Sun-3, Sun-4, Sun386i, SPARCstation, SPARC, and Sun LaserWriter are trademarks, and the Sun logo, Sun Workstation, and Sun Microsystems are registered trademarks, of Sun Microsystems, Inc.

PC is a trademark, and IBM is a registered trademark, of International Business Machines Corporation.

Ethernet is a registered trademark of Xerox Corporation.

UNIX is a registered trademark of AT&T in the USA and other countries.

VT-100 is a trademark of Digital Equipment Corporation.

Epson is a trademark of Epson America, Inc.

Hayes, Hayes Smartmodem 1200, and Hayes Smartmodem 2400 are trademarks of Hayes Microcomputer Products, Inc.

PostScript is a registered trademark of Adobe Systems, Inc. Adobe also owns copyrights related to the PostScript language and the PostScript interpreter. The trademark PostScript is only used herein to refer to material supplied by Adobe or to programs written in the PostScript language as defined by Adobe.

ClickArt is a registered trademark of T/Maker Co. 1987 T/Maker Company

SPAM is a registered trademark for a pork product packed only by Geo. A. Hormel & Co. Corp.

Wyse 50 is a trademark, and Wyse is a registered trademark of Wyse Technology.

LaserJet is a trademark of Hewlett-Packard Company.

Credits

The Sun386i Administration Cookbook was produced by Sun BOS Information Architecture and the Sun BOS Cookbook Engineering Team.

Copyright

Legal Notice:

Yellow Pages is a registered trademark of British Telecommunications plc. in the United Kingdom, and may also be a trademark of various other companies around the world. Sun Microsystems Inc. will revise future versions of software and documentation to remove references to Yellow Pages.

Table of Contents

Preface

The Sun386i workstation is a Sun workstation that uses the same underlying mechanisms as other Sun workstations. This manual points out some of the similarities that Sun386i systems share with other Sun systems, as well as the features that distinguish Sun386i workstations. The Sun386i Administration Cookbook contains tables and explanations grouped by functional area that provide high-level information about frequently performed tasks. In addition, this manual provides procedures for tasks such as disabling YP, creating multiple domains, and placing Sun386i features on nonSun386i networks, as well as explanations of file system differences, the automounter, Secure RPC, and how SNAP works.

Reminders

- This is an internal Sun document. It is not designed for customers, but instead for Sun's technical personnel so that they can better support the Sun386i product line.
- Unless otherwise specified, the information contained in this book pertains to Sun386i SunOS 4.0.2 and should be used with systems running the 4.0.2 release of the software.
- Do not contact BOS Engineering or USAC if you have any questions or comments regarding this manual. Instead, mail comments to `cookbook@East.Sun.Com`, and address questions to the standard Sun internal technical support mail aliases.
- Almost all of the procedures in this manual have been tested. Although we have made a good-faith effort to emulate real-world network environments in the lab, you may run into site-specific problems. Please let us know the details if you do.
- The procedures and Sun386i configurations contained in this document are fully supported by Sun.

Restricting Sun386i Features

The Sun386i system was designed for less technical users and administrators. The features introduced to help simplify the tasks of the target audience sometimes complicate the lives of more technically proficient users and administrators. However, because Sun386i software is based on the same SunOS software that other Sun systems use, you can restrict Sun386i-specific features to make the environment almost identical to that of Sun-3, Sun-4, and SPARCstation systems. Specifically, you can:

- Disable automatic installation of systems via Automatic System Installation and prevent users from creating their own accounts by editing the `/etc/policies` file on the YP master (see the "`policies` Map" section in Chapter 7 for details).
- Install a new Sun386i system on a non-Sun386i network by choosing the option to join the network as a YP client, which disables configuration probing on the new system.
- Administer systems by using familiar Sun-3/4 techniques, editing files yourself instead of using SNAP.

Terminology

Definitions of network roles are confusing, regardless of the architecture. New network roles were defined for Sun386i systems to help clarify these roles.

In the following table and throughout this manual, the icons below represent different architectures:

Sun-3, Sun-4, and SPARCstation systems

Sun386i systems

Network Role	Architecture	Definition
Boot Server		A Sun386i YP server that answers configuration probing requests via the `pnp` daemon. Unlike other Sun systems, by default all Sun386i clients require a server to boot (even to single-user mode).
Cluster Server		A Sun386i system with a disk that has the optional clusters loaded and exported; clients access clusters from this system.
Dataless Client		A system with its own disk, containing the `/` and `swap` partitions; relies on server for major file systems (such as `/usr`) before booting in single- or multi-user mode. Similar to Sun386i diskful client.
Diskful Client		A system with a disk and a partial copy of the files that are bundled with SunOS; relies on an assigned server for `/usr`, `/usr/local`, and `/usr/cluster` to boot and operate. Similar to Sun-3/4 dataless client.
Diskless Client	 	A system without its own disk; must rely on a server for /(root), `/usr`, and `/files` to boot, access applications, and store files.
Home Directory Server	 	Any system with a disk that provides disk space for users' home directories.
Master Server	 	Sun-3/4 definition – A master YP server that provides YP lookup and update services to other systems, and contains the master copies of YP database files. Sun386i definition – Same as Sun-3/4 definition, except that it also provides resource allocation services (IP addresses, UID/GID) to other systems.
Network Client		A system with a disk and a copy of the files that are bundled with SunOS. Relies on a master or slave server to boot, and can install itself automatically if the YP master is a Sun386i with ASI enabled. A YP client with configuration probing enabled.
Slave Server	 	A system that provides YP lookup services to other systems, and maintains copies of database files from the YP master server.

continued on the next page

Network Role	Architecture	Definition
Standalone		Sun-3/4 definition – A system with its own disk, containing `/`, `/usr`, and `swap` partitions, that does not need a server to boot. Can operate on its own or be connected to a network. Sun386i definition – Same as Sun-3/4 definition, except that a Sun386i standalone is not connected to a network and is the YP master server of its own domain.
YP Client		Sun-3/4 definition – A system that uses YP maps from a YP server, and requires a YP server to boot. Sun386i definition – Same as Sun-3/4 definition. Also, a network client with configuration probling disabled.
Mail Server		The system to which mail is sent for a given user.

Chapter 1:
Peripherals Administration

Legend

- ★ Special features are implemented on Sun386i systems
- D Sun386i default settings are different from Sun-3 or Sun-4 SunOS 4.0 systems
- ◆ Explicit steps required to retain SNAP compatibility or other ease-of-use features
- Performing this task may prevent subsequent use of SNAP or ease-of-use features
- Sun386i software cluster must be loaded from tape or diskette
- 🐞 Some bugs or missing functionality in release 4.0.2
- ✓ Sun386i SunOS 4.0.1 bugs are fixed in 4.0.2

1.1 Adding and Configuring Disks

Except for cabling and hardware differences, you can install an additional Sun386i™ supported disk on a Sun386i system just as you would on other Sun workstations with `mkfs`(8) or `newfs`(8) and edits to `/etc/fstab` and `/etc/exports`.

Because Sun386i systems do not use `suninstall`, you cannot use the Sun-3™, Sun-4™, or SPARCstation™ `suninstall` procedure to install a disk. However, the procedures for installing a Sun386i expansion unit with a disk, shown in Sun386i System Setup & Maintenance, are simple and straightforward.

For simplicity, Sun386i documentation encourages the use of just one disk partition (`c`), encompassing the entire disk, for extra disks added to Sun386i systems. This partition and additional partitions are mounted on `/filesn`, a Sun386i-specific partition where n represents the order in which the disk was added (most of the system disk is `/files`, the first additional disk is `/files1`, and so on). As an alternative, you can set up multiple partitions on a single disk, or choose not to use the `/filesn` naming convention.

If you add a second disk to a Sun386i system and you plan to routinely boot from the system disk, run `newfs`(8) on the additional disk to overwrite any system software that might be there. Sun386i SNAP Administration (June 1989 edition) provides details.

Exporting — If you are exporting the file system(s) on an additional disk, it's recommended that you establish a symbolic link from `/export` to the appropriate `/filesn` partition(s) and export that link, rather than exporting the partition directly. This is because Sun386i system disks are shipped with the three partitions `/`, `/usr`, and `/files`, with all extra space in `/files`. `/export` is a directory, not a file system, so using symbolic links prevents `/export` from filling up the root file system. Using `/export` also takes advantage of the fact that symbolic links are followed at mount time; it is the file to which the symbolic link resolves that is actually mounted. Chapter 9 provides more information.

SCSI device numbers — In the Sun386i system unit, the default SCSI disk unit number is 2 (`sd2`). In the expansion unit, the default disk unit number is 0 (`sd0`), the same default as for Sun-3/4 or SPARCstation external disks.

Boot order — The default boot order for Sun386i systems is:

- `fd0`
- `st0`
- `sd2`
- `sd0`
- `ie0`

Reference: Chapter 9 of this manual ("About `/export`" section)
Sun386i System Setup & Maintenance (Appendix A)
Sun386i SNAP Administration (June 1989 edition, Chapters 1, 4, Appendix A)
Sun386i Advanced Administration (February 1989 edition, Chapter 7)
System & Network Administration (Chapter 10)

1.2 Installing Modems

The `/etc/ttytab` file works exactly as on Sun-3, Sun-4, and SPARCstation SunOS 4.0 systems; you can install modems using the standard Sun-3, Sun-4 or SPARCstation procedures, which include editing `/etc/ttytab` and `/etc/remote`, with one exception: On Sun386i systems, carrier detect is controlled by `ttysoftcar` instead of by rebuilding the kernel.

Avoiding kernel rebuild — To simplify administration, Sun386i systems allow you to skip the traditional kernel rebuild step. Perform the following steps as root on the system that will have the modem:

1. Add an entry for the modem to the `/etc/remote` file. Sun386i Advanced Administration and the `remote`(5) `man` page provide examples.
2. Set the `remote` status flag in `/etc/ttytab`. This flag indicates that `getty`(8) should wait for carrier detect when opening this serial port, and should set the baud rate specified.

   The argument to use with `/usr/etc/getty` in `/etc/ttytab` is Dbaudrate, and has the format:

   port `"/usr/etc/getty Dbaud" dialup` device_status status_flags

   Example:

   ```
   ttya  "/usr/etc/getty D2400"  dialup  on  remote
   ```

3. Type **`/usr/etc/ttysoftcar -n /dev/tty`**port. This tells the kernel that the modem will provide the carrier detect signal, and eliminates the need to rebuild the kernel.
4. Type **`kill -1 1`** to restart the `init` process (effectively enabling the modem).

If you want to ensure that the modem is recognized each time you reboot, include the `ttysoftcar` command in step 3 in a `/etc/rc.*` file.

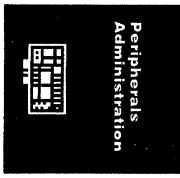
`remote` flag — Sun386i systems support the `remote` status flag in `/etc/ttytab`, which instructs the serial port driver to wait for the state of carrier detect to be asserted before opening the serial port.

MS-DOS — On Sun386i systems, MS-DOS® normally tries to attach the serial port (`COM1`) when you open a DOS Window. Therefore, before administering a modem or terminal (with or without SNAP) make sure that MS-DOS is not using the serial port. Check the DOS Windows™ Device menu and, if necessary, detach `COM1` and reboot the DOS Window.

`/etc/ext_ports` entries — For a modem to be visible to SNAP, you must make entries in the `/etc/ext_ports` file on the Sun386i master YP server and then rebuild YP. SNAP determines port availability by checking `/etc/ext_ports`, instead of the local `/etc/ttytab` file. If a device does not have an entry in `/etc/ext_ports`, SNAP could inadvertently overwrite existing modem or terminal entries in the local `/etc/ttytab` file if a user assigns a device to a port already in use. (SDR 6206)

`/dev/cu*` entries — To recognize a modem, SNAP also must find a `/dev/cu*` dialer device entry (similar to `/dev/ttyd*` and `/dev/cua0` for dial out on Sun-3, Sun-4, and SPARCstation systems) that corresponds to the serial port where the modem is connected. SNAP checks the name of the port used and then creates a corresponding dialer device name that begins with `cu` instead of `tty` (when installing a modem, the dialer device name for `ttya` is `cua`). If you create a dialer device entry in `/dev` yourself with the `mknod`(8) command and you want SNAP to recognize the modem, the name you supply must adhere to this naming scheme. For instance, if you want to use SNAP to administer a modem connected to `ttya`, create a `cua` entry in the `/dev` directory, not a `cua0` entry as stated in Sun386i Advanced Administration.

Hayes modems only — You can use SNAP only to add Hayes™ and Hayes-compatible modems.



Sun386i Advanced Administration — This manual omits the `ttysoftcar` command in the procedure for installing modems. You should run this command before restarting the `init` process, as shown in "Avoiding kernel rebuild" on the previous page.

Sun386i Advanced Administration also incorrectly states that you can add only dial-out modems to the Sun386i AT-bus serial port. In fact, you can add either dial-in or dial-out modems to both the Sun386i serial port and the Sun386i AT-bus serial port.

Hayes modems — SNAP does not enable dial-in on Hayes 2400 or 9600 modems, and the directions shown in Sun386i Advanced Administration are wrong. To use either modem for dialing in, you must perform the steps shown in the "Printers, Terminals, and Modems" section of Administrator's & Developer's Notes for Sun386i SunOS 4.0.2. (SDRs 6256, 6429)

✓ Serial port access — In Sun386i SunOS™ 4.0.1, once SNAP has owned a serial port for a modem, file protection on the `/dev/tty` entry for the port is not properly reset when you remove the modem or disable it from SunOS access. As a result, MS-DOS cannot access the port. This is fixed in 4.0.2.

Workaround for 4.0.1: To enable MS-DOS access, as superuser:

1. Disable SunOS access to the port by removing or disabling the modem with SNAP.
   If you don't want to use SNAP, then:
   a. Edit the local `/etc/ttytab` so that the device_status specified with the `/usr/etc/getty` line is **`off`**. For example:

      ```
      tty  "/usr/etc/getty D2400"  dialup  off  secure remote
      ```

   b. As root, edit `/etc/ext_ports` on the YP master server, changing the entry for this modem to **`off`**. For example:

      ```
      system_name:ttya  modem  off  2400  hayes
      ```

   c. Still as root on the YP master server, enter the following command:

      ```
      cd /var/yp; make
      ```

2. Enter **`chmod 666 /dev/tty`**port to enable MS-DOS to access the port.
3. Reboot the system.

Reference: Chapter 9 of this manual ("Adding a Modem Via SNAP" section)
Chapter 10 of this manual (`ttytab` and `ext_ports` descriptions)
Sun386i Advanced Administration (February 1989 edition, Chapter 4)
System & Network Administration (Chapter 11)
Sun386i SNAP Administration (June 1989 edition, Chapter 3)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `ttytab`(5), `remote` (5), `ext_ports` (5), `ttysoftcar`(5)
Administrator's & Developer's Notes for Sun386i SunOS 4.0.2 (for Hayes modem information)

1.3 Installing Printers

The local printcap file, `/etc/printcap`, works exactly as on Sun-3, Sun-4, and SPARCstation systems, and you can install Sun386i printers the same as Sun-3/4 and SPARCstation printers.

`ypprintcap` — In addition, YP on Sun386i systems support a domain-wide printcap file (`ypprintcap`) that you can use to centralize and simplify printer naming.

Sun386i print spooling software always looks in a workstation's `/etc/printcap` file before consulting `ypprintcap`, so you can use the local `/etc/printcap` file to override or supplement what's in `ypprintcap`.

If you don't specify a printer with the `lpr`(1) command, `lpr` checks your `PRINTER` environment variable; if it is not set, then `lpr` checks `/etc/printcap` for the default printer (`lp`). On Sun386i systems only, `lpr` then checks `/etc/ypprintcap` to locate the default printer. Also, if you don't have a local spool directory on your Sun386i system for a particular network printer, `lpr` creates the spool directory for you.

`ypprintcap` — Sun386i workstations ship default printer entries in `/etc/ypprintcap`. The default printer entries supplied are for printers that SNAP supports. See Chapter 10 for more information on `ypprintcap`.

default printer — `lp` is set up as the default printer in the `ypprintcap` file.

MS-DOS and serial printers — On Sun386i systems, MS-DOS normally tries to attach the serial port (`COM1`) when you open a DOS Window. Therefore, before administering a serial printer (whether with or without SNAP), make sure that the serial port is not being used by DOS. Check the DOS Windows Device menu and, if necessary, detach `COM1` and reboot the DOS Window.

`ypprintcap` and **`ext_ports`** — For a printer to be visible to SNAP, you must make entries in the following files on the master YP server and then rebuild YP. These files are:

- `/etc/ypprintcap`
- `/etc/ext_ports`

Local printer entries in an individual machine's `/etc/printcap` file do not affect SNAP operation, but you must administer such entries by updating the file manually (as on a Sun-3, Sun-4, or SPARCstation system) because SNAP only manages printers that are available network wide.

SNAP uses `/etc/ext_ports` to determine port availability.

Disabling a printer with SNAP — Page 69 of the revised Sun386i SNAP Administration manual is incorrect regarding the effects of disabling a printer. When you disable a printer with SNAP, jobs in the printer queue continue to print but SNAP disables the queue, preventing new requests from being printed at that printer.

`lpc abort` and **`lpc disable`** — Pages 72 and 73 of the revised Sun386i SNAP Administration manual contain incorrect statements about the `lpc` command. `lpc abort` disables printing and attempts to kill the daemon; `lpc disable` disables the queue for all users, including superuser.

Reference: Chapter 9 of this manual ("Adding a Printer Through SNAP" section)
Chapter 10 of this manual (`ypprintcap` and `ext_ports` descriptions)
Sun386i Advanced Administration (February 1989 edition, Chapter 4)
System & Network Administration (Chapter 11)
Sun386i SNAP Administration (June 1989 edition, Chapter 3)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `ext_ports`(5), `ypprintcap`(5)

Avoiding Printer Overload
To avoid overloading individual printers on large networks, assign different default printers to different users. You can set a user's default printer in either of two ways:

- Set the `PRINTER` environment variable in the user's `.login` file, so that the user's default printer is defined as something other than `lp`. This method ensures that a user has the same default printer no matter where the user logs in. Additionally, the `PRINTER` environment variable overrides the following method for altering the default printer.
- Modify the entry for `lp` in the local `/etc/printcap` file on the user's workstation. This will override the domain-wide definition for `lp`; the default printer is determined on a per-workstation basis.

1.4 Installing Terminals

The `/etc/ttytab` file works exactly as on Sun-3, Sun-4, and SPARCstation systems. The `/etc/termcap` file and `terminfo` database are identical to those on Sun-3, Sun-4, and SPARCstation systems, so you can manually add any terminal listed in the `/etc/termcap` file or `terminfo` database. On Sun386i systems, you can use SNAP to add and administer VT-100™ and WY-50™ terminals and compatibles.

`local` flag — The Sun386i workstation supports the `local` status flag in `/etc/ttytab`, which instructs `getty`(8) to ignore the state of carrier detect when opening the serial port. For terminals, this flag should be set to `local`.

MS-DOS — On Sun386i systems, MS-DOS normally tries to attach the serial port (`COM1`) when you start a DOS Window. Therefore, before administering a modem or terminal (whether with or without SNAP) make sure that MS-DOS is not using the serial port. Check the DOS Windows Device menu and, if necessary, detach `COM1` and reboot the DOS Window.

`ext_ports` — For a terminal to be visible to SNAP, you must make entries in the `/etc/ext_ports` file on the Sun386i master YP server and then rebuild YP. SNAP determines port availability by checking `/etc/ext_ports`, instead of the local `/etc/ttytab` file. If a device does not have an entry in `/etc/ext_ports`, SNAP could inadvertently overwrite existing modem or terminal entries in local `/etc/ttytab` files if a user assigns a device to a port already in use.

✓ Port access — In Sun386i SunOS 4.0.1, once SNAP has modified a serial port for a terminal, file protection on the `/dev/tty` file for the port is not properly reset when you remove the terminal or disable it from SunOS access. As a result, MS-DOS cannot access the port. This is fixed in 4.0.2.

Workaround for 4.0.1: To enable MS-DOS access, as superuser:

1. Disable SunOS access to the port by removing or disabling the modem with SNAP.
   If you don't want to use SNAP to disable SunOS access to the terminal, then:
   a. Edit the local `/etc/ttytab` so that the device_status specified with the `/usr/etc/getty` line is **`off`**. For example:

      ```
      tty  "/usr/etc/getty D2400"  vt100  off  secure  local
      ```

   b. As root, edit `/etc/ext_ports` on the YP master server, changing the entry for this terminal to **`off`**. For example:

      ```
      system_name:ttya  terminal  off  9600  wyse-50
      ```

   c. Still as root on the YP master server, enter the following command:

      ```
      cd /var/yp; make
      ```

2. Enter **`chmod 666 /dev/tty`**port to enable MS-DOS to access the port.
3. Reboot the system.

Reference: Chapter 9 of this manual ("Adding a Terminal Through SNAP" section)
Chapter 10 of this manual (`ttytab` and `ext_ports` descriptions)
Sun386i Advanced Administration (February 1989 edition, Chapter 4)
System & Network Administration (Chapter 11),
Sun386i SNAP Administration (June 1989 edition, Chapter 3)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `ttytab`(5), `ext_ports`(5)

Notes

Chapter 2:
File System Activities

Legend

- ★ Special features are implemented on Sun386i systems
- D Sun386i default settings are different from Sun-3 or Sun-4 SunOS 4.0 systems
- ● Explicit steps required to retain SNAP compatibility or other ease-of-use features
- ⚠ Performing this task may prevent subsequent use of SNAP or ease-of-use features
- 💾 Sun386i software cluster must be loaded from tape or diskette
- 🐞 Some bugs or missing functionality in release 4.0.2
- ✓ Sun386i SunOS 4.0.1 bugs are fixed in 4.0.2

2.1 File System Layout

The SunOS 4.0 file system layout differs between Sun-3, Sun-4, and SPARCstation systems and Sun386i systems.

Helpful Hints

While the new Sun386i file system layout is designed to make administrative tasks easier, there may be instances when you need to do things the "old way." Here are some hints:

Using `/usr` — There is no space provided for additional files on `/usr`, but you still can create empty directories (mount points) in `/usr`. Then you can mount local or remote file systems on the mount points just as on other Sun systems. Alternatively, you can create symbolic links from `/usr` to other partitions on the disk where more space is available. To make `/usr` writeable so that you can create the additional mount points or symbolic links, issue this command as superuser:

```
mount -o remount,rw /usr
```

If after adding the mount points or symbolic links you want to make `/usr` read-only again, reboot the system so that it reads `/etc/fstab`, which specifies the `/usr` partition as read-only.

Even before you add any mount points to `/usr`, issuing a `df`(1) command might show `/usr` as more than 100 percent full. This is normal—there should still be enough room for additional mount points or symbolic links.

Using the automounter — At existing sites, administrators can use automounted directories as complements to existing NFS™ mount points. For example, if users are accustomed to accessing an NFS-mounted directory as `/usr/eeng` they can still do so, even if the administrator has also made the directory available as `/vol/eeng` via the automounter. Newly created file systems can be made available through `/vol`, so that users no longer have to do local administration (such as creating a mount point and editing `fstab` or using the `mount`(8) command) to get access to new network file systems.

D `/usr` — The `/usr` partition is reserved for SunOS and is read-only. It appears more than 100 percent full on Sun386i systems.

Automounter — Sun386i systems start the automounter for easy access to administrator-defined network directories (`/vol`), users' home directories (`/home`), and exported file systems on other workstations (`/net`). As of SunOS 4.0, all Sun systems support the automounter, but only Sun386i systems turn it on by default.

Optional clusters — As of Sun386i SunOS 4.0.2, new Sun386i systems are shipped with all the optional clusters installed on the disk. You can increase free disk space by removing optional Application SunOS™ and the Developer's Toolkit clusters that you don't need from the disk. Systems upgrading from 4.0.1 to 4.0.2 will have the same clusters loaded after the upgrade as before; the upgrade procedure does not load all optional clusters.

Additional disks — Sun386i documentation suggests that you set up additional disks (not the system disk) with a single partition mounted as `/filesn`. In contrast, there is no suggested partitioning scheme or mount point for additional disks added to Sun-3, Sun-4, or SPARCstation systems.

`/export` — The `/export` directory on Sun386i systems contains symbolic links to directories, not the directories themselves. This is because `/export` is a directory that resides in root (`/`) and doesn't have much free space on Sun386i systems. On Sun-3, Sun-4, and SPARCstation systems, `/export` is the mount point for a disk partition.

Home directories — By convention, Sun386i home directories appear to be in `/home/username`, an automounted directory that resolves to `/files/home/groupname/username` on the user's home directory server. Sun-3, Sun-4, and SPARCstation home directories are typically in the directory `/home/servername/username`. You can mix these two conventions within a network or YP domain. See Chapter 9 for details.

Free space — Sun386i system disks consolidate all free disk space into one partition, `/files`. Sun386i systems use symbolic links and loopback mounts, both SunOS 4.0 features, to help accomplish this while still maintaining backwards compatibility with standard SunOS directories such as `/tmp`.

Reference: Chapter 9 of this manual ("Inside the Automounter" and "Inside the Sun386i File System" sections)
Sun386i SNAP Administration (June 1989 edition, Appendix A)
System & Network Administration (Chapter 6)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `lofs(4S)`

2.2 Mounting Directories

You can manually mount Sun386i directories the same as you do Sun-3, Sun-4, and SPARCstation directories: Create a mount point and then issue the `mount(8)` command. To make the mount permanent, edit the `/etc/fstab` file on the local system and then issue the `mount -a` command, just as on other Sun systems.

On all Sun systems, symbolic links are followed at mount time, so that the file that a specified symbolic link resolves to is what is actually mounted. "Inside the Sun386i File System" in Chapter 9 details how symbolic links are used on Sun386i systems.

2.3 Automounting Directories

The automounter (`/usr/etc/automount`) is a daemon that automatically mounts remote NFS file systems on first access. The automounter is available on SunOS 4.0 and later systems.

Starting the automounter — On Sun386i systems, an entry in the `/etc/rc.local` file starts the automounter. The automounter reads the `auto.master` map on the YP master and then establishes the mount points specified (if they are not already present) and manages them. By default, those mount points are:

- `/vol`, using the `auto.vol` automounter map (both Sun386i conventions)
- `/home`, using the `auto.home` automounter map
- `/net`, using the built-in `-hosts` automounter map

Sun-3, Sun-4, and SPARCstation system users can start the automounter by issuing the `automount(8)` command or by adding `automount` commands to `/etc/rc.local` and rebooting.

Home directories — Sun386i home directories are mounted on `/home/username`, while Sun-3, Sun-4, and SPARCstation home directories typically are mounted on `/home/servername/username`. For guidelines on reconciling these differences, see Chapter 9.

- Mount points — All Sun systems automount directories on mount points that begin with `/tmp_mnt`. When you display automounted directories on Sun-3, Sun-4, and SPARCstation systems, the paths begin with `/tmp_mnt/auto000xxx`. On Sun386i systems, the automounter uses loopback mounts to eliminate the need to specify `auto000xxx` in paths. Additionally, paths of Sun386i automounted directories are further simplified; typically, `/tmp_mnt` is not displayed (for example, `/home`). This is because the default `.login` file on Sun386i systems sets the `AUTOMOUNT_FIXNAMES` environment variable.

Reference: Chapter 9 of this manual ("Inside the Sun386i File System" section)
Appendix A of this manual, The Automounter, a USENIX '88 paper written by Brent Callaghan and Tom Lyon
Sun386i SNAP Administration (June 1989 edition, Appendix A)
Sun386i Advanced Administration (February 1989 edition, Chapter 6)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `auto.home`(5), `auto.vol`(5), `auto.master`(5), `automount`(8)

2.4 Exporting Directories

You export Sun386i directories just as you do Sun-3, Sun-4, and SPARCstation directories, by editing the `/etc/exports` file and issuing the `/usr/etc/exportfs` command. The SunOS 4.0 convention is that all exported directory trees have paths that begin with `/export`.

If you want to export a subdirectory on any system running SunOS 4.0, you should do so explicitly by including a line for it in the `/etc/exports` file, and making certain that none of its parent directories within the same file system are already exported. For example, you cannot export both `/a/b` and `/a/b/c`; if you do, `exportfs` returns an error.

- Symbolic links — As a Sun386i convention, all entries in `/export` are symbolic links (or subdirectories containing symbolic links) that point to the "real" location of the file system to be exported. For example, symbolic links that are shipped in `/export` on Sun386i systems are to home directories, directories for optional clusters that have been loaded, and on-line help directories. For compatibility with other Sun systems and to simplify file system administration, `/etc/exports` entries refer to symbolic links in `/export`, rather than to the real location of the local file systems. This scheme also prevents the `/export` directory from filling up.

  In contrast, on other Sun systems `suninstall` uses the `/export` directory to mount a disk partition. The `/export` partition is then used to store exported files such as `/export/root/client_name`. Sun-3, Sun-4, and SPARCstation systems support symbolic links in `/export`, but do not use them by default. For more background on how Sun386i systems use symbolic links in `/export`, see Chapter 9.

  What's exported — Also as a convention, whole file systems are not exported on Sun386i systems. This enables a finer level of control of the machines and users that can mount the directory.

Reference: Chapter 9 of this manual ("About `/export`" section)
Sun386i SNAP Administration (June 1989 edition, Appendix A)
Sun386i Advanced Administration (February 1989 edition, Chapters 6, 7)
System & Network Administration (Chapter 6)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `exportfs`(8), `exports`(5)

2.5 Backing Up and Restoring Files

You can use the dump(8), `restore`(8), `tar`(1), or `cpio`(1) commands on Sun386i systems just as on Sun-3, Sun-4, and SPARCstation systems. The DOS `BACKUP`, `RESTORE`, `COPY`, and `XCOPY` commands are useful for exchanging files with PCs, but their use is not recommended for SunOS backups since these DOS commands do not preserve certain file system information.

`bar` command — In addition, the Sun386i system provides the `bar`(1) command, a backup utility that you can use from the command line or via SNAP Backup. The `bar` command has the same options as `tar`(1), including the ability to back up individual files. In addition, `bar` offers options to compress files, follow directory symbolic links, get file names from a file, restore files to a new directory or to a new owner, check ownership before overwriting archives, and stop reading after all files are restored. `bar` also handles multiple-volume backups on both 3.5-inch diskettes and quarter-inch tapes. See the `bar`(1) `man` page for details.

Performing Full System Backups

When backing up and restoring an entire system, use the dump(8) and `restore`(8) commands instead of SNAP or the `bar`(1) command. (Be sure to make a copy of memory-based SunOS that includes the `restore` command; see the "`restore` command" bug near the end of this section.) The `bar` command attempts to restore the file `/bin/bar`; in the process, `/bin/bar` is truncated to zero length, which causes the `bar` process to core dump when it takes a page fault.

If you must restore a full system backup that was performed with SNAP or `bar`, follow these steps:

1. As root, halt the system if it is still running by entering **`/etc/halt`**.
2. Boot the system in single-user mode, without running the `/etc/rc.boot` file, by entering **`b -bsw`** at the PROM monitor prompt.
3. Mount the `/usr` and `/files` partitions by entering:

   ```
   mount -n /usr
   mount -n -o remount,rw /usr
   mount -n /files
   ```

   You must mount and then remount `/usr` because the first mount command reads the settings in `/etc/fstab`, where it states to mount `/usr` read-only. The second command mounts `/usr` as read-write.
4. Make copies of any files that could currently be executing, so that `bar` doesn't overwrite them during the restore process. In addition, these copy and move commands ensure that you will be able to reboot the system in the unlikely event that the restore does not go smoothly.

   ```
   cd /sbin
   mv sh sh.old
   cp sh.old sh
   mv init init.old
   cp init.old init
   cd /bin
   mv bar bar.old
   cp bar.old bar
   sync
   cd /
   ```

5. Insert the backup tape or diskette(s) and restore the files, as follows:

   For backups created with `bar`, but without the `-z` option

   Enter `/bin/bar.old xvfp` device

   For backups created with SNAP, or with `bar -z`

   In this case, extra steps are required to extract `/usr/ucb/compress`, or the library with which it is dynamically linked. Note that SNAP uses `/dev/rst8` or `/dev/rfd0a` when making backups.

   a. Create a temporary file called `/tmp/tmpfile` that contains the following lines:

      ```
      /usr/ucb/compress
      /lib/libc.so.2.0
      ```

   b. Run `bar.old` to extract everything except the `compress` program and the C library.

      ```
      /bin/bar.old xvfpXZ device /tmp/tmpfile
      ```

   c. Run `bar.old` again to extract `compress` into the `/tmp` directory.

      ```
      /bin/bar.old xvfpSZ device /usr/ucb/compress /tmp/compress /usr/ucb/compress
      ```

   d. Run `bar.old` again to extract the C library into the `/tmp` directory.

      ```
      /bin/bar.old xvfpSZ device /lib/libc.so.2.0 /tmp/libc.so.2.0 /lib/libc.so.2.0
      ```

   e. Move the `compress` program and the C library back to their original locations so that you have the new versions of the files.

      ```
      mv /tmp/compress /usr/ucb/compress
      mv /tmp/libc.so.2.0 /lib/libc.so.2.0
      ```

6. When the restore operation is complete, delete the following files, which you no longer need:

   ```
   rm /sbin/sh.old
   rm /sbin/init.old
   rm /bin/bar.old
   ```

7. Remove the tape or last diskette from the drive.
8. Reboot the system by typing `/etc/reboot`. This ensures that you'll run the newly restored versions of files such as `sh` and `init`, and that the `/usr` partition is mounted read-only.

D Sun386i device names — The following device names are valid for use with `bar`, `cpio`, `tar`, `dump`, or `restore` on a Sun386i system:

- `/dev/rfd0a` — Sun386i high-density diskette drive; reserves the last cylinder for error handling and mapping of bad sectors (remapping is not yet supported by the diskette driver).
- `/dev/rfdl0a` — Sun386i low-density diskette drive; also reserves the last cylinder for error handling and mapping of bad sectors, which is currently unsupported (SNAP Backup does not support this device.)
- `/dev/rfd2a` — Optional 5.25-inch diskette drive that you can purchase; reserves the last cylinder (must format with `fdformat -2` command).
- `/dev/rfd0c` — Sun386i high-density diskette drive; uses all cylinders.
- `/dev/rfdl0c` — Sun386i low-density diskette drive; uses all cylinders. (SNAP Backup does not support this device.)
- `/dev/rfd2c` — Optional 5.25-inch diskette drive that you can purchase; uses all cylinders (must format with `fdformat -2` command).
- `/dev/rst8` — Sun386i cartridge tape drive, QIC-24 format.

Notes about **dump** — On a Sun386i system, the default device that dump uses in the 4.0.1 release is `rmt8`; as of 4.0.2, the default is `rst8`. You must also specify the size of the storage media with the `-s` option:

- Tape — 2300 feet
- Cartridge — 425 feet
- 1.44 high-density Mbyte diskette — 1422 blocks

Using diskettes to dump and restore can be very time-consuming; tape is a much faster alternative. However, if you use diskettes, you must specify `-D` (for diskettes) with both the dump and `restore` commands.

Same diskette device for backup and restore — Make sure that the device name used to restore files from diskettes is the same one that was used to back up the files, regardless of which backup/restore method you use.

SNAP Backup uses `/dev/rfd0a` when copying files to diskettes. Should you use the `bar` command to restore files backed up with SNAP, specify the same `/dev/rfd0a` device.

Symbolic links — SNAP Backup uses the `bar -K` command (new as of Sun386i SunOS 4.0.2). `bar -K` follows only symbolic links that are specified on the command line.

bar archives — SNAP can read `bar` archives on tapes written in QIC-24 format (with `/dev/rst8`) or 3.5-inch, high-density diskettes (written with `/dev/rfd0a`). SNAP cannot:

- Read tapes or diskettes created with `cpio`, `tar`, or dump formats.
- Read `bar` archives on low-density 3.5-inch diskettes, high-density diskettes written with the `/dev/rfd0c` device, 5.25-inch diskettes, or tapes written with devices that do not use QIC-24 format.
- Copy files onto diskettes that have not been formatted as shown in Sun386i System Setup and Maintenance. (`bar`, `cpio`, and dump all require formatted diskettes for backup.)

advanced_admin cluster — To use the dump and `restore` commands, the optional `advanced_admin` cluster must be on the system. (The `bar`, `tar`, and `cpio` commands are shipped on every Sun386i system disk.)

Using **bar** remotely — You cannot use `bar` or SNAP to write to devices that are not attached to the local machine.

Compressing **bar** files — When you enter the `bar` command with the `-Z` option, `bar` uses the `/tmp` directory to compress and uncompress the files. If there is not enough room in the `/tmp` directory for `bar` to copy the file and compress or uncompress it, the file will be added to the `bar` archive uncompressed, or extracted from the `bar` archive and left in its compressed format. Make sure there is enough room in `/tmp` for 1.6 times the size of the largest file to be archived or extracted. (SDR 4896)

recursive symbolic links with **bar** — The `bar` command follows symbolic links when you specify the `h`, `K`, or `L` options. If `bar` tries to follow a recursive symbolic link (a symbolic link that resolves to a parent directory or the same file), it will eventually produce a segmentation fault.

duplicate `bar` archives — The `d` function modifier to the `bar` command, to create a duplicate `bar` archive, does not work.

Simultaneous SNAP backups — SNAP Backup does not prevent a user from scheduling two backups to different devices at the same time; it does not lock a user's backup catalog files for exclusive access. If you schedule two backups, they both try to update the backup catalog files simultaneously, and data can be lost from those files. This limitation applies whether you are using diskette and tape drives on the same or different systems.

SNAP Remove Entry feature — The Remove Entry feature in SNAP Restore changes the ownership of a user's backup catalog files to `root`. After using this feature, manually change the ownership of these files:

```
cd ~/.backup
chown username Backup*
```

`restore` command — In Sun386i SunOS 4.0.1 and 4.0.2, the `restore`(8) command is not loaded in memory-based SunOS (the two boot diskettes you use to reload software if your root partition has been damaged). If you have a copy of Application SunOS on diskettes, you should make a copy of the memory-based SunOS diskette (diskette #2) and place the `restore` command on the copy, by performing the following steps.

1. As superuser, insert the diskette labeled "Application SunOS diskette #2" into the drive and enter:

   ```
   dd if=/dev/rfd0c of=/tmp/image bs=9k count=160
   ```

2. Place a new, formatted diskette into the drive, and copy the image file made in step 1 to the diskette:

   ```
   dd if=/tmp/image of=/dev/rfd0c bs=9k count=160
   ```

3. The `restore` program is included with the optional `advanced_admin` cluster. Check to see if the `advanced_admin` cluster is loaded by entering **`load`**. If `advanced_admin` is not listed, then add it to the system by entering

   ```
   loadc advanced_admin
   ```

   and inserting the diskette specified.

4. Mount the diskette on `/munixfs` and copy the `restore` program to it by entering the commands below, exactly as shown:

   ```
   mkdir /munixfs
   mount /dev/fd0c /munixfs
   cp -p /usr/etc/restore /munixfs/etc
   ```

5. Enter **`sync`** and wait for the light on the front of the diskette drive to go out before proceeding to the next step.

6. Unmount the diskette and delete the image file created in step 1.

   ```
   umount /munixfs
   rm /tmp/image
   ```

7. Label the diskette "Application SunOS diskette #2 (with restore command)" or something similar.

You now have a memory-based Sun386i SunOS diskette containing the `restore` command. Should you need to restore the contents of a disk, use the diskette just created instead of the original diskette #2 supplied with the system. (SDR 5177)

Backing up/restoring complete file systems — If you use `/bin/bar` to restore an entire file system, `bar` attempts to extract the `/bin/bar` file from the media and subsequently core dumps because it cannot resolve page faults correctly. Therefore, do not use `bar` or SNAP (which uses `bar`) to back up and restore an entire file system. Use `dump` and `restore` instead (be sure to read the previous note on the `restore` command). If you must restore a full system backed up with SNAP or `bar`, perform the steps in the "Performing Full System Backups" section earlier in this chapter. (SDR 6185)

Documented **`bar`**(1), **`tar`**(1), **`dump`**(8), **`restore`**(8), **`cpio`**(1) bugs — The `man` page for each of these commands describes additional bugs.

✓ SNAP backup and restore privileges — As of Sun386i SunOS 4.0.2, if you are in the `operator` group, SNAP backup and restore functions run with root privileges (suid). You have full access to all local files, but access only to NFS files in file systems that were exported with root access, or that have read-write privileges set for everyone. If you are not in the `operator` group, SNAP backup and restore functions run with your privileges, enabling you to back up and restore any files you can access (both NFS and local files).

Using SNAP Restore remotely — As of Sun386i SunOS 4.0.1, you could not use SNAP to restore files across the network unless the file systems had been exported with root access, or unless everyone had write permission on the file.

With Sun386i SunOS 4.0.2, when a user in the `operator` group selects files for backup via SNAP, `backup` runs as `root`. This means the user in the `operator` group has all privileges to back up and restore files on the local disk. However, no NFS files can be accessed unless those files are readable and writeable by everyone, or those file systems have been exported with `root` access. Basically, `operator` backups are intended for saving files on local disks.

If the user is not in the `operator` group, SNAP backup and restore operations will run as the user who performs them, and not as `root`. In this case, only the files to which that user has read-write permissions are accessible to him or her. This is intended primarily for users backing up the files they own.

Reference: Chapter 9 of this manual ("Backing Up Data Through SNAP" section)
Sun386i SNAP Administration (June 1989 edition, Chapter 2)
Sun386i System Setup and Maintenance (Chapter 3)
System & Network Administration (Chapter 7)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `bar`(1), `tar`(1), `cpio`(1), dump(8), `restore`(8)

2.6 Establishing File System Quotas

The same quota facility is available on Sun386i, Sun-3, Sun-4, and SPARCstation systems. The preconfigured kernels shipped with Sun386i systems don't include the code for the quota facility, so you must build a new kernel to use this feature. However, there is a bug that prevents quotas from working with the `/dev/root` pseudo device, so to run quotas on Sun386i systems, perform the steps below.

1. Edit the `/etc/fstab` file, changing all `/dev/root*` entries to `/dev/sd2*`. Add the desired `quota` setting for each file system on which you want to run quotas.
2. Load the `config` and `disk_quotas` clusters from tape or diskette.
3. Follow the instructions for "Configuring the Kernel" in the `README` file in the directory `/sys/sun386/conf`.

config, disk_quotas clusters — Make sure the `config` cluster, included with the Sun386i SunOS Developer's Toolkit software, is loaded before you rebuild the kernel. To run the quota utilities, the `disk_quotas` cluster of Sun386i Application SunOS must also be on the system.

/dev/root* devices — Quota utilities don't work with `/dev/root*` pseudo devices. Edit the `/etc/fstab` file as shown previously in step 1 to work around this. (SDR 4586)

Rebuilding kernels on diskless Sun386i systems — You cannot rebuild a kernel on a diskless Sun386i client because a diskless client mounts its server's directory as read-only. However, if a diskless client has a Sun386i file server, you can rebuild its kernel on the server. If the client's file server is a Sun-3, Sun-4, or SPARCstation system, perform the steps in the "Rebuilding Kernels on Sun386i Diskless Systems" section of Chapter 4. (SDR 4573)

✓ **README** file — In Sun386i SunOS 4.0.1, the README file omitted `sun386` as a value for machine type. This is corrected in Sun386i SunOS 4.0.2. The file also referred to the Sun System Manager's Guide for further details. The correct title of this book is System & Network Administration.

Reference: Chapter 4 of this manual "Rebuilding Kernels on Sun386i Diskless Systems" section)
System & Network Administration (Chapter 7)
Sun386i Developer's Guide (Chapter 2)

Notes

Chapter 3:
Users, Groups, Logins

Legend

- ☆ Special features are implemented on Sun386i systems
- D Sun386i default settings are different from Sun-3 or Sun-4 SunOS 4.0 systems
- ● Explicit steps required to retain SNAP compatibility or other ease-of-use features
- Performing this task may prevent subsequent use of SNAP or ease-of-use features
- Sun386i software cluster must be loaded from tape or diskette
- 🐞 Some bugs or missing functionality in release 4.0.2
- ✓ Sun386i SunOS 4.0.1 bugs are fixed in 4.0.2

3.1 Controlling Automatic Account Creation

logintool — The Sun386i New User Accounts facility, a feature of the `logintool` graphical interface to logging in on Sun386i systems, lets users create their own accounts. Sites that need more control over their user accounts can restrict the New User Accounts feature on Sun386i networks.

New User Accounts — By default, users can create accounts for themselves through the `logintool` New User Accounts feature.

SNAP user accounts — Users can also create and administer user accounts with SNAP, since all users whose accounts were created with SNAP or New User Accounts have all SNAP privileges by default.

Restricting New User Accounts

To disable automatic account creation through New User Accounts:

1. Become superuser on the master YP server.
2. Edit the `/etc/policies` file and set the `newlogin` policy to **`restricted`**.
3. Enter **`cd /var/yp; make`**

For networks that are not running YP and networks with a non-Sun386i YP master, the New User Accounts feature is automatically restricted.

Users cannot create their own accounts when New User Accounts is restricted, or the system is not running YP, or the YP master is not a Sun386i system. If they try to, a message is displayed that tells them to contact their system administrator to obtain a user name and password. Administrators then can establish user accounts by editing password and group files. Since SNAP does not rely on the `newlogin` policy in the `policies` map, administrators also can use SNAP to create user accounts when `newlogin` is restricted.

Reference: Chapter 9 of this manual ("Adding a User Through SNAP," "Adding a User via New User Accounts," "Adding a Group Through SNAP" sections)
Sun386i SNAP Administration (June 1989 edition, Chapter 8)
Sun386i Advanced Administration (February 1989 edition, Chapter 3)

Disabling logintool

To disable `logintool` and use `login`(1) instead (as on Sun-3, Sun-4, and SPARCstation systems), perform the steps below:

1. Log in as root to the system where you don't want `logintool` to run.
2. Edit `/etc/ttytab`, removing the `-n`, `-l`, and `-s` switches to `/usr/etc/getty` for the console entry, so that it looks like:

   ```
   console "/usr/etc/getty std.9600" sun on secure
   ```

3. Reboot the system to activate the change.

Disabling User Account Creation with SNAP

You can also disable the ability to create and administer user accounts with SNAP, which you should do on Sun386i systems that are part of multiple domains or non-Sun386i networks. To disable user account creation with SNAP:

1. Edit the `/etc/ypgroup` file on the YP master, deleting any users listed with the `accounts` group (only members of this group can create and administer user accounts in SNAP). The `accounts` entry should look like:

   ```
   accounts:*:11:
   ```

2. Remake the YP maps by entering **`cd /var/yp; make`**

Reference: Sun386i Advanced Administration (February 1989 edition, Chapter 3)
Sun386i SNAP Administration (June 1989 edition, Chapter 5)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `group`(5), `login`(1), `logintool`(8), `getty`(1), `policies`(5)

3.2 Controlling Root Logins

Sun386i systems use the same mechanisms as other SunOS 4.0 systems to control root access.

On Sun386i systems, the defaults (any of which you can easily change) are as follows:

Root password — The root password, as shipped, is the host ID.

Console login — You can log in to the console as root since the `console` entry in `/etc/ttytab` has the `secure` flag set.

Single-user mode — No root password is required to enter single-user mode since the `console` entry in `/etc/ttytab` has the `secure` flag set.

Terminal login — You cannot remotely log in or log in from a terminal as root; you must first log in as yourself and then use the `su`(1) command to become root. This is because the `secure` flag is not set for the `tty` entries in `/etc/ttytab`. This substantially increases security and lets you track who accesses root privileges.

Changing the Defaults

You can change the root password by using the `passwd root` command as superuser, when the system is running in multi-user mode. If you change root's password in single-user mode, `passwd`(1) cannot re-encrypt root's secret key in the `/etc/.rootkey` file. After the system enters multi-user mode, you can use the `keyserv -n` command to change the `/etc/.rootkey` file. Then reboot the system.

Additionally, you can remove the root password entirely by editing `/etc/passwd`. If you later add a new password, you might have a problem with secure RPCs, so, in single-user mode, you should also remove any entry for root in `/etc/publickey`, `/etc/.rootkey`, and `/etc/keystore`. ("Secure RPC" in Chapter 9 contains more information.)

Root logins are enabled or disabled in the system's `/etc/ttytab` file through the `secure` flag. As shipped, the `ttytab` file considers only the console to be "secure." Its line looks like this:

```
console    "/usr/etc/getty -n -s -l std.9600" sun    on   secure
```

The `secure` setting implies that the physical location of the console is secure, and so allows a user to log in to the console as root. By default, root cannot log in via a terminal, modem, or remote log in on Sun386i systems, because the word `secure` is missing on the lines for the `tty` devices.

You can change root login policies as follows:

- Controlling root login via the secure flag — By removing the `secure` flag from the `console` entry in `/etc/ttytab`, you control both of the following:
  - Access as root from the console. When you remove the `secure` flag, users must log in first as themselves and then use the `su` command if they need root privileges.
  - Root access in single-user mode. Removing the `secure` flag forces a prompt for the root password when a user enters single-user mode.
- Restricting access to root via `su`(1) — Add users to the `wheel` group as described by the `su`(1) `man` page (this applies to all Sun systems).

D `secure` setting — The default Sun386i `/etc/ttytab` file considers only the console to be "secure." On other Sun systems, `ttytab` lists all potential login devices as "secure," thus enabling root login through modems and local and remote terminals.

Reference: Sun386i Advanced Administration (February 1989 edition, Chapter 2)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `ttytab`(5), `su`(1)

Root Privileges in SNAP and Organizer
To simplify network-wide security issues, root can use SNAP only for backing up and restoring data. The root account cannot use SNAP to administer users, groups, printers, terminals, modems, or systems. To use SNAP for these tasks, log into an account which is a member of the appropriate administrative group (`networks`, `accounts`, `operator`, `devices`). For more information on these groups, see Sun386i SNAP Administration.

✓ `root` use of Organizer™ — On systems running Sun386i SunOS 4.0.1, `root` cannot run the Organizer. This is fixed in Sun386i SunOS 4.0.2.

Workaround for 4.0.1 — Use the `-ROOT` option to run Organizer as `root`.

Reference: Sun386i SNAP Administration (June 1989 edition, Chapter 1)

3.3 Controlling lockscreen and screenblank

★ automatic screen saver — If you don't do any work on a Sun386i system for 30 minutes, by default the `lockscreen`(1) program automatically darkens the screen and displays a moving image of the Sun logo. The automatic screen saver runs when you are logged into the system, as well as when you have logged out.

You might want to change the default screen-saver program from `lockscreen`(1) to `screenblank`(1), since `screenblank` repaints the screen much faster than `lockscreen` does. (One drawback to `screenblank` is that it does not display any moving images, so it could appear that the system is turned off.) Alternatively, you might not want either screen-saver program to start automatically running after a certain period of inactivity.

Using screenblank Instead of lockscreen
1. Become superuser on the Sun386i system on which you're changing the screen-saver program.
2. Edit `/etc/ttytab`, removing the `-l` switch from the console entry so that it looks like:
```
console "/usr/etc/getty -n -s std.9600" sun on secure
```
3. Reboot the system to activate the change.

Disabling an Automatic Screen Saver
1. Become superuser on the Sun386i system on which you're disabling the automatic running of either `lockscreen` or `screenblank`.
2. Edit `/etc/ttytab`, removing the `-sl` switch from the console entry so that it looks like:
```
console "/usr/etc/getty -n std.9600" sun on secure
```
3. Reboot the system to activate the change.

3.4 Creating and Maintaining User and Group Accounts

Sun386i YP files — Sun386i systems use several additional YP files that other Sun systems lack: `/etc/yppasswd`, `/etc/ypgroup`, and `/etc/ypaliases`. These files use the same format, are edited in the same way, and produce the same YP maps (with some additions) as the `/etc/passwd`, `/etc/group`, and `/etc/aliases` files on any Sun YP master server. The `yp*` versions of these files enable the separation of local and network-wide administration of logins and aliases on the YP master server, and make changing the YP master easier. (On other Sun systems acting as a master server, it is difficult to set up local logins or aliases because the YP makefile uses the "local" `passwd`, `group`, and `aliases` files to create its domain-wide YP maps).

SNAP groups — Every group created with SNAP is similar to a user, with its own name, home directory (`home_server:/files/home/groupname/groupname`), and entry in the `/etc/yppasswd`, `/etc/ypgroup`, and `/etc/ypaliases`, and `/etc/auto.home` files and their associated maps. Differences between users and groups are that you cannot log in with a group name (the password fails), and a group home directory contains only the `copy_home` script and the default files that are copied to the home directories of users as they are added to the group.

Each user account, whether created with the New User Accounts feature of `logintool`, SNAP, or manually by editing the appropriate files, is associated with one primary group, and optionally, with one or more secondary groups (up to a maximum of 16 on systems running SunOS 4.0 or later). A primary group provides defaults for new user accounts, including membership in secondary groups and default `.*` files (such as `.login` and `.cshrc`). Both primary and secondary group membership can govern users' application permissions, file permissions, and the ability to perform tasks using SNAP. See Sun386i SNAP Administration for a more detailed description of group accounts and how to create and edit them.

An advantage to using SNAP or New User Accounts to create user and group accounts is that you do not have to login as root on both the system where the home directory will reside and on the master YP server. Therefore, you do not need access to the root password for these systems.

Home directories — SNAP and New User Accounts use a path name containing an automount mount point to specify home directory location. The home directory path for Sun386i system users as set up by SNAP and New User Accounts in the YP `passwd` map is `/home/username`, because this is where home directories are automounted.

You cannot create home directories in `/usr` on Sun386i systems since `/usr` is more than 100 percent full.

For details on the automounter and hints on maintaining home directories on networks containing Sun386i and Sun-3, Sun-4, or SPARCstation systems (which use a different convention for specifying home directories), see Chapter 9.

Groups — The default primary group assigned to user accounts created with New User Accounts or SNAP is `users`. (However, you can use SNAP to select a different primary group when creating or changing an account.)

Each user account on a network that includes systems running SunOS 3.x can belong to a maximum of eight groups.

Defaults — New users get default environment files such as `.cshrc` and `.login` only from `/home/groupname/defaults/*`, their primary group's home defaults directory. However, you can modify these default files.

- UID/GID allocation — GID and UID allocation in SNAP and New User Accounts is automatic. Although within a YP domain you can control the range of UID and GID numbers allocated by SNAP and New User Accounts (use the `ugid_alloc.range` file available with Sun386i SunOS 4.0.2 — see Chapter 10), you cannot allocate a specific UID or GID with these programs.

- Group entries — If you have manually administered the group files, make sure valid entries for all groups exist in the `passwd`, `group`, and `auto.home` YP maps by using the `ypcat`(1) command. Entries must be consistent among the three maps for SNAP compatibility. SNAP permits the addition of new users to a primary group only if that group entry exists in these files and their associated YP maps. If a group entry exists in `/etc/yppasswd` and `/etc/ypgroup`, but not in `/etc/auto.home`, you can still use SNAP to administer users who have this group as their primary group; however, you cannot add any users to this group.

- Existing user and group accounts —You can use SNAP to change passwords and group membership of accounts created on a Sun-3, Sun-4, or SPARCstation system only if you have manually created the accounts following the rules in Sun386i Advanced Administration.

  Aliases for secondary groups — In Sun386i SunOS 4.0.1 and 4.0.2, when SNAP adds a user as a secondary member of a group, it does not add the user to that group's mail alias. (SDR 5622)

Reference: Chapter 9 ("Inside the Sun386i File System," "Inside the Automounter," "Adding a User Through SNAP," "Adding a User via New User Accounts," "Adding a Group Through SNAP" sections)
System & Network Administration (Chapter 14)
Sun386i Advanced Administration (February 1989 edition, Chapter 3)
Sun386i SNAP Administration (June 1989 edition, Chapter 5)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `auto.home`(5), `yppasswd`(5), `ypgroup` (5), `ypaliases` (5)

3.5 Disabling or Re-enabling a User Account

The easiest way to disable or re-enable a user account is through SNAP. If you must disable an account using manual methods, there are some SNAP compatibility issues you should know about.

- SNAP-disabled accounts — When an account is disabled by SNAP, the login-shell field of the `passwd` map is changed from `/bin/csh` to `/usr/etc/sorry`, which displays an `account disabled` message and will also display the contents of the `.sorry` file in the user's home directory, if that file exists. Thus, the login shell is disabled but the user's password is not disabled.

Manually disabled accounts — To alter an account manually so that SNAP recognizes it as disabled, change the user's login shell in `/etc/yppasswd` to `/usr/etc/sorry`. Do not follow the SunOS convention of adding an asterisk (*) to the password field. Here is an example of an account that is disabled in the `passwd` map:

```
dwu:dDTITpJWA4DWM:101:101:Dr. Wu:/home/dwu:/usr/etc/sorry
```

If there is an asterisk in the password field, then SNAP will not be able to change a user's password because the encryption of the password entered by the user will not match the encrypted password in `/etc/yppasswd`. The `passwd`(1) and `login`(1) commands work similarly in this respect.

If you re-enable the user account through SNAP, the user will be given the default login shell for his or her primary group (this is normally `/bin/csh`).

Reference: Chapter 10 of this manual (`yppasswd` description)
Sun386i Advanced Administration (February 1989 edition, Chapter 3)

3.6 Changing Passwords

You can change user passwords on Sun386i systems the same as on other Sun systems, by issuing the `passwd`(1) command on the local system.

`yppasswd` file — Sun386i YP masters use the file `/etc/yppasswd` instead of `/etc/passwd` to keep track of all passwords on the network. The presence of the `yppasswd` file lets you keep a local user account, with its password in `/etc/passwd`, on the YP master. Thus, the YP master can be more like any other system on the network, with YP files separate from local files. These `yp*` files also make it easier to switch YP masters, as described in Sun386i Advanced Administration.

The YP `passwd` map is based on the `/etc/yppasswd` file, so when you edit `/etc/yppasswd` on Sun386i systems, the `passwd` map is updated when you remake the YP maps. This is the same YP map used on Sun-3/4, and SPARCstation systems.

Changing passwords — When you use `passwd`(1) to change a password on Sun386i systems, the `passwd` command looks in the local `/etc/passwd` file and changes the password there, if it finds it. If `/etc/passwd` does not contain an entry for you, but the file does contain the line `+::0:0:::`, then `passwd` invokes `yppasswd`(1), which looks for your entry in `/etc/yppasswd` on the YP master and changes the entry if it is there. The YP password daemon, `yppasswdd`(8C), then updates the `passwd` YP map.

(On Sun-3, Sun-4, and SPARCstation systems:

- `passwd` does not invoke `yppasswd`; to change YP password information on systems other than Sun386i systems, you must enter the `yppasswd` command.
- The file used to make the YP `passwd` map is `/etc/passwd` on Sun-3, Sun-4, and SPARCstation YP masters as opposed to the `/etc/yppasswd` file on Sun386i YP masters.)

`passwd` map — SNAP never uses the local `/etc/passwd` file, only the YP `passwd` map.

publickey.byname map — If a user has an entry in the `publickey.byname` YP map for use with Secure RPC, and if his password cannot successfully decrypt the private key, he will get decryption messages when logging in.

Reference: Chapter 10 of this manual (`yppasswd` description)
Sun386i Advanced Administration (February 1989 edition, Chapters 2, 8)
System & Network Administration (Chapter 14)

3.7 Transferring Sun386i User Defaults

New user accounts created with SNAP or the New User Accounts feature on Sun386i systems are given a standard collection of default files designed to control and complement the Sun386i Desktop environment. When desirable, users moving from another Sun system to a Sun386i system can also update their environments to reflect these new defaults, using the following procedure:

1. Log in to the Sun386i system using your own name.
2. Update your default files (`.cshrc`, `.defaults`, `.login`, `.mailrc`, `.orgrc`, and `.sunview`) by typing **`update_defaults`**

   Follow the prompts and instructions that appear on the screen.

   If the directory `/home/users/defaults` is not available to your system or you lack the permissions to access it, you'll be prompted for an alternate directory. Type the following directory name as an alternate:

   `/files/home/users/users/defaults`

   (If you are on a diskless system, this alternate directory probably won't be available to your system; you'll need to temporarily log in to a Sun386i system that has a disk and rerun `update_defaults`.)
3. Edit new default files, if necessary:

   If a user previously had custom versions of any of these default files (such as settings for the `path` in the `.cshrc` or `.login` file), the changes must be reapplied by editing the new versions that you have copied.

   The previous versions of the files, which you may want to use for comparison, are stored under `~/filename.old`.
4. Set up a standard mail folder directory by entering:

   ```
   mkdir ~/mail
   chmod 755 ~/mail
   cd ~/mail
   touch dead.letter old_mail personal
   touch inbox outbox trash
   ```
5. Check mail delivery.

   Non-Sun386i YP master — If the YP master is a Sun-3, Sun-4, or SPARCstation system, then mail is automatically delivered to the spool directory.

   Sun386i YP master — If the YP master is a Sun386i system, then check the `policies` map to see if mail is being delivered to the spool directory or to your home directory, using the command:

   ```
   ypcat -k policies | grep mail_delivery
   ```

   Mail to spool directory — If the mail is to be delivered to your spool directory, then comment out the following line (by preceding it with a #) in your `.login` file:

   ```
   setenv MAIL ~/mail/inbox
   ```

   Mail to home directory — If mail is to be delivered to your home directory (a feature available on Sun386i networks), leave the line intact since it is correctly set.

   "Setting Up and Administering Mail" in Chapter 5 provides more information about mail delivery.

If a User Frequently Switches Machines

Users who frequently switch between non-Sun386i and Sun386i systems may wish to edit their `~/.sunview` files to avoid starting Organizer, Help Viewer, or any other applications not currently available on Sun-3, Sun-4, or SPARCstation systems.

Also, since Sun386i systems are based on SunOS 4.0, you might want to run SunOS 4.x on Sun-3, Sun-4, and SPARCstation systems that you frequently access, so that the environments of machines more closely resemble each other.

Alternately, you could use symbolic links that point into some architecture-specific directories on the various machines you use, so that you could have different environments on different machines.

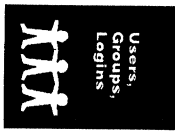
Minimum Settings

`sunview/walking_menus` — Make sure that the Defaults Editor value for the `sunview/walking_menus` entry is set to `enabled` for this user. (The Sun386i root menu will not come up if this entry is incorrectly set.)

If you cannot update your defaults, you should consider placing the following lines in your `.login` file:

`setenv DOS_CMDTOOL off`	Prevents DOS from starting if you simply make a typo from SunOS on Sun386i systems.
`setenv AUTOMOUNT_FIXNAMES true`	Displays automounted directory names starting with the automount directory name (for example, `/home/fred` is displayed on Sun386i systems instead of `/tmp_mnt/home/fred`).

Notes



Chapter 4:
Establishing and Maintaining a System

Legend

- Special features are implemented on Sun386i systems
- D Sun386i default settings are different from Sun-3 or Sun-4 SunOS 4.0 systems
- Explicit steps required to retain SNAP compatibility or other ease-of-use features
- Performing this task may prevent subsequent use of SNAP or ease-of-use features
- Sun386i software cluster must be loaded from tape or diskette
- Some bugs or missing functionality in release 4.0.2
- ✓ Sun386i SunOS 4.0.1 bugs are fixed in 4.0.2

4.1 Changing System Names

You can change the name of a Sun386i diskful or diskless system using similar procedures as for a Sun-3, Sun-4, or SPARCstation system. These procedures are documented in Sun386i SNAP Administration. In addition to the steps in that manual, to change the name of a Sun-3/4 or SPARCstation system, or a Sun386i system that has configuration probing disabled (see "Sun386i Probing" in Chapter 5), you must also:

1. Change the name of the system in the local `/etc/hosts` file.
2. Change the name of the system in the `/etc/rc.boot` file on Sun-3/4 or SPARCstation systems, or in the `/etc/net.conf` file on Sun386i systems.

When you rename any Sun system that's part of a network, you should also check the `/etc/sm` and `/etc/sm.bak` directories on other machines on the network. If you find the old system names in either directory, consider replacing the old name with the new one. If you don't, those machines will display warning messages from the lock manager in their console windows. This is because the lock manager frequently attempts to re-establish connections using the old system name. If you don't replace the old name with the new one in `/etc/sm` or `/etc/sm.bak`, then you can just ignore the messages.

Host and domain names — On Sun386i systems with configuration probing disabled, both the host name and domain name are set in `/etc/net.conf`. On Sun386i systems where configuration probing is enabled, the host name and domain name are determined by responses from a slave server on the network. See Chapter 10 for more information on `/etc/net.conf`.

By comparison, on Sun-3, Sun-4, and SPARCstation systems, the host name is set in `/etc/rc.boot` and the domain name is set in `/etc/rc.local`.

System name changes — SNAP does not provide the ability to change system names, but will support administration of any system with a new name if you follow the procedures in Chapter 8 of Sun386i SNAP Administration (June 1989 edition).

Reference: Chapter 10 of this manual (`net.conf` and `hosts` descriptions)
Sun386i SNAP Administration (June 1989 edition, Chapter 8)
System & Network Administration (Chapter 12)

4.2 Enabling and Disabling Kernel Features

The Sun386i kernel is preconfigured on the Sun386i system disk. The Sun386i kernel includes all of the required portions of the SunOS kernel, as well as the optional features most often used. The optional features not included in the Sun386i preconfigured kernel are those that are commented out in the `SDST386` file (for diskful systems) and the `DL386` file (for diskless systems), both in the `/sys/sun386/conf` directory.

You can add or disable features listed in either the `SDST386` or `DL386` file by reconfiguring the kernel. Reconfiguring a Sun386i kernel is similar to reconfiguring any other Sun kernel — you modify a copy of one of the kernel files provided in `/sys/sun386/conf`. To reconfigure a Sun386i kernel:

1. Load the optional `base_devel` and `config` clusters, included with the Sun386i SunOS Developer's Toolkit software.
2. Follow the instructions in the `README` file in the directory `/sys/sun386/conf`.

Rebuilding Kernels on Sun386i Diskless Systems

Because the diskless client mounts the server's `/export/cluster/sun386.sunos4.0.2` directory as read-only, you cannot build a kernel on a diskless Sun386i system. However, if a diskless client has a Sun386i system as its file server, you can build the diskless client's kernel on the file server.

If a diskless client does not have a Sun386i system as its file server or you want to rebuild the kernel on the diskless machine, you can use the following procedure to build the client's kernel. You also can use this procedure if the file server is a Sun386i system.

Performing this procedure gives the diskless client the ability to write in the file server's `/export` partition.

1. Log in as `root` on the file server.
2. Edit the `/etc/exports` file and change the `/export/cluster/sun386.sunos4.0.2` line to:

   `/export/cluster/sun386.sunos4.0.2 -root=client,access=otherclients`

   where client is the diskless client, and otherclients are the other diskless clients with the same server.
3. Save the newly edited `/etc/exports` file.
4. Enter **`exportfs -av`**
5. Log in as `root` on the diskless client.
6. Edit the diskless clients's `/etc/fstab` file, changing mount permissions for `/usr/cluster` from read-only to read-write.
7. Save the newly edited `/etc/fstab` file.
8. Enter **`umount /usr/cluster; mount /usr/cluster`**
9. Now follow the steps for building kernels in the `README` file in `/sys/sun386/conf`, using the `DL386` file as a guide instead of the `GENERIC` file shown in the directions.
10. After the diskless client has rebooted, edit the diskless client's `/etc/fstab` file, changing mount permissions for `/usr/cluster` from read-write back to read-only.
11. Enter **`umount /usr/cluster; mount /usr/cluster`**
12. On the server, edit `/etc/exports`, removing the `-root=client` section of the `/export/cluster/sun386.sunos4.0.2` line.
13. Enter **`exportfs -av`**

Kernel sizes — The Sun386i kernel is preconfigured. You may notice that a rebuilt kernel is larger than the standard `vmunix` file shipped with the system — even if you build your kernel with only the "standard" options. This is normal.

The size difference occurs because the `vmunix` file built as part of the standard release can safely exclude all code not needed in the preconfigured kernel, whereas the object files provided in the `config` cluster must contain code to accommodate every kernel option.

`config`, `base_devel` clusters — Before you can rebuild the kernel, the `config` and `base_devel` clusters, both included with the Sun386i SunOS Developer's Toolkit software, must be loaded.

`README` file corrections — In Sun386i SunOS 4.0.1, the `README` file in the directory `/sys/sun386/conf` referred to the Sun System Manager's Guide for further details. The file now contains the correct reference, to System & Network Administration.

The Sun386i SunOS 4.0.1 `README` file also omitted `sun386` as a value for machine type. This is corrected in Sun386i SunOS 4.0.2.

Kernel rebuilds on diskless clients — You cannot rebuild a kernel on a diskless Sun386i client because a diskless client mounts its server's directory as read-only. However, if a diskless client has a Sun386i file server, you can rebuild its kernel on the server. If the client's file server is a Sun-3/4, you can rebuild the kernel by performing the steps earlier in this section. (SDR 4573)

Reference: System & Network Administration (Chapter 9)
On-line SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `config`(8)

4.3 Installing Drivers

Sun386i systems support two categories of device drivers: traditional device drivers that you add to a system and then rebuild the kernel to use, and drivers that you can load dynamically into the kernel at any time. Device drivers that are not linked into the kernel are called loadable drivers. You don't have to rebuild and reboot the kernel to add loadable drivers to the system; instead, just use the `modload`(8) command, included in the Sun386i core system, to load the driver into a running system.

You also can convert existing nonloadable drivers to loadable drivers by following the directions in the Writing Device Drivers manual. If you are not sure whether or not a driver is loadable, see if the driver code begins with a "wrapper" module, also described in the Writing Device Drivers manual.

`config`, `base_devel` clusters — If you need to rebuild the kernel (because you are not installing a loadable driver), the `config` and `base_devel` clusters, both included with the Sun386i SunOS Developer's Toolkit software, must be loaded.

✓ **README file** — The Sun386i SunOS 4.0.1 kernel `README` file referred to the Sun System Manager's Guide for further details. The file now contains the correct reference, to System and Network Administration.

Reference: "Enabling and Disabling Kernel Features" earlier in this chapter
Sun386i Developer's Guide (Chapter 8)
Writing Device Drivers (Chapter 5)
System & Network Administration (Chapters 9, 16)
On-line SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `config`(8), `modload`(8)

4.4 C2 Security

Sun386i systems fully support C2 security. However, this feature is not enabled in the preinstalled kernel shipped with each system, so you must rebuild the kernel to get C2 security. For instructions on how to rebuild the kernel, see "Enabling and Disabling Kernel Features" earlier in this chapter.

If you enable C2 security, do not subsequently add new users via New User Accounts or SNAP, because these programs do not support the split password and group files required for C2 operation (`passwd.adjunct` and `group.adjunct`). "Controlling Automatic Account Creation" in Chapter 3 describes how to disable account creation through SNAP and New User Accounts.

4.5 Sun386i Boot Messages

Boot-time messages are generated by the kernel and by `/etc/rc*` scripts. Sun-3, Sun-4, and SPARCstation systems always display these messages as they are booting, while Sun386i systems let you decide whether or not boot messages should be displayed.

Progress meter replaces messages — By default, Sun386i systems boot without displaying the standard SunOS boot messages; instead, a progress meter dynamically displays the status of the boot procedure.

Enabling Boot Messages

Complete the following steps to enable display of the standard messages.

1. Enter the PROM monitor by pressing L1-a. You can do this without logging out, since enabling boot messages is a low-risk operation.
2. At the > prompt, type:

   ```
   > q 494 1
   > c
   ```

3. Redisplay the screen, if you are in SunView™.
4. As superuser, edit the `/etc/net.conf` file on the local machine, changing the `VERBOSE=` setting to:

   ```
   VERBOSE=yes
   ```

These new settings will remain in effect through reboots and power cycles.

Note that the `net.conf` change is not strictly necessary if configuration probing is enabled, because this file is updated automatically at boot time. However, for the sake of simplicity it is usually best to change both settings as shown in the above steps.

No messages — On Sun386i systems, PROM location 494 is set to 0 and `VERBOSE=no`; no messages appear.

Reference: Chapter 5 of this manual ("Sun386i Probing" section)
Chapter 9 of this manual ("Booting a Network Client" section)
Chapter 10 of this manual (`net.conf` description)
PROM User's Manual (Chapter 18)

Kernel Messages

Improved kernel error messages — The Sun386i system logging daemon, `syslogd`, translates some of the cryptic error and warning messages generated by the kernel into solution-oriented, nontechnical messages. For example:

- `Ethernet jammed` is displayed as `Network traffic overflow`
- `no carrier` is displayed as `Problem with network connection. Check cables and transceivers.`

On Sun386i systems, `syslogd` uses the `/etc/In` and `/etc/Out` files when translating messages. When it intercepts a string, `syslogd` looks for that string in the `/etc/In` file. If the string is there, `syslogd` uses an index to find the string's translation in `/etc/Out`, and then generates the translated message. If the intercepted string is not in `/etc/In`, or if either `/etc/In` or `/etc/Out` is not found, `syslogd` generates the standard, unaltered message.

To rewrite kernel messages yourself, see the Sun386i Developer's Guide.

To disable translation, delete or rename `/etc/In` or `/etc/Out`.

Reference: Sun386i Developer's Guide (Chapter 6)

4.6 Four Megabyte Processes and Stack Limits

Many Sun386i utilities in `/etc`, `/usr/etc`, `/usr/bin`, and `/usr/ucb` are 4-Mbyte processes. This means that these processes have a combined limit of 4 Mbytes for text, data, and stack size. This limit helps improve system performance, particularly on 4-Mbyte systems, because 4-Mbyte processes use only one page table for mapping their address space. (On Sun-3, Sun-4, and SPARCstation systems, 4-Mbyte processes are not implemented.)

Because a child process inherits its stack limit from its parent, if you change the stack limit of a shell such that there is no room to map text, data, bss, and shared libraries, the 4 Mbyte process will be unable to run from that shell. When this happens, you will see the following message:

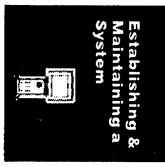
```
crt0: mmap of ld.so failed.
```

To avoid this problem with a shell set to a stack size limit that is too large, use the `unset4`(8) command to change any such program you need to run, so that the program is no longer a 4-Mbyte process. For example, to change `uucp`, become superuser and enter the following commands:

1. **`mount -o remount,rw /usr`**
2. **`unset4 /usr/bin/uucp /usr/lib/uucp/*`**

To determine if a process has a 4-Mbyte limit, issue the `/usr/etc/check4` command. If it does not have a 4-Mbyte process limit, `check4` displays the message `process is not a 4MB process`. If the process is a 4-Mbyte process, then `check4` merely redisplays the prompt.

Reference: On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `check4`(8), `set4`(8), `unset4`(8)



Notes

Chapter 5:
Establishing and Maintaining a Network

continued on the next page

Legend

- Special features are implemented on Sun386i systems
- Sun386i default settings are different from Sun-3 or Sun-4 SunOS 4.0 systems
- Explicit steps required to retain SNAP compatibility or other ease-of-use features
- Performing this task may prevent subsequent use of SNAP or ease-of-use features
- Sun386i software cluster must be loaded from tape or diskette
- Some bugs or missing functionality in release 4.0.2
- ✓ Sun386i SunOS 4.0.1 bugs are fixed in 4.0.2

5.1 Restricted Networks

A Sun386i system, when connected to a network to which it is not yet known, can probe for its IP address using DRARP. The response that servers on the network make to such a request can be controlled by restricting the network. When a network is restricted, Sun386i systems will not be able to automatically install themselves.

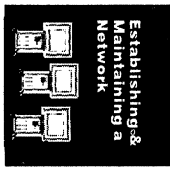
For more information on how this works, see Chapter 9.

Customer sites can restrict the network so that new Sun386i systems will not automatically join a YP domain via Automatic System Installation. On a network running YP:

1. Log onto the YP master server of the domain and become root.
2. Edit the `/etc/policies` file. Setting `pnp` to **`restricted`** (this restricts the network, so that new Sun386i systems cannot use ASI to join the YP domain). If the YP master is a non-Sun386i system, see Chapter 7 for information about adding the `policies` map.
3. Enter **`cd /var/yp; make`**

Automatic System Installation and multiple domains —Special rules apply when using ASI on networks with multiple domains. The `ip_address_allocation` policy should be set to `drarp_only` on only one domain on a network. See "Setting Up Additional Sun386i YP Domains" in Chapter 8 for details.

Automatic System Installation enabled by default — Automatic System Installation is enabled in the `policies` map on Sun386i systems set up as YP master servers; therefore Sun386i networks are unrestricted by default.



What Users See on a Restricted Network

On a restricted network, if a user attempts to add a system by plugging in and powering up a new system, he or she will see the following message:

```
This Network is Restricted
```

The user is also directed to contact the network administrator, with the Ethernet® address of the workstation, to have the proper files set up (either using SNAP or manual procedures).

SNAP runs on both restricted and unrestricted networks. If you add systems manually instead of using SNAP, use care to ensure that all the files are updated properly if you subsequently want to administer those systems with SNAP. This is because SNAP cannot always deal with inconsistencies in these files.

Reference: Chapter 9 of this manual ("Booting a Network Client" and other related sections)
Sun386i SNAP Administration (June 1989 edition, Chapter 8)

5.2 Sun386i Probing

Sun386i systems can probe the network every time they boot to obtain information about the environment. The various types of probing (in the order they are performed), are:

- Network probing
- IP address probing
- Configuration probing

Reference: Chapter 7 of this manual ("Adding New Sun386i YP Files and Maps" section)
Chapter 8 of this manual ("About Configuration Probing in Multiple Domains" section)
Chapter 9 of this manual ("Booting a Network Client" section)
Chapter 10 of this manual (`net.conf` and `policies` descriptions)
Sun386i Advanced Administration (February 1989 edition, Chapters 1, 2)

Network Probing

To determine if the network is connected, a Version 2 Ethernet packet is sent to the Loopback Assistant, using a multicast address. The system determines that the network is connected if there are no transmission errors. No response to this packet by another system is required (if another system responds, the response will be ignored).

Network probing is used to initially determine if the system should be set up as a standalone system. On subsequent boots it detects changes in network role (from standalone to networked) or reports errors when a networked system is no longer connected to the Ethernet.

All Sun386i systems perform network probing the first time they boot (when the system is new or after `unconfigure`(8) has been run).

On subsequent boots, Sun386i diskless systems always perform network probing, while all other Sun386i systems only probe the network if `PNP=yes` in the `/etc/net.conf` file.

Ethernet packet function code — The function code of the Ethernet packet sent in network probing is a zero (0), which can cause error replies from a Loopback Assistant.

IP Address Probing

To determine the proper IP address to use for a given machine's Ethernet address, a Sun386i can broadcast a DRARP request for its IP address (all other Sun systems use the RARP protocol, DRARP is an extension of RARP). The IP address is returned by a YP server running `rarpd` (8). The server returns information which is found either in the ARP cache on the server system, or the `ethers` and `hosts` maps. In the event that the IP address information is not found, a new IP address can be allocated automatically and returned.

A Sun386i diskless system broadcasts a request for its IP address every time it boots. A Sun386i diskless client initially uses RARP to resolve its IP address. If it receives no response to the RARP request, the Sun386i diskless system alternates DRARP and RARP requests until a server replies with the IP address.

All other Sun386i systems use IP address probing the first time they boot (when the system is new or after `unconfigure`(8) has been run). On subsequent boots, Sun386i YP servers and standalone systems never use IP address probing, while all other clients only probe for their IP address if `PNP=yes` in the `/etc/net.conf` file.

DRARP packets between networks — During IP address probing, a Sun386i system broadcasts DRARP packets to the network. On large networks connected by Ethernet bridges, these Ethernet broadcast packets are sent to the entire network. This can result in a Sun386i conversing with a distant Sun386i and joining a YP domain which is physically very far removed from the client. The machines appear to have conspired on the network.

If the bridges can be configured so as to filter out DRARP packets (opcode 0x8035), this should be done. Typically, the "conspiring" of Sun386i systems does not occur when routers are used to connect network segments since routers do not forward Ethernet broadcast packets.

Configuration Probing

To detect network reconfiguration (such as changes in host or domain names), Sun386i client systems can broadcast an RPC request to obtain the following information from a server on the network each time they boot:

- Domain name
- Host name
- Network role
- Time zone
- Time

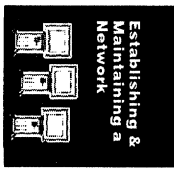
A Sun386i YP server (which runs the pnp daemon, `rpc.pnpd`(8)) answers the configuration request. The server returns information found in the `hosts`, `ethers`, and `systems` YP maps.

Sun386i YP servers and standalone systems never perform configuration probing. Sun386i YP clients and diskless systems installed with the Sun386i diskless server kit do not perform configuration probing the first time they boot. Also by default, `PNP=no` in the `/etc/net.conf` file on those clients and therefore configuration probing is not done on subsequent boots. Conversely, Sun386i network clients and diskless systems installed with ASI or SNAP do perform configuration probing the first time they boot. Also by default, `PNP=yes` in the `/etc/net.conf` file on those clients and therefore configuration probing is done on subsequent boots.

⚠ Booting — A system with configuration probing enabled will not continue to boot, even to single-user mode, until it receives the configuration information.

Multiple domains — If a network has multiple domains that know about the Sun386i client in their YP maps and have Sun386i YP servers in different domains, you must take additional steps to control which domain the Sun386i joins when booting; see Chapter 8 for details.

`net.conf`(5) setting — If configuration probing is not enabled, the `VERBOSE` setting in `/etc/net.conf` is used instead of the NVRAM to enable boot messages from the `rc.*` scripts.



Disabling Configuration Probing

To disable configuration probing on a boot client so that a system gets its configuration information locally, perform the following steps:

1. Log in to the system as root.
2. Create an `/etc/hosts` file containing the name of the local system.

   `ypmatch` hostname **`hosts > /etc/hosts`**

   If single-user mode is often used, also add the addresses of any hosts typically accessed to the `/etc/hosts` file.
3. Copy `/usr/etc/ifconfig` into `/sbin` to configure the network when booting.
4. Edit `/etc/net.conf`, changing `PNP=yes` to `PNP=no`.
5. Reboot this system; it will boot without performing configuration probing.

Because this procedure sets the system up to use `ifconfig` (rather than `netconfig`), it disables network probing (except for diskless clients), the DRARP portion of IP address allocation (again, except for diskless clients), and configuration probing.

Re-enabling Configuration Probing

To re-enable configuration probing on a network, diskless, or diskful client, perform the following steps:

1. Check that the system's Ethernet address is in the YP `ethers` map (on the YP master) by entering:

   `ypcat -k ethers | grep` hostname

   Do not enable configuration probing unless this entry exists.
2. Make sure that there is an available Sun386i boot server in the same domain by entering the command:

   `/usr/etc/rpcinfo -b ypserv 2`

   and checking for names of Sun386i systems. Do not enable configuration probing unless a Sun386i YP master or slave server exists.

3. Become root on the system.
4. Edit `/etc/net.conf`, changing `PNP=no` to `PNP=yes`.
5. Make sure `/sbin/netconfig` exists, and then delete `/sbin/ifconfig`.
6. Delete `/etc/hosts`.

Disabling probing — If you disable configuration probing, the system reconfiguration procedures discussed in Sun386i SNAP Administration will not work.

net.conf(5) — There is no on-line or hard-copy man page describing the `/etc/net.conf` file. (NOSDR)

Single-User Mode and Configuration Probing

If Sun386i boot servers aren't running or the network is down, you can still work locally by doing the following steps:

1. Press L1-a to get to the PROM monitor prompt (>).
2. Boot the system in single-user mode by typing **b -bs**
3. Repair any possible file system inconsistencies by entering **/sbin/fsck**
4. Remount the root file system for root access by entering **mount -o remount,rw /**
5. Clear the mount table by entering **> /etc/mtab**
6. Mount the root and `/usr` file systems by typing **mount -f /; mount /usr**
7. Reset the terminal characteristics by typing **stty dec**
8. Mount all file systems on the disk by entering **mount -at 4.2**
9. Mount all loopback-mounted file systems by entering **mount -at lo**

5.3 checkconfig Program

The `checkconfig` program is similar in concept to `fsck`(8). `checkconfig` verifies that the configuration of a Sun386i system is correct in the YP maps and requests corrections in the YP databases on the YP master, if necessary.

Each time a Sun386i system boots, it automatically runs the `checkconfig` program from the `/etc/rc` script. `checkconfig` checks to see if the system is a YP server, checks if this system has a valid entry in the `systems` map, and checks if this system has a valid entry in the `bootservers` map if this system is a YP server.

Sun386i SunOS 4.0.2 — If the YP master for the system running `checkconfig` is not a Sun386i running SunOS 4.0.2, `checkconfig` fails when trying to update the `systems` and `bootservers` YP maps. This is because only Sun386i SunOS 4.0.2 has the software necessary to perform the update. If the system displays verbose messages, the error:

```
YP updated failed, returned YPerr code 6
```

will be displayed when rebooting.

no **man** page — `checkconfig` does not have a man page on-line or in the SunOS Reference Manual.

YP Server Check

If `checkconfig` determines that the system is a YP server and it does not have a local copy of the YP maps, it copies the maps from the YP master (using `ypsync`(8)) and reboots the system. If `checkconfig` determines that the system is not a YP server but it does have a local copy of the YP maps, it deletes the maps from `/var/yp/`domain_name and reboots the system.

systems Map Check
If `checkconfig` does not find an entry for this system in the YP `systems` map, `checkconfig` contacts the `ypupdated` daemon on the YP master, which makes an entry for this system. If the `systems` map contains an entry for this system but the information is incorrect, `checkconfig` contacts `ypupdated` on the YP master, which corrects the entry.

bootservers Map Check
If this system is a YP server but lacks an entry in the YP `bootservers` map, `checkconfig` contacts the `ypupdated` daemon on the YP master, which makes an entry for this system. If an entry for this system exists in the `bootservers` map but the information is incorrect, `checkconfig` contacts `ypupdated` on the YP master, which corrects the entry.

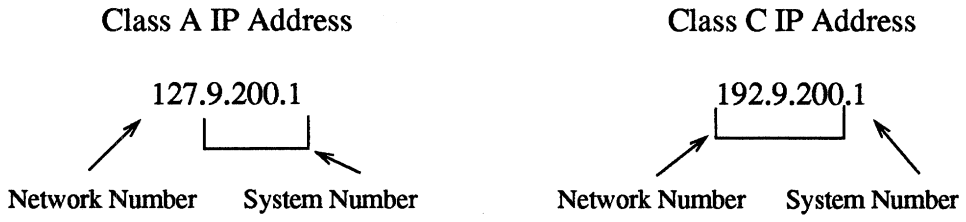
5.4 IP Addresses

The IP (network) address on any Sun system consists of four fields separated by periods, with each field containing a number from 0 through 255. These four fields are grouped into two parts:

- The network number, which is the same for all systems on the network
- The system number, which is unique to this system within the network; the range of available system numbers for a network is determined by the network number

The class of network determines which fields of the IP address form the network number and which fields form the system number.

- Class A networks — The network number is the first field of the address, and the last three fields form the system number. The first bit of a class A address is zero, for example, `127.9.200.1`.
- Class B networks — The first two fields of the address form the network number, and the second two fields form the system number. A first bit of one and a second bit of zero denotes a class B address, for example, `128.9.200.1`.
- Class C networks — The first three fields of the address form the network number, and the fourth field is the system number. Many Sun systems are on class C networks. Class C addresses have a first and second bit of one, for example, `192.9.200.1`.



In addition, some sites split large networks into subnetworks by using part of the system number to designate a subnet number. System & Network Administration describes how to establish subnetworks.

You assign IP addresses to Sun-3, Sun-4, and SPARCstation systems during the `suninstall` process. Sun386i systems can assign default IP addresses automatically during Automatic System Installation (ASI) or via SNAP. However, you should assign a nondefault number if you might later connect this network to another. Contact the Network Information Center (NIC) at 1-800-235-3155 to obtain a unique network number for your site if you do not already have one and plan to join the Internet. (Some sites may already have a range of network numbers from NIC; contact the head of network services at your site to check if this is the case.)

IP address assignment — You can control the assignment of IP addresses for each system allocated by SNAP or Automatic System Installation by creating the `/etc/ipalloc.netrange` file on the master server (see the `ipalloc.netrange`(5) `man` page). Whenever SNAP or Automatic System Installation (ASI) must assign an address, it chooses an address from the pool specified in this file, if it exists.

Range maximum — If you specify more than 40 ranges, the `ipallocd`(8C) daemon ignores additional ranges supplied, without printing a warning.

Reusing a system number — If a system is removed within one hour after being added with SNAP, you cannot reuse its IP address, and therefore its system number, until the one-hour period has expired. If you use SNAP to add a system and try to reuse the system number within that hour, SNAP responds that the system number is already in use.

Reference: Sun386i SNAP Administration (June 1989 edition, Chapter 8)
Sun386i Advanced Administration (February 1989 edition, Chapter 2)
System & Network Administration (Chapter 12)

Changing the IP Address on Sun386i Systems

The steps to change an IP address are similar for all Sun systems, and depend on the circumstances necessitating the change. The table in this section provides steps to:

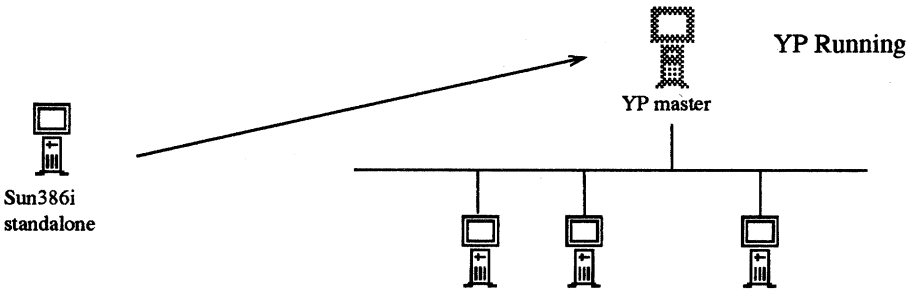
- Change an address of an individual Sun386i system, when moving it to a different network or to another location on the same network
- Change the address of all Sun386i systems (when subnetting or merging two networks, for example)

For information on changing YP domains, see Chapter 8.

The following table is arranged sequentially by network role. If you are moving an individual system, perform only the steps associated with that system, as shown in the left column. Similarly, if you are changing all addresses but the network does not include systems of a particular network role, then skip the steps associated with those systems. For instance, if the network does not have any slave servers, skip steps 4–6.

If Changing IP Address of This Sun386i System	Do These Steps
All Clients	1. As superuser on each client, change the client's address and address of the client's boot server in `/etc/hosts` (clients with configuration probing enabled won't have this file).
	2. Shut down each client by entering **`/etc/halt`**
All Systems	3. Remove the ARP cache entry for the old IP address by performing one of the following three steps: ♦ Leave any system getting a new IP address off for one hour. ♦ As superuser, enter the following on all systems on the network for each client whose address is changing: **`arp -d`** hostname ♦ Reboot all systems on the network.
Slave Servers	4. As superuser on each slave server, replace the old address with the new one in the `/etc/hosts` file.
	5. Delete the `/var/yp/`domain_name directory from each slave server.
	6. Shut down each slave server by entering **`/etc/halt`** and powering down.
All Systems	7. As superuser on the master server, change addresses for each system in `/etc/hosts` and `/etc/networks`. Also update `/etc/netmasks` if the new IP addresses are in a different subnet.
	8. Remake YP maps on the master server (**`cd /var/yp; make`**).
Master Server	9. As superuser, reboot the master server (**`/etc/reboot`**).
Slave Servers	10. Turn on all slave servers.
Diskless Clients	11. Copy the new version of `/etc/hosts` from the master server by entering the following on each boot server: **`rcp`** master_name**`:/etc/hosts /etc`**
	12. Get the hexadecimal equivalents for the new network addresses for each server's diskless clients by entering: **`/usr/etc/install/script/convert_to_hex`** IP_address
	13. On each client's boot server, log in and as superuser update the symbolic links for each diskless client's boot file to reflect the new IP address. In the `/tftpboot` directory, enter: mv old_ip_address.<s386,sun3,sun4> new_ip_address.<s386,sun3,sun4>
All Clients	14. On any NFS server of clients with new IP addresses, re-export the file systems to get the new host information by entering: **`exportfs -a`**
	15. Turn on.

5.5 Upgrading a Standalone System to the Master Server



How you upgrade a system that's been set up as a standalone (not connected to a network) to a YP master depends on whether you:

- Can use the default IP address (`192.9.200.1`) and YP domain name (`YP.noname`); use the default IP address only if this network will not be connected to another one.
- Want to unconfigure the system; Chapter 2 of Sun386i SNAP Administration (June 1989 edition) gives instructions on how to do this.

D YP masters — Sun386i standalone systems are YP masters of a network of one system.

✓ **`unconfigure`** — The Sun386i `unconfigure`(8) command had a bug in Sun386i SunOS 4.0.1 whereby the command deleted all software in `/files/vol`. This is fixed in Sun386i SunOS 4.0.2.

Using Default IP Address and YP Domain Name

1. Plug in the Ethernet transceiver cable.
2. Log in to the workstation as root.
3. Reboot the system.
4. Follow the directions that will appear on the screen.

Using Nondefault IP Address and YP Domain Name

1. Plug in the Ethernet transceiver cable.
2. Log in to the workstation as root.
3. Edit the `/etc/net.conf` file by:
   a. Changing the `NETWORKED=NO` line to `NETWORKED=YES`
   b. Changing the domain name value — `DOMAINNAME=new_domain_name` (`YP.domain_name` is a SunOS 4.0 convention, and is recommended for Sun386i systems)
   c. Setting the `PNP` value to `PNP=YES`, if it is currently set to `NO`
4. Edit the local `/etc/hosts` file on this system. The `/etc/hosts` file will contain one entry:

   ```
   127.0.0.1 hostname localhost loghost mailhost timehost  # Desktop
   ```

   a. Replace the IP address (`127.0.0.1`) with the IP address for this system.
   b. Delete the word `localhost` from the line.

c. Add the following line:

```
127.0.0.1 localhost
```

For example, if the new IP address is `192.9.200.1` and the host name is `spam`, then the lines in `/etc/hosts` would be:

```
192.9.200.1 spam loghost mailhost timehost # Desktop
127.0.0.1 localhost
```

Pay attention to the guidelines in the procedure for "General Steps: Setting Up Multiple Domains" in Chapter 8 because you may need to set some domain policies if this is not the only YP domain on this network. You might also have to change the domain name; see Sun386i Advanced Administration for details.

5. For SNAP to recognize this system as a YP master, add an entry for this network to the local `/etc/networks` file, using the syntax:

   network_name network_number aliases

   For example:

   ```
   the_network  192.9.200  localnet
   ```

6. Display the contents of the `/etc/publickey` file. If it contains the old domain name (`YP.noname`), replace the old name with the new domain name (for example, `YP.`new_domainname).
7. Similarly, edit the `/etc/netgroup` file, replacing all occurrences of the old domain name with the new one.
8. Force the rebuilding of the `netid` YP map, so that services using Secure RPCs can validate superuser on this system:

   ```
   cd /var/yp
   rm netid.time
   ```

9. Remake the YP maps by entering `cd /var/yp; make`
10. Rename the default domain name directory with the new domain name:

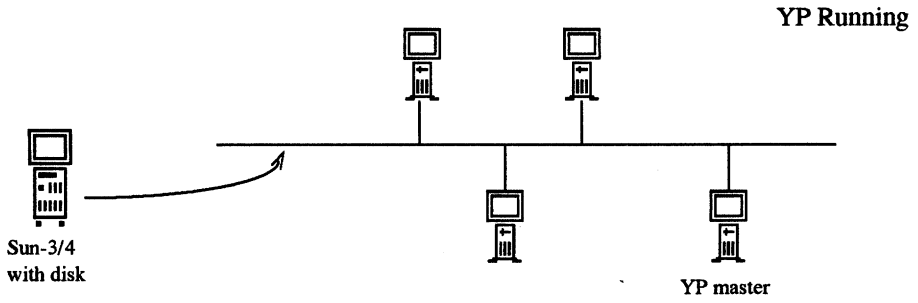
    ```
    cd /var/yp
    mv YP.noname new_domainname
    ```

11. Reboot the system. It will come up as a YP master, using the address you gave it in step 4.

YP master recognition — For SNAP to recognize this system as a YP master, and to be able to add other systems to the network with SNAP, the `/etc/networks` file on this system must contain an entry for this network. See the `networks`(5) `man` page for file format details.

Reference: Chapter 8 of this manual ("General Steps: Setting Up Multiple Domains")
Sun386i Advanced Administration (February 1989 edition, Chapter 2)
Sun386i SNAP Administration (June 1989 edition, Chapters 2, 7, 8)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) – `networks`(5)

5.6 Adding Non-Sun386i Clients to Sun386i Networks



You can use SNAP to add a Sun-3, Sun-4, or SPARCstation system to a Sun386i YP network (a YP network with a Sun386i master server). SNAP automatically adds entries to the same YP maps that it does when you add a Sun386i system with SNAP. You can also add a Sun-3/4 or SPARCstation client to a Sun386i YP domain without using SNAP. Both methods follow.

⚠ Diskless non-Sun386i systems — Because Sun386i systems do not support `suninstall`(8), diskless Sun-3, Sun-4, or SPARCstation systems cannot be served by Sun386i systems.

Adding a Non-Sun386i Client Using SNAP

Sun386i SNAP Administration explains how to use SNAP to add a Sun-3/4 or SPARCstation client with a disk. After adding the client to network files, you must run `suninstall` on the Sun-3/4 or SPARCstation system before it will boot. `suninstall` loads software onto non-Sun386i systems and prompts for configuration information for that system. Regardless of whether or not you use SNAP to add a non-Sun386i client, you must run `suninstall` on the client.

When running `suninstall`, be sure to follow the disk partitioning guidelines in Sun386i SNAP Administration (June 1989 edition, Chapter 8). Also, when prompted for a domain name, specify the Sun386i YP domain name (which, by convention, starts with `YP`).

When `suninstall` finishes, you then must manually set up printers (see Chapter 1), user accounts (see Chapter 3), and mail delivery (detailed later in this chapter). Also, if you want to run the automounter on this system, you must start it (see Chapter 9).

Reference: Chapter 1 of this manual ("Installing Printers" section)
Chapter 3 of this manual ("Creating and Maintaining User and Group Accounts" section)
Chapter 9 of this manual ("Inside the Automounter," "Adding a Network Client Through SNAP" sections)
Sun386i SNAP Administration (June 1989 edition, Chapter 8)
Sun386i Advanced Administration (February 1989 edition, Chapter 1)
Installing the SunOS (Chapters 3, 8)
Sun Software Technical Bulletin (December 1989)

Adding a Non-Sun386i Client without Using SNAP

If you don't want to use SNAP to add a Sun-3/4 or SPARCstation client with a disk, you must:

1. Manually add client information to the `/etc/hosts` file (also to `/etc/systems` and `/etc/ethers` files, if you want to use SNAP to display this system's information) on the YP master.

2. Propagate changes to the YP maps (`cd /var/yp; make`).
3. Run `suninstall` on the client. See the instructions in Installing the SunOS for details.

Reference: Chapter 10 of this manual (`ethers`, `hosts`, and `systems` descriptions)
Installing the SunOS (Chapters 3, 8)
Sun386i SNAP Administration (June 1989 edition, Chapter 8)

5.7 Upgrading a Client to a Slave Server on a Sun386i Network

How you upgrade a client to a slave server on a Sun386i network depends on whether or not you have a user in the `networks` group.

No User in **`networks`** Group

1. Log in to the master YP server as root.
2. Add the client that you're upgrading to a slave server to the `/etc/ypservers` file.
3. (Optional) If you want SNAP to display a system as a slave server, edit the entry for the system in `/etc/systems`, changing its network role to **`slave_bootserver`**.
4. Remake the YP maps by entering **`cd /var/yp; make`**
5. Log out on the master YP server.
6. On the system that you are upgrading to a YP slave server, log in as root.
7. Get a local copy of the YP maps by entering:

   `/usr/etc/yp/ypinit -s` master_server_name
8. Reboot the system. This starts `ypserv` and on Sun386i systems runs `checkconfig`, which adds this system to the `systems` and `bootservers` YP maps.

Sun386i Advanced Administration — Both commands in step 4 on page 24 are missing a required master_server_name argument. The commands should be:

`/usr/etc/yp/ypinit -s` master_server_name (for SunOS 4.0)

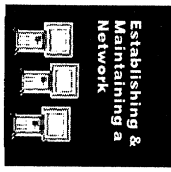
`/etc/yp/ypinit -s` master_server_name (for SunOS 3.x)

User in **`networks`** Group

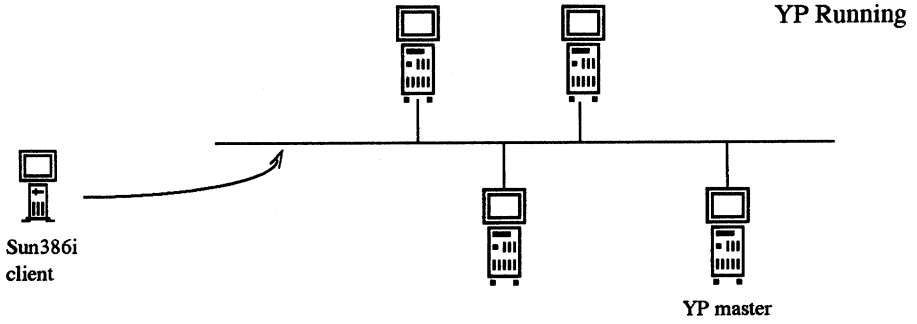
1. On the system you are upgrading to a YP slave server, log in as a user in the `networks` group.
2. Add this system to the `ypservers` YP maps and get a local copy of the YP maps by entering:

   `/usr/etc/yp/ypinit -s`
3. Reboot the system. This starts `ypserv` and runs `checkconfig`, which adds this system to the `systems` and `bootservers` YP maps.

Reference: Sun386i SNAP Administration (June 1989 edition, Chapter 8)
Sun386i Advanced Administration (February 1989 edition, Chapter 2)



5.8 Installing Sun386i Clients on a Network with a Non-Sun386i Master



This section describes adding a Sun386i client, with or without a disk, to a Sun-3/4 or SPARCstation YP network (one that doesn't have a Sun386i master). The following information pertains to all Sun386i clients being added to a non-Sun386i network.

Configuration probing disabled — Configuration probing is automatically disabled when you add a Sun386i system to a Sun-3/4 network through ASI (using option 3, to join as a YP client). Sun386i systems with configuration probing turned off use the `/etc/net.conf` file for some information that is set in the `rc.*` files for other Sun systems. See Chapter 10 for more information on `net.conf`.

Automounter and `ypprintcap` — You must perform additional steps if you want to use the following features, both of which are described in Chapter 7:

- Automatic mounting of NFS directories via the automounter
- Use of `ypprintcap` features on Sun386i systems

Missing features — If you add a Sun386i YP client to a YP domain that has a non-Sun386i system as the YP master, the Sun386i system lacks the following services:

- All SNAP services except backup and restore
- The ability to add new users via New User Accounts (`logintool` works, though)
- Uniformly accessed applications and home directories through `/vol` and `/home`, respectively
- Automatic access to newly added printers from Sun386i systems
- Configuration probing
- Mail delivery to home directories

You can acquire the last four services above if you add Sun386i YP maps to the non-Sun386i YP master, as described in Chapter 7.

SNAP help — When you press the Help key over a grayed-out SNAP category, the message that appears states that YP is not running. This is not necessarily the case. SNAP assumes that YP is not running if it cannot find all of the YP maps that it uses, some of which are specific to Sun386i systems.

Reference: "Sun386i Probing" earlier in this chapter
Chapter 7 of this manual
Chapter 10 of this manual (`net.conf` description)
Sun386i SNAP Administration (June 1989 edition, Chapter 8)
Sun386i Advanced Administration (February 1989 edition, Chapters 3, 4, 9)

Installing a Sun386i Client with a Disk

1. As root on the YP master, edit `/etc/hosts` by:
   a. Adding the IP address and system name.
   b. Adding `timehost` to the list of aliases for a server that is typically available (such as the master YP server) so that when a Sun386i system boots and runs `/etc/rc.local`, it synchronizes its time with this time host. For example:

      ```
      192.9.200.1 master timehost
      ```
2. Remake the YP database by entering **`cd /var/yp; make`**
3. Attach the Sun386i system to the network and turn it on.
4. From the menu that appears, choose:

   3. Join an Existing Yellow Pages Domain as a YP Client.
5. Set up the time zone and network settings, as prompted.

As with Sun-3/4 and SPARCstation systems, you must manually administer Sun386i systems because SNAP is unavailable. Also, all Sun systems running SunOS 4.0 or later always set the subnet mask through the YP `netmasks` map, if it exists and contains a mask for the network. (See Chapter 10 for more information about the `netmasks` map.)

Installing a Sun386i Client without a Disk

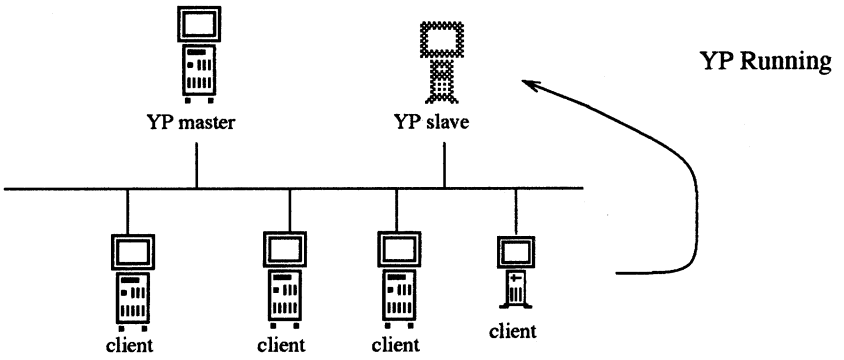
A Sun-3 or Sun-4 system can serve a diskless Sun386i system, provided that the server is running:

- SunOS 4.0 or later
- YP
- The server kit for Sun386i diskless systems (provided with Sun386i SunOS 4.0.1, and with Sun386i SunOS 4.0.2 for new customers)

Server kit — You must install the Sun386i diskless server kit on the Sun-3/4 system. The release notes Installing Sun386i SunOS 4.0.2 provide instructions on diskless server kit installation and use.

Reference: Sun386i Administrator's & Developer's Notes for SunOS 4.0.2
Sun386i Administrator's & Developer's Notes for SunOS 4.0.1
System & Network Administration (Chapter 14)

5.9 Upgrading a Sun386i YP Client to a Slave Server on a Non-Sun386i Network



Sun386i systems can be slave servers on networks that do not have a Sun386i YP master server. In addition to offering the same features as any Sun slave server, Sun386i YP slave servers also can respond to configuration probing requests that Sun386i clients might send. If Sun386i clients have configuration probing enabled, a Sun386i YP server is required to answer client requests for information such as host name and domain name. Configuration probing is disabled automatically on Sun386i clients added to networks with non-Sun386i YP masters via ASI. However, if a Sun386i slave server is on the network, you can re-enable configuration probing on these clients ("Configuration Probing" earlier in this chapter provides details).

A Sun386i YP server verifies its network role and starts all the relevant boot server daemons every time it boots. Every half hour the `ypsync` program (a Sun386i feature) on all Sun386i YP servers automatically verifies that the most up-to-date version of the YP maps is being used.

⚠ Restricted network required — When adding a Sun386i slave to a Sun-3/4 domain, be sure to make the network restricted by setting `pnp` to **`restricted`** in the file `/etc/policies` on the YP master. (You should also set the other `policies` values as shown:

```
newlogin    restricted
ip_address_allocation    none
mail_delivery    spool_area
```

"Adding New Sun386i YP Files and Maps" in Chapter 7 provides details.

If you don't restrict the network, the Sun386i slave server can allocate network resources, such as IP addresses, without recording them in the master copy of the YP maps.

These steps upgrade a Sun386i client to a YP slave server. To add some of the Sun386i network administration features to a non-Sun386i network, see Chapter 7.

1. Log in to the master YP server as root.
2. Add the client that you're upgrading to a slave server to the `ypservers` map on the YP master. To do this, issue the following commands:

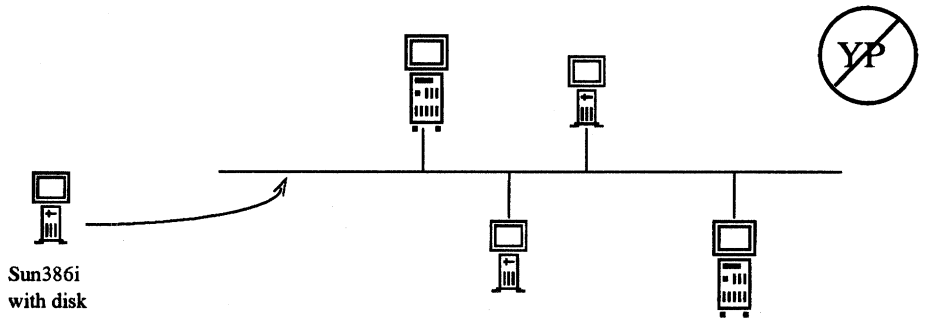
   ```
   cd /var/yp/domainname
   /usr/etc/yp/makedbm -u ypservers > /tmp/ypservers
   echo Sun386i_hostname >> /tmp/ypservers
   /usr/etc/yp/makedbm /tmp/ypservers ypservers
   /usr/etc/yp/yppush ypservers
   ```

   Ignore the message `YP server not registered at` Sun386i_hostname. This indicates that the master was unable to push the YP maps to the slave server. The slave server will retrieve a copy of the maps when you run `ypinit` (step 5 of this procedure).

3. Log out on the master YP server.
4. Log in as root to the Sun386i client that you are upgrading to a slave server.
5. Get a local copy of the YP maps by entering `/usr/etc/yp/ypinit -s`
   If you see the error message `ypsync: You do not have the 'networks' privilege`, this indicates that you did not perform step 2 correctly.
6. Reboot the system.

Reference: Chapter 7 of this manual ("Adding New Sun386i YP Files and Maps")

5.10 Installing Sun386i Systems on Non-YP Networks



You can add a Sun386i system with a disk to a network that is not running YP. However, because the installation menus set up the Sun386i system with YP running, you must decide if you want to continue running YP on the Sun386i system, or if you should turn YP off.

Diskless Sun386i systems — Diskless Sun386i systems cannot be installed on a network that is not running YP.

YP use — Sun386i systems automatically enable YP when they boot.

SNAP requirements — SNAP will not run on Sun386i systems that are on non-YP networks unless you configure a Sun386i system as the master YP server for all Sun386i systems, which changes the network into a Sun386i network running YP.

`/bin/mail` — For Sun386i SunOS 4.0.1, `/bin/mail` required YP to deliver mail. (You could still send mail and read it by NFS mounting `/var/spool/mail` from another system.) `/bin/mail` does not require YP in the Sun386i SunOS 4.0.2 release.

Reference: Sun386i SNAP Administration (June 1989 edition, Chapter 8)
Sun386i Advanced Administration (February 1989 edition, Chapters 3, 9)

Installing a Sun386i System Not Running YP

1. Do not plug the Sun386i system into the network.
2. Set up the new Sun386i system by following the instructions in the "Disabling YP" section of Chapter 6.
3. Select an IP address for this system, and then make sure it is not already in use by checking the contents of `/etc/hosts` on another Sun system that is part of the network that this Sun386i system will join. The network number will be the same as the other systems, but the system number must be different. (See "IP Addresses" earlier in this chapter for an explanation of address components.)

4. As root, edit the local `/etc/hosts` file on this system. The `/etc/hosts` file will contain one entry:

   ```
   127.0.0.1 hostname localhost loghost mailhost timehost  # Desktop
   ```

   a. Replace the IP address (`127.0.0.1`) with the IP address for this system.

   b. Delete the word `localhost` from the line.

   c. Add the following line:

   ```
   127.0.0.1 localhost
   ```

   For example, if the new IP address is `192.9.200.57`, then the lines in `/etc/hosts` would be:

   ```
   192.9.200.57 hostname loghost mailhost timehost # Desktop
   127.0.0.1 localhost
   ```

5. Connect the Sun386i system to the network and reboot the system.
6. Log into the Sun386i system as root.
7. Add any other systems already on the network to the `/etc/hosts` file on the Sun386i system. You can get the required information from `/etc/hosts` on another Sun system on the network.
8. Add any network user accounts to the local `/etc/passwd` and `/etc/group` files on the Sun386i system. You can get this information from the `/etc/passwd` and `/etc/group` files on another Sun system on the network.
9. Check the time and date and reset them if they are incorrect, using the `date`(1) command. You might want to add `cron` jobs to synchronize the time automatically at a given interval (for instance, once a day).

Converting a Non-YP Network into a Sun386i YP Network

If the Sun386i system is to run YP in a network that was not previously running YP, you can set up the Sun386i system as the master YP server. Sun386i SNAP Administration and the "Upgrading a Standalone System to the Master Server" section earlier in this chapter describe two procedures for doing this.

5.11 Setting Up and Administering Mail

Home directory delivery — On Sun386i systems in a YP domain that has a `policies` map with `mail_delivery` set to `home_directory` (the default), mail is delivered to the user's home directory, ensuring that users can read mail from any Sun386i system with the home directory mounted.

Aliases — On Sun386i systems, an alias is automatically set to a user's home directory server for each user when you use SNAP or New User Accounts to create an account. (On non-Sun386i systems, administrators must add an entry to `/etc/aliases` on the YP master and then remake the YP maps.)

As an alternative, you can have mail delivered to the spool directory on Sun386i systems, as on other Sun systems. Mail delivery on a Sun386i system works exactly as on other Sun systems if either:

- YP is not running on the Sun386i system, or
- This system's `mail_delivery` policy in the `policies` map on the YP master is not set or is set to `spool_area`

Reference: Chapter 8 of this manual (for multiple domain issues regarding mail)
Sun386i Advanced Administration (February 1989 edition, Chapter 9)

Sending Mail

Here's what happens when a user delivers mail to another system:

1. If the message is addressed to another system either explicitly (username@system) or implicitly (username, where username is an alias for username@system), the `/bin/mail`(1) program on the local system passes the message to the local `sendmail`(1) program.

   If the message is addressed to someone on the local system (the sender's system is also the addressee's system), then `/bin/mail` delivers the message; `sendmail` is not involved.

   If the message is addressed to a user who is not known by the sending system (the user has neither a login ID nor an alias entry), and no other system is specified, then `/bin/mail` returns the mail to the sender with an error message.

2. The local `sendmail` program routes the message to remote `sendmail` programs until each addressee's system is reached.

3. `sendmail`(1) on each addressee's system passes the mail to `/bin/mail`.

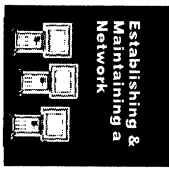
4. On Sun386i systems only, `/bin/mail` checks to see if YP is running.

   If YP is not running, `/bin/mail` automatically delivers the mail to `/var/spool/mail/`username.

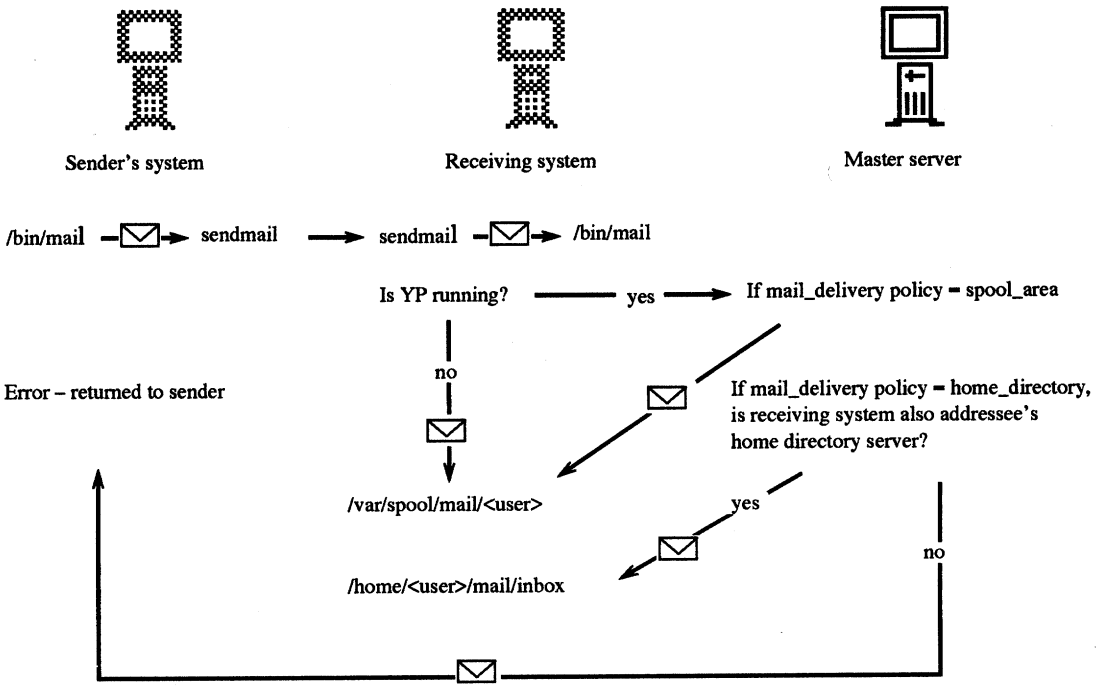
   If YP is running, `/bin/mail` checks the `mail_delivery` policy in the YP `policies` map. If the policy is set to `home_directory`, then `/bin/mail` checks the `auto.home` map to see if the receiving system is the home directory server. If it is, `/bin/mail` delivers the message to `/home/`username`/mail/inbox`. If this machine is not the addressee's home directory server, `/bin/mail` returns the mail to the sender's system with an error.

   If the `mail_delivery` policy is set to `spool_area`, `/bin/mail` delivers the message to `/var/spool/mail/`username on the mail server (the same location as for Sun-3, Sun-4, or SPARCstation mail delivery).

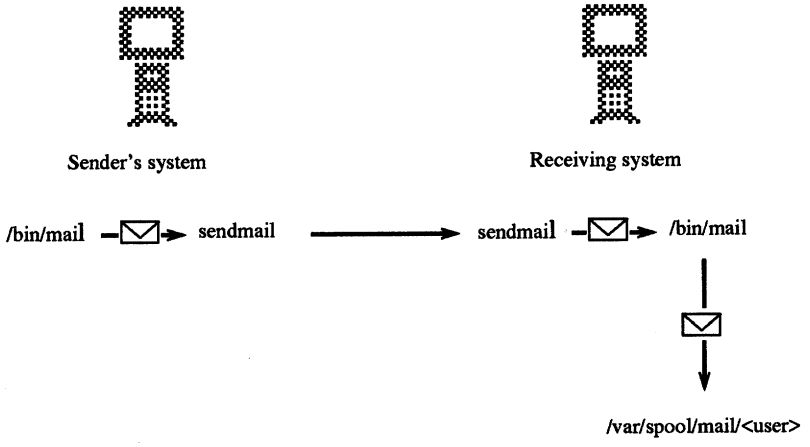
The diagrams on the next page show how mail delivery works on Sun386i and on Sun-3, Sun-4, and SPARCstation systems.



Mail Delivery Between Sun386i Systems



Mail Delivery Between Sun-3, Sun-4, or SPARCstation Systems



Reading Mail

When a user reads mail on Sun386i systems using `mailtool`(1), `/usr/ucb/Mail` determines the location of a user's mailbox by checking the `MAIL` environment variable. This variable is set to `~/mail/inbox` in the default Sun386i `.login` file.

When a user reads mail on a Sun-3, Sun-4, or SPARCstation system, `/usr/ucb/Mail` displays the contents of `/var/spool/mail/`username.

Mail Delivery to a Spool Area

To have mail for all users in a YP domain delivered to a spool area rather than to home directories, or to enable users with accounts created on Sun386i systems to have their mail delivered to a non-Sun386i system instead of to a Sun386i system, perform the following steps:

1. Become superuser on the YP master server and edit the `/etc/policies` file if it exists. Change the line:

   ```
   mail_delivery          home_directory
   ```

   to:

   ```
   mail_delivery          spool_area
   ```

2. Still as superuser on the YP master, include an alias for each Sun386i user in `/etc/ypaliases` if the master is a Sun386i system, or in `/etc/aliases` if the YP master is a Sun-3/4 system. Use the format:

   user:user@mail_server

   where mail_server is the name of the non-Sun386i system that is to receive the mail.

3. Remake the YP policy map by entering the command:

   ```
   cd /var/yp; make
   ```

4. Have each user comment out the following line, if it exists, in his or her `.login` file:

   ```
   #setenv MAIL ~/mail/inbox
   ```

5. Tell users to log out and log back in again so that the `.login` file change takes effect.
6. For each group, edit `/home/groupname/defaults/.login` and comment out this same line. (This will ensure that new users added through SNAP or New User Accounts get a corrected `.login` file.)
7. Now mail will be delivered to the spool area (`/var/spool/mail`) on the machine specified for each user in the `aliases` map (as in username@system), as on Sun-3, Sun-4, and SPARCstation systems. A user can either:

   - Read mail on system
   - Use NFS to mount `/var/spool/mail` from system onto any system from which the user wants to read mail

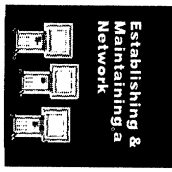
   For example, if the mail server is a Sun386i system, as superuser add the following line to `/etc/exports` on the mail server:

   ```
   /export/var/localhost/spool/mail
   ```

   Then, also as superuser, enter the following line into the `/etc/fstab` file on any system from which users will read mail:

   ```
   mailbox_server:/export/var/localhost/spool/mail /var/spool/mail nfs rw 0 0
   ```

   Be careful not to give root access to clients that mount the mail spool directory; with root access, it is easy for users (or programs such as `unconfigure`) to unwittingly remove or overwrite other users' mailboxes that reside on the mailbox server. When exported as just shown, the default is no root access.



SNAP and New User Accounts — User accounts created by SNAP and New User Accounts (NUA) are set up so that users receive and read mail from their respective home directories. (The `mail_delivery` policy on Sun386i domains is set to `home_directory` and the `MAIL` environment variable in each user's default `.login` file is set to `~/mail/inbox`.) SNAP and New User Accounts also make entries in the YP `aliases` map, creating user@home_server aliases, where home_server is the name of the home directory server (where mail is automatically sent).

non-Sun386i mail file locking — Users whose mail is delivered to their home directories should read their mail only when logged in to a Sun386i system. This is because mail services on Sun-3, Sun-4, and SPARCstation systems only support mail file locking on spool directories.

Delivery policy — SNAP assumes that the mail delivery policy is set to `home_directory`. Therefore, if the mail delivery policy is set to `spool_area`, after creating a user account using SNAP or New User Accounts you must edit `/etc/ypaliases` on the master server. Change username@home_server (the entry that SNAP added to the file when it created the user account) to username@mail_server.

`/bin/mail` — For 4.0.1, `/bin/mail` required YP even when it was not delivering mail to home directories. `/bin/mail` does not require YP in the Sun386i SunOS 4.0.2 release.

5.12 Setting Up and Using UUCP

The same UUCP (UNIX-to-UNIX Copy Program) functionality is available on Sun386i systems as on other Sun systems.

UUCP Process Size Differences

As with many other Sun386i utilities in `/etc`, `/usr/etc`, `/usr/bin`, and `/usr/ucb`, `uucp` is a 4-Mbyte process. (See page 41 for a description of 4-Mbyte processes.)

If you run `uucp` in a shell that has a stack limit of 4 Mbytes (or larger) you'll see the message:

```
crt0: mmap of ld.so failed
```

To avoid this problem in a shell set to a stack size limit that is too large, use the `unset4` command to change `uucp` so that it is no longer a 4-Mbyte process.

Become superuser and enter the following commands:

1. **`mount -o remount,rw /usr`**
2. **`unset4 /usr/bin/uucp /usr/lib/uucp/*`**

To determine if a process has a 4-Mbyte limit, issue the `/usr/etc/check4` command. If it is not a 4-Mbyte process, `check4` displays the message `process is not a 4MB process`. If the process is a 4-Mbyte process, then `check4` merely redisplays the prompt.

`comm` cluster — The `comm` cluster, included with Sun386i Application SunOS software, must be loaded to use `uucp`.

- **`uucp man` page** — The `uucp(1C)` `man` page refers to Installing the SunOS for information on installing optional software. That manual is pertinent only for installing software on Sun-3, Sun-4, or SPARCstation systems.

Reference: System & Network Administration (Chapters 15, 21)
Sun386i Advanced Administration (February 1989 edition, Chapters 4, 9)
On-line Sun386i SunOS 4.0.2 `man` pages (`man_pages` optional cluster must be loaded) — `uucp(1C)`, `check4(8)`, `set4(8)`, `unset4(8)`

5.13 Installing Third-Party Software

Third-party software installation on Sun386i systems is basically the same as on other Sun workstations, except as noted below.

- **`/usr`** — The `/usr` directory (except for `/usr/local`) is reserved for files that are bundled with SunOS. This can be a problem when third-party installation scripts load software into `/usr`, and do not permit customers to override this default location. Because `/usr` is read-only and very full, you cannot store additional software in `/usr` unless you repartition the disk.

  Sun-3, Sun-4, and SPARCstation systems have an installation option whereby you can make `/usr` as large as you like. On Sun386i systems, you can remount `/usr` for write access and create a directory for loopback mounting or create a symbolic link from `/usr` to a partition such as `/files`, which has space available for adding software.

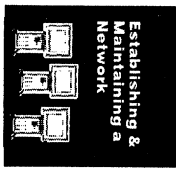
  `/usr/local` — `/usr/local` is usually private to a workstation, though it is shared between a boot server and its diskless or diskful clients. This area is used only by a single architecture and is loopback mounted, typically to `/files/local/sun386`, so that it has as much space as is available on the local system.

  Third-party software — By Sun386i convention, `/usr/local/application` is where you should install third-party software if it's going to be local to the workstation (but shared with diskless and diskful clients). You should install third-party software for network-wide use in `/files/vol/application`, and then make it available through the automounter.

  Shell scripts, binaries — On a Sun386i network, directories under `/vol/local` are the standard location to store shell scripts and binaries for all architectures. `/vol/local` is an automount point. Typically `/files/vol/local` on the master server is automounted on `/vol/local`.

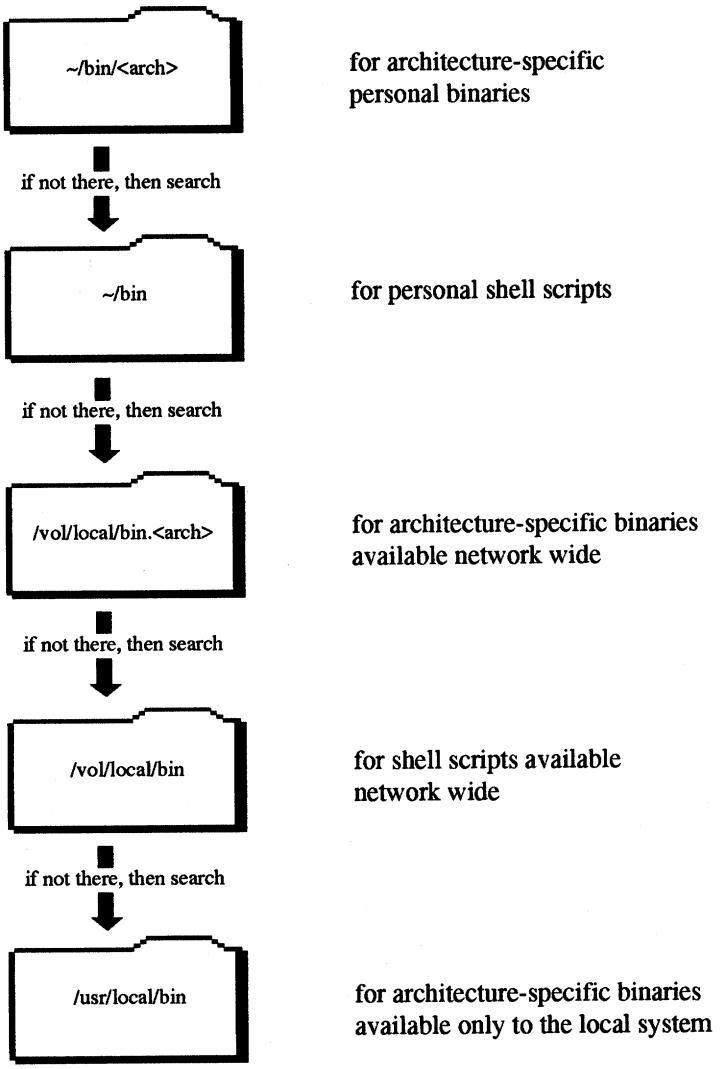
  `/vol` — `/vol` is an automounted directory. To add new automount points by hand, edit `/etc/auto.vol` on the YP master and remake the `auto.vol` YP map.

  Search path — The default search path defined in the `.cshrc` file assumes a heterogeneous site (`/vol/local/bin.`architecture, `/vol/local`, and `/usr/local`, are included in the default `.cshrc` file).



Where You Should Install Software

The diagram below shows the default user search path on a Sun386i network. Where possible, try to install applications in one of these directories.

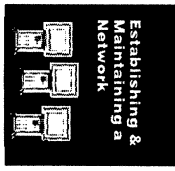


This search path allows individual users to have their own programs, allows site-wide installations across the network, supports multiple processor architectures, and permits software installation on individual workstations. Users don't have to modify their environment to take advantage of software that's been installed.

✓ **`unconfigure`** — In Sun386i SunOS 4.0 and 4.0.1, running the `unconfigure`(8) command removed software from `/files/vol`. This is fixed in Sun386i release 4.0.2.

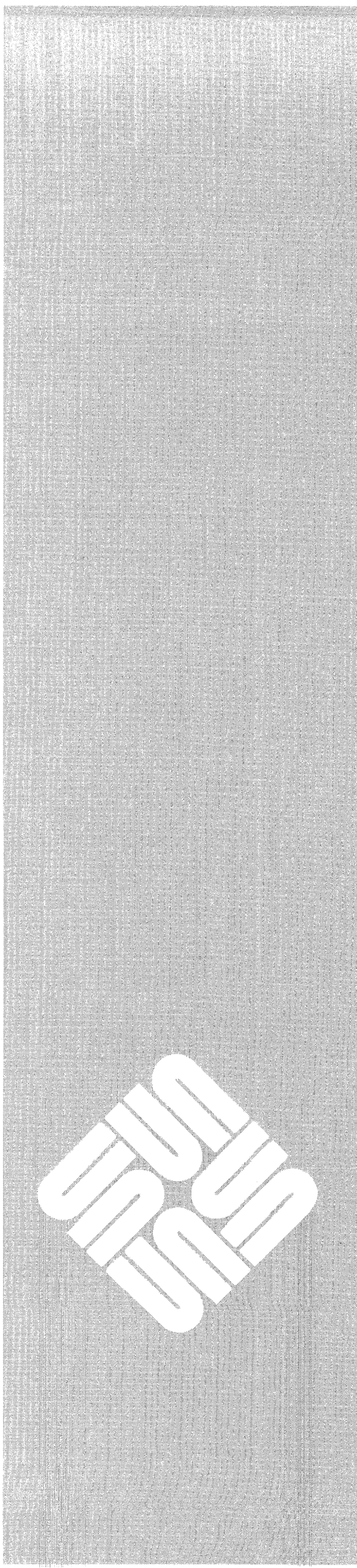
Reference: Chapter 9 of this manual ("Inside the Sun386i File System" section)
Sun386i Advanced Administration (February 1989 edition, Chapter 6)
Sun386i Developer's Guide (Chapters 6, 9)

Notes



5

HINTS AND TIPS



5

HINTS AND TIPS

Sun386*i* YP Master Hints

Sun386*i* YP Master Hints and Tips

This month's Hints and Tips include suggestions on how to solve `ypserver not responding` and `fsck` inconsistency problems when rebooting Sun386*i* machines. This rebooting was originally done by a customer trying to kill some logins.[2]

Use these hints when booting single user, using `ypinit -m`, and remaking YP does not allow proper booting.

Multiple Problems and Symptoms

You can suspect that the original problem killing some logins, `fsck` inconsistencies, and `ypserver not responding` are three symptoms of the same problem.

This problem may have caused something on the disk to be lost, allowing the YP master to become confused and look for a 'master' when it *is* is master.

The Procedure

Use the below procedure to help overcome the `ypserver not responding` error message.

1. Boot single user.
2. Use the *hostname(1)* command to set the hostname *exactly* as it is specified in the `/etc/net.conf` file.
3. Use the *domainname(1)* command to set the domainname *exactly* as it is specified in the `/etc/net.conf` file, *including* the ' `YP.` ' on the front.

---

[2] This month's hints are submitted by Chuck Kollars, Sun ECD Marketing Support, Boston Development Center.

4. Edit the `/etc/net.conf` file. Ensure than `PNP=no` . Note that `PNP=yes` means to probe the network for configuration information, which a YP master or slave should *not* do.

5. Edit the `/etc/ypservers` file. Ensure that the intended YP master system name is in this file.

6. Edit the `/etc/hosts` file. Ensure that the three following conditions exist:

   First, ensure that there is an entry for this host. It should *not* include the `loopback` alias.

   Second, ensure that there are entries for `loghost` , `timehost` , and `mailhost` . Each may an alias on the entry for this host, or may point to some other host on the network. This depends on the network configuration and the customer's wishes.

   Third, ensure that there is a separate entry for `127.0.0.1 loopback` .

7. Edit the `/etc/netgroup` file. Ensure all domainnames are specified correctly, including ' `YP.` '.

8. Edit the `/etc/publickey` file. Remove all lines except the one for `nobody` .

9. Remake the YP maps by using the commands shown below.

```
# cd /var/yp
# rm *.time
# domainname        /* Check for correct output */
# hostname          /* Check for correct output */
# make
```

10. Remove stale secure RPC key information, as shown next.

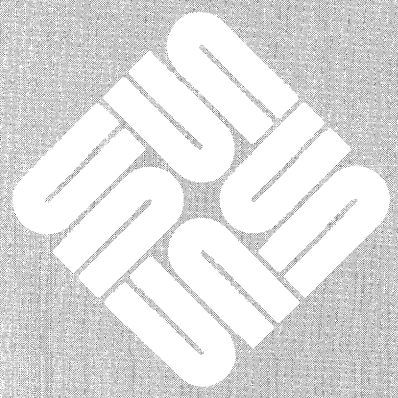
```
# cd /etc
# rm keystore
# rm .rootkey
```

11. Sync the disk and then reboot using the below commands.

```
# sync
# sync
# reboot
```

6

THE HACKERS' CORNER



6

THE HACKERS' CORNER

The `cleandisk` Script

`cleandisk`: Removes Unwanted Files

This month's **Hackers' Corner** contains `cleandisk`, a short script that you can use routinely to clean up any extraneous or temporary files that might waste your disk space.[3]

`cleandisk` runs without intervention and displays in kBytes how much disk space is being freed for each file as it is being deleted.

Why might you want `cleandisk` ? Occasionally, you will have a core dump file from a program that died or a `typescript` file that you forgot about from some `script` session you ran some time ago. Also, you may have opened a large file using `textedit` and created an equally large backup file that you no longer need. If you have such files taking up disk space, `cleandisk` may be for you.

`cleandisk` checks your entire home file system. If you put it in your `~/bin` directory (or some other location contained in your `$path` variable), you could include it in your `~/.logout` file, or in your `crontab` file for regular cleanups.

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.

Using the `cleandisk` Script

Note that you *may* wish to edit the `cleandisk` script before running. In the original version, the files that are deleted may not match those you wish to delete. Each cryptic entry is commented for your convenience.

---

[3] This month's **Hackers' Corner** is submitted by Bill Petro, GSG Marketing, Mountain View, California, USA.

Use the below procedure to run `cleandisk`.

1. Save the script to a file named `cleandisk`.
2. **`% chmod +x cleandisk`**
3. **`% cleandisk`**

The `cleandisk` Script

The code for the `cleandisk` script appears below.

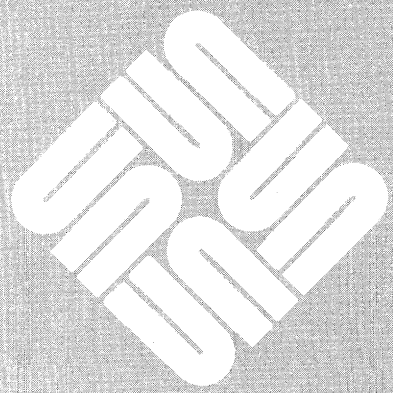
For an online copy of the `cleandisk` code, simply send email to *sun!stb-editor*. Please specify that you want the December 1989 **Hackers' Corner** code.

```
#!/bin/csh -f
# cleandisk - script that displays name and size of extraneous files while it
#             removes them
# Bill Petro - 8/89
#
# This script removes extraneous files including:
#       *.bak, *.BAK    "bak" files
#       .??*.bak        "dot" bak files
#       *.CKP'          checkpoint files
#       #*              emacs work files
#       *%              textedit backup files
#       *.o             object files
#       *.shar          shar files
#       typescript      "script" output files
#       core            core dumps (usually large)

nice find ~ '(' -name cpre -o -name \
        '*.bak' -o -name \
        '*.BAK' -o -name \
        '.??*.bak' -o -name \
        '*.CKP' -o -name \
        '#*'  -o -name \
       '*%' -o -name \
       '*.o' -o -name \
        '*.shar' -o -name \
        'typescript' -o -name \
       core ')' \
     -user $USER -type f -exec ls -s {} \; -exec \rm '{}' \;
```

7

HARDWARE, CONFIGURATIONS, & UPGRADES



7

HARDWARE, CONFIGURATIONS, & UPGRADES

Software Release Levels

As of October 27, 1989

Operating Systems

Product Name	Current Release
SunOS	4.0.3
SunOS SPARCstation 1	4.0.3c
SunOS 386*i*	4.0.2

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	6.0
SunLink MCP	6.0
SunLink X.25	6.0
SunLink Channel Adapter SCA	6.0
SunLink CG3270	6.0
SunLink MHS	6.0
SunLink HSI	6.0
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.x.	

Unbundled Languages

Product Name	Current Release
Sun C++ (Sun-3,4 and SunOS 4.x)	2.0
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 (Sun-2,3,4 and SunOS 3.x,Sys4-3.2)	2.0
Cross Compilers (SunOS 4.x, Sun-3,4***)	3.0
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.x.

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

*** Runs on Sun-3,4 and produces output that also runs on Sun386*i*.

**** 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	3.0
SunPHIGS	1.1
Sun58TE	1.0

Unbundled Applications

Product Name	Current Release
SunSimplify	1.1
SunTrac (Sun-2)	1.2
SunTrac (Sun-3,4,386*i*)	1.3
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.

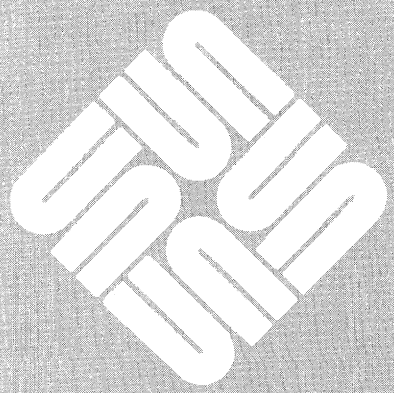
You will note that the Software Technical Bulletin (STB) contains articles from time to time that detail technical changes in a given software product's next available release.

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.

8

CUMULATIVE INDEX: 1989



Index

C

D

E

F

G

H

I

K

L

M

N

T