Contents

Preface v

1. Introducing the Source Compatibility Package 1
   What the Compatibility Package Is 1
   Why the Compatibility Package Is Provided 1
   Installing the Compatibility Package 2
   Using the Compatibility Package 2

2. Compatibility Package Features 5
   Source Compatibility Package Contents 5
       Commands 5
       Printing Commands 13
       Mail Commands 14
       System Management 15
       System Calls 15
       Signals 18
       Library Routines 19
       Routines in libcurses 20
       Routines in libdbm 21
       Routines in libtermcap 21
       Routines in librpcsoc 21
The SunOS 5.x operating system is based on AT&T’s System V Release 4.0 (SVR4) and differs considerably from previous releases of the SunOS operating system. We realize these differences can be confusing and may require a period of adjustment. To ease this transition, the SunOS/BSD Source Compatibility Package is provided. This package contains many of the commands, library routines, and files you are familiar with from the SunOS environment.

Scope of this Manual

In the context of this guide, SunOS 4.1 is a universal term that includes these releases:

- SunOS 4.1
- SunOS 4.1.1
- SunOS 4.1.2
- SunOS 4.1.3

A complete discussion of general compatibility issues can be found in Solaris 1.x to 2.x Transition Guide. This guide provides a comparative listing of SunOS 4.1 interfaces and their SunOS 5.2 counterparts.

How This Book Is Organized

This guide is provided to document the contents of this package, its usage, and how to install it. Chapter 1 provides an overview of the package and explains the mechanics of using it. Chapter 2 provides detailed descriptions of each interface provided in this package. In some cases, a brief tutorial is also provided.
Notations

References to literal computer input (commands you enter) are shown in a bold, monospace font, and computer output (screen messages produced by the system) are shown in a monospace font, as in the following example:

```bash
$ ls --l report
--rw--r-----r---- 1 jim doc 3239 May 26 11:21 report
```

Substitutable text elements (that is, text elements that you are expected to replace with specific values) are shown in an italic font, as in the following example:

```
$ cat filename
```

The italic font is a signal that you are expected to replace the word filename with the name of a file.

Instructions to the reader to type input usually do not include explicit instructions to press the RETURN key at the appropriate times (such as after entering a command or a menu choice) because this instruction is implied for all system commands and menus.

However, an instruction to press the RETURN key is explicitly provided during an interactive routine, when you are expected to press RETURN without having typed any text. An instruction to do so will be provided, as follows:

```
Type any key to continue: RETURN
```

Control characters are shown by the string CTRL-char where char is any character. To enter a control character, hold down the CTRL key and press the letter shown. Be sure to type the letter exactly as specified: when a lowercase letter is shown (such as the “d” in the example above), enter that lowercase letter. If a character is shown in uppercase (such as “CTRL-D”) you should enter an uppercase letter.

The system prompt signs shown in examples of interactive sessions are the standard default prompt signs for the system under the following conditions:

- the dollar sign ($) for an ordinary user
- the pound sign (#) for the owner of the root login

Ordering Sun Documents

The SunDocsSM program provides more than 250 manuals from Sun Microsystems, Inc. If you live in the United States, Canada, Europe, or Japan, you can purchase documentation sets or individual manuals using this program.
For a list of documents and how to order them, see the catalog section of SunExpress™ On The Internet at http://www.sun.com/sunexpress.
Introducing the Source Compatibility Package

The SunOS/BSD Source Compatibility Package is an optional package in the SunOS 5.x operating system that makes available many SunOS 4.1 and BSD interfaces otherwise not included, or which may differ in functionality between the SunOS 4.1 and SunOS 5.x releases. This package should be used as a transition aid, and your future development should not depend on its continued existence.

What the Compatibility Package Is

The Source Compatibility Package is a collection of commands, libraries and header files. Many of these may also be present in the default SunOS 5.x environment, but their behavior is different between the two versions. These interfaces are located in the /usr/ucb, /usr/ucblib, and /usr/ucbinclude directories.

Why the Compatibility Package Is Provided

This package and the SunOS Binary Compatibility Package (see the Binary Compatibility Guide) are provided as transition aids to help users migrate from SunOS 4.1 to SunOS 5.x. While applications are being ported, this package is available to aid in compiling the existing sources.
Installing the Compatibility Package

This package is optional that is, it is not automatically installed. This package is not required by any default SunOS 5.x commands, utilities, or libraries. Refer to Solaris Advanced Installation Guide or x86: Installing Solaris Software for instructions on how to install this package.

The contents of the compatibility package are installed into several directories, each corresponding to a default environment counterpart:

- /usr/ucb contains the source compatibility package commands. These commands existed in the /usr/bin, /usr/ucb, and /usr/etc directories in the SunOS 4.1 release.
- /usr/ucblib contains the source compatibility package libraries and SunOS/BSD system calls that are implemented here as library routines. These interfaces existed in /usr/lib in the SunOS 4.1 release.
- /usr/ucbinclude contains the source compatibility package header files. These files existed in the /usr/include directory in the SunOS 4.1 release.

Using the Compatibility Package

You can make use of the compatibility package in several ways: permanently, temporarily, or on a case-by-case basis.

To always get the compatibility package version of a command, you must set your PATH environment variable so that /usr/ucb appears before /usr/bin.

The following command displays the current setting of your PATH environment variable:

$ echo $PATH
/home/medici/bin:/bin:/usr/ucb:/etc:.  
$

For the current shell to find the compatibility package commands by default, switch the order of /usr/bin and /usr/ucb:

$ PATH=/home/medici/bin:/usr/ucb:/usr/bin:/etc:.  
$
To make this change permanent, put the above line in the .profile file in your home directory.

**Note** - If you’re using the C shell, use the following command to switch the order of /usr/bin and /usr/ucb. To make this change permanent in the C shell, put this line in the .cshrc file in your home directory. Changes to the .cshrc file do not take effect until that file has been sourced. See the csh(1) manual page for more information.

```bash
% set PATH=/home/medici/bin /usr/ucb /usr/bin /etc .
%
```

You can keep the SunOS 5.x commands as your default and still access the compatibility package commands by specifying the command’s full path. This is useful when you are interested in the compatibility package version of a particular command only for a given instance:

```bash
$ which df
/usr/bin/df
$ df
/ (/dev/root ): 231024 blocks 105323 files
/proc (/proc ): 0 blocks 123 files
$ /usr/ucb/df
Filesystem kbytes used avail capacity Mounted on
/dev/root 1750592 826496 749032 52% /
```

Compatibility package header files and libraries used by the compatibility package version of the C compiler (cc) and linker (ld) are located in /usr/ucbinclude and /usr/ucblib.

By setting your path as shown above, you’ll use /usr/ucb/cc when you compile C programs. /usr/ucb/cc sets its default paths to pick up the following directories in the order specified:

1. User-specified include directories and libraries.
2. The compatibility include files and directories.
3. The base SunOS 5.x system headers and libraries, if unresolved symbols remain.
Compatibility Package Features

This chapter describes the contents of the Source Compatibility Package. The contents are organized as follows:

- Commands
- System calls
- Signals
- Library routines
- Header files

For more information on the interfaces described here, see the SunOS Reference Manual.

Source Compatibility Package Contents

This section lists the contents of the Source Compatibility Package, and then describes the individual interfaces.

Commands

The commands below are described in alphabetical order. Following those descriptions, some common usage of related commands is discussed. These commands are located in /usr/ucb.

Table 2-1 below lists the commands available with the Source Compatibility Package. Two versions of some commands exist: one in the default SunOS 5.x command location, and the other in the compatibility package found in /usr/ucb. These
commands are shown in bold in the table below. The descriptions in this guide
pertain only to the /usr/ucb versions, although, where relevant, the differences
between the two versions are discussed.

<table>
<thead>
<tr>
<th>TABLE 2–1  Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commands</td>
</tr>
<tr>
<td>Mail</td>
</tr>
<tr>
<td>hp722</td>
</tr>
<tr>
<td>renice</td>
</tr>
<tr>
<td>aedplot</td>
</tr>
<tr>
<td>hpplot</td>
</tr>
<tr>
<td>plot reset</td>
</tr>
<tr>
<td>arch</td>
</tr>
<tr>
<td>implot</td>
</tr>
<tr>
<td>rusage</td>
</tr>
<tr>
<td>atoplot</td>
</tr>
<tr>
<td>install</td>
</tr>
<tr>
<td>sed</td>
</tr>
<tr>
<td>basename</td>
</tr>
<tr>
<td>ldd</td>
</tr>
<tr>
<td>shutdown</td>
</tr>
<tr>
<td>bgplot</td>
</tr>
<tr>
<td>lint</td>
</tr>
<tr>
<td>stty</td>
</tr>
<tr>
<td>biff</td>
</tr>
<tr>
<td>ln</td>
</tr>
<tr>
<td>sum</td>
</tr>
<tr>
<td>cc</td>
</tr>
<tr>
<td>logger</td>
</tr>
<tr>
<td>t300</td>
</tr>
<tr>
<td>chown</td>
</tr>
<tr>
<td>lpc</td>
</tr>
<tr>
<td>t300s</td>
</tr>
<tr>
<td>crtplot</td>
</tr>
<tr>
<td>lpq</td>
</tr>
<tr>
<td>t4013</td>
</tr>
<tr>
<td>df</td>
</tr>
<tr>
<td>lpr</td>
</tr>
<tr>
<td>t450</td>
</tr>
<tr>
<td>du</td>
</tr>
<tr>
<td>lprm</td>
</tr>
<tr>
<td>tek</td>
</tr>
<tr>
<td>dumbplot</td>
</tr>
<tr>
<td>lptest</td>
</tr>
<tr>
<td>test</td>
</tr>
<tr>
<td>echo</td>
</tr>
<tr>
<td>ls</td>
</tr>
<tr>
<td>touch</td>
</tr>
<tr>
<td>expr</td>
</tr>
<tr>
<td>mach</td>
</tr>
<tr>
<td>tr</td>
</tr>
<tr>
<td>fastboot</td>
</tr>
<tr>
<td>mail</td>
</tr>
<tr>
<td>tset</td>
</tr>
<tr>
<td>fasthalt</td>
</tr>
<tr>
<td>mkstr</td>
</tr>
<tr>
<td>ucblinks</td>
</tr>
<tr>
<td>file</td>
</tr>
<tr>
<td>pagesize</td>
</tr>
<tr>
<td>rusers</td>
</tr>
<tr>
<td>from</td>
</tr>
<tr>
<td>plot</td>
</tr>
<tr>
<td>vipw</td>
</tr>
<tr>
<td>gigiplot</td>
</tr>
<tr>
<td>plottoa</td>
</tr>
<tr>
<td>vplot</td>
</tr>
</tbody>
</table>
TABLE 2–1  Commands (continued)

<table>
<thead>
<tr>
<th>Commands</th>
</tr>
</thead>
<tbody>
<tr>
<td>groups</td>
</tr>
<tr>
<td>hostid</td>
</tr>
<tr>
<td>hostname</td>
</tr>
</tbody>
</table>

**arch**
Displays your machine architecture. arch uses the SunOS 5.x uname –m command. arch is not available in the default SunOS 5.x release.

**basename**
Strips the full path name off a file. If you follow the path name with a suffix, that suffix is stripped, too:

```
$ basename /usr/src/rundog.c .c
rundog
$`n
basename is useful for writing shell scripts. This version parses arguments differently than the SunOS 5.x version.

**biff**
Enables and disables immediate notification of the arrival of mail. When notification is enabled, biff displays the header and first few lines of mail messages when they are received. This command is not available in the default SunOS 5.x release.

**cc**
This command is identical to the unbundled C compiler in /usr/ccs/bin/acc, except that the following directories are included:

- User-specified include directories and libraries
- The compatibility include files and directories
- The default SunOS 5.x headers and libraries, if unresolved symbols remain

Compatibility package header files and libraries referenced by the compiler (cc) and linker (ld) are located in /usr/ucbinclude and /usr/ucblib.

**chown**

Comaptibility Package Features  7
Changes the owner of a file. This version allows you to optionally change the group as well. The group may be either a decimal group ID (GID) or a group name found in the GID file.

\texttt{df}

Displays the free disk space of currently mounted file systems. This command differs from /usr/bin/df in several ways:

- Blocks sizes are reported in units of 1024-bytes in this version of df, while /usr/bin/df uses 512-byte blocks.
- Perhaps the most obvious difference between the two versions of this command is their output format, as shown by the example below.

\begin{verbatim}
$ /usr/ucb/df
Filesystem kbytes used avail capacity Mounted on
/dev/root 1750592 826496 749032 52% /

$ df
/ (/dev/root ): 231024 blocks 105323 files
/proc (/proc ): 0 blocks 123 files
\end{verbatim}

\texttt{du}

Displays the number of disk blocks used by files in a directory. It is useful for seeing how disk space is being used. This version reports disk usage in kilobytes, while the default du command reports disk usage in blocks.

\texttt{echo}

Echoes arguments to the standard output. This version and the default echo command recognize certain escape sequences differently.

\texttt{expr}

Evaluates an expression. This version supports the following additional operators and keywords: match, substr, index, length().

\texttt{fastboot}

Reboots your machine. This command is not available in the default SunOS 5.x release.

\texttt{fasthalt}

Halts your machine. This command is not available in the default SunOS 5.x release.

\texttt{file}

Determines the file type by examining the file’s contents. This version supports symbolic links. When the \texttt{-L} option is specified for a file that is a symbolic link, file tests the file referenced by the link rather than the link itself.
Displays mail header information, including the sender and time the mail was received. This command is not available in the default SunOS 5.x release.

groups
Displays the groups to which a user belongs. Although the id command in the default SunOS 5.x release has been enhanced to perform this function, groups is also available for compatibility.

hostid
Prints the numeric identifier (in hexadecimal) of the current host. This command is not available in the SunOS 5.x release.

hostname
Displays the name of your machine and allows you to rename it. This command is not available in the base release. uname -n in the SunOS 5.x release displays the machine name, but does not allow the super-user to rename the machine.

install
Installs files into a destination directory. This version has different options and syntax than the SunOS 5.x one.

ld
This command is identical to the linker in /usr/ccs/bin/ld, except that it ensures that libraries in /usr/ucblib are linked in by ld, before the default SunOS 5.x libraries.

When unresolved symbols remain, the libraries in /usr/lib are referenced. Note that /usr/ucblib/libucb.so is always linked when this command is invoked.

lint
Checks C programs for bugs, argument type errors, and potential portability problems. This command uses the lint libraries under /usr/ucblib.

ln
Creates links to files or directories. The default ln removes an existing target if it has the requisite permissions. Because the SunOS 4.1 behavior is not obtained by default, the compatibility package contains a version of ln that does not remove an existing target.

logger
Adds entries to a system log. Priorities, tags, and files to write to may be specified.

lpc
lpc, the line printer control program, is the same as in SunOS 4.1, except that it does not make use of the file /etc/printcap. Use lpc to start and stop a printer, disable or enable a spooling queue, rearrange the order of printing jobs, or display the status of each printer. This command is not available in SunOS 5.x.

lpq
Returns information on all print jobs on the local printer; however, for remote printers, it provides the status only for jobs belonging to the user. `lpq` is the equivalent of the SunOS 5.x command `lpstat`.

The command `lpq username` provides the status of all files, on all machines (both local and remote), belonging to the named user. In these examples, user `wesson` sends files `fileX` and `fileY` to the local printer `printerA`, and `fileZ` to a remote printer:

```
$ lpr fileX
$ lpr fileY
$ lpq
Rank Owner Job Files
  active smith printerA-111230 file1
  1st wesson printerA-111231 fileX
  2nd wesson printerA-111232 fileY
  3rd jones printerA-111233 file9

$ lpr --Pfar_printer fileZ
$ lpq --Pfar_printer
Rank Owner Job Files
  4th wesson far_printer-111332 fileZ

$ lpr wesson
Rank Owner Job Files
printerA:
  1st wesson printerA-111231 fileX
  2nd wesson printerA-111232 fileY
far_printer:
  4th wesson far_printer-111332 fileZ
```

`lpr`

Sends files to the printer. The compatibility package version of `lpr` does not support the `-r` option (which removes a file after spooling it). Nor does it allow for specific fonts to be mounted on font positions 1, 2, 3, or 4. `lpr` is the equivalent of the SunOS 5.x `lp` command.

`lprm`

Removes files from the print queue. This is the SunOS 4.1 equivalent of the SunOS 5.x `cancel` command.

`lptest`

Produces a ripple pattern for testing printers.

`ls`

Lists a file or the contents of a directory. The compatibility package version of `ls` differs from the default `ls` command in two ways:

- The `-l` option does not print the group.
- The `-s` option reports block sizes in blocks of 1024 bytes, instead of 512.
mach

Reports the machine type. The SunOS 5.x uname –p command provides the same behavior. However, because many scripts and makefiles use mach, it has been placed in the compatibility package. This command is not available in default SunOS 5.x release.

Mail and mail

Reads and sends mail messages. This command is a version of mailx(1) with defaults set to behave like the BSD mail command. See the mailx(1) manual page in the SunOS Reference Manual for complete information on the BSD default settings.

mkstr

Creates a file of all error messages in a C program source file. This is useful for programs with a large number of diagnostic error messages, it reduces the size of the program, and the system overhead when running the program. This command is not available in the default SunOS 5.x release.

pagesize

Reports the size of a page of memory in bytes. This is useful for scripts. This command is not available in the default SunOS 5.x release.

plot

Graphics filters for various plotters. The following filters allow you to run plot on a variety of terminals: aedplot, atoplot, bgplot, crplot, dumbplot, gigiplot, hp7221plot, hpplot, implot, plottoa, t300, t300s, t4013, t450, tek, and vplot.

printenv

Reports environment variables as currently set. It is similar to the env command. This command is not available in the default SunOS 5.x release.

ps

Displays the status of the current processes. This version displays more information, in a different format, than that of the SunOS 5.x version.
Alters the *nice* value of running processes. This behavior is now present in the SunOS 5.x priocntl command; however, for SunOS 4.1 compatibility reasons, renice is included in the compatibility package. This command is not available in the default SunOS 5.x release.

**reset and tset**

SunOS 5.x These commands are similar to tput reset and tput init respectively; that is, they reset and set terminal-behavior characteristics. However they differ enough to be included here. reset and SunOS 5.x tset are not available in the default SunOS 5.x release.

**rusage**

Executes the given command and reports on the resource usage. rusage reports the total time elapsed (wall clock), the system CPU, the user CPU, plus 11 other fields in a one line summary following the commands output, if any. This command is not available in the default SunOS 5.x release.

**sed**

/usr/ucb/sed is similar to /usr/bin/sed except it strips initial SPACE and TAB characters from text lines.

**shutdown**

This version of shutdown allows you to specify a time and a warning message to be sent. It also sends shutdown messages to other systems if they have mounted filesystems from the machine being shut down. See the shutdown(1B) manual page for any limitations on SunOS 4.1 functionality.

**stty 5.2**

The SunOS 5.x version of stty incorporates many of the features of SunOS 4.1 stty. The compatibility package version is included for full SunOS 4.1 functionality.

**sum**

Calculates checksums in a machine-dependent way. Also, it displays the file in kilobytes instead of 512-byte blocks.

**test**

Evaluates conditional expressions. The SunOS 4.1 and SunOS 5.x versions of test differ with respect to the –f option. In the SunOS 4.1 release, this option means “is the target not-a-directory?” while in the SunOS 5.x release it means “is the target a plain file?”.

**touch**

Updates the access and modification times of a file. The SunOS 5.x version of touch does not support the –f option, which attempts to force the touch in spite of read and write permissions on the file.

**tr**
Translates characters from the standard input to the standard output. There are two differences between /usr/bin/tr and this version:

- In SunOS 4.1, when translating from string1 to string2, if string2 is shorter than string1, string2 is padded out to the length of string1 by repeating string2’s last character.
- This version does not require enclosing alphanumerical ranges inside square brackets.

tset
See reset.

ucblinks
Creating symbolic links in /dev for SunOS 4.1 devices whose names are different in the SunOS 5.x release. When these links are no longer needed, they must be removed by hand.

users
Displays a short list of logged-in users. This command is not available in the default SunOS 5.x release. The SunOS 5.x equivalent is who –q.

vipw
Edits the /etc/passwd file, and sets the appropriate locks. vipw runs consistency checks on the password file when the edits are complete and the locks have been removed. If these checks fail, vipw does not allow the incorrect file to be installed. This command is not available in the default SunOS 5.x release.

whereis
Locates the binary, source, and manual page files for the given command. whereis searches a standard list of common locations for these files. This command is not available in the default SunOS 5.x release.

whoami
Displays your effective user ID; whoami works even if you have used su to change your current user ID. This command is not available in the default SunOS 5.x release, although it is equivalent to id.

Printing Commands

The compatibility package contains five printing commands that are not in the default SunOS 5.x release: lpr, lpq, lprm, lpc, and lptest. This is how they compare with the base commands:
TABLE 2–2 Printing Commands

<table>
<thead>
<tr>
<th>SunOS 4.1 Command</th>
<th>SunOS 5.x Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td>lpr</td>
<td>lp</td>
</tr>
<tr>
<td>lpq</td>
<td>lpstat</td>
</tr>
<tr>
<td>lprm</td>
<td>cancel</td>
</tr>
<tr>
<td>lpc</td>
<td>no equivalent</td>
</tr>
<tr>
<td>lptest</td>
<td>no equivalent</td>
</tr>
</tbody>
</table>

Here is an overview of some important features:

- There are no SunOS 4.1 commands for administration, maintenance, or accounting of printers; you must use SunOS 5.x schemes. The exception is the lpc command, which is included.

- The SunOS 4.1 printcap file does not exist. The printer information is now contained in the files in the directories /etc/lp and /etc/lp/printers/printer-name. You can change this information using the lpadmin command. For more information on this command, refer to the lpadmin(1) manual page in the SunOS Reference Manual.

- Printing spooling commands do not depend on the lpd daemon, as in SunOS 4.1. Instead, they use an HPI (High Performance Interface) with the SunOS 5.x printing daemon, lpsched.

Mail Commands

The SunOS 4.1 version of mail, based on sendmail, has been included in the Source Compatibility Package; it has a different user interface than the base SunOS 5.x version of mail. Additionally, some useful mail facilities are included for compatibility.

TABLE 2–3 Mail Commands

<table>
<thead>
<tr>
<th>Commands</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>biff</td>
<td>immediate notification of mail</td>
</tr>
<tr>
<td>from</td>
<td>display mail header lines</td>
</tr>
</tbody>
</table>
### System Management

Because of limitations in underlying SunOS 5.x support, the commands in this section for stopping and starting your system cannot provide full SunOS 4.1 functionality. Nonetheless, they are included here for compatibility, so that existing shell scripts invoking them can still function. For example, the compatibility package versions of fastboot and fasthalt do not prevent disk checks. See the manual pages on following commands to find out about specific limitations: fastboot, fasthalt, and shutdown.

### System Calls

The following system calls are implemented in the compatibility package as library routines:

#### TABLE 2-4 System Calls

<table>
<thead>
<tr>
<th>System Calls</th>
</tr>
</thead>
<tbody>
<tr>
<td>WIFEXITED</td>
</tr>
<tr>
<td>WIFSIGNALED</td>
</tr>
<tr>
<td>WIFSTOPPED</td>
</tr>
<tr>
<td>flock</td>
</tr>
<tr>
<td>fstatfs</td>
</tr>
<tr>
<td>getdtablesize</td>
</tr>
<tr>
<td>gethostid</td>
</tr>
<tr>
<td>gethostname</td>
</tr>
</tbody>
</table>
The following routines reside in libucb.

flock
Applies or removes an advisory lock on a file. The compatibility version of flock has been implemented on top of fcntl(2) locking. This version is not completely compatible with the SunOS 4.1 version. The following differences exist:

- To obtain a shared lock (LOCK_SH), read permission on file is required.
- To obtain an exclusive lock (LOCK_EX), write permission on file is required.
- EOPNOTSUPP may not be returned as a result of a call to flock.
- Locking a segment that is already locked by the calling process causes the old lock type to be removed and the new lock type to take effect. That was not the case with flock in the SunOS 4.1 release.
- Locks are not inherited by a child process in a fork(2) function.
- Locks obtained through the flock mechanism under SunOS 4.1 were known only within the system on which they were placed. This is no longer true.

fstatfs
Gets statistics about an open file on a mounted file system.

getdtablesize
Gets the size of a process’s descriptor table. This routine calls the getrlimit(2) system call.

gethostid
Returns the 32-bit identifier for the current host. This routine calls the sysinfo(2) system call.

gethostname
Returns the standard hostname for the current processor. This routine calls the sysinfo(2) system call.

getpagesize
Returns the number of bytes in a page. This routine calls the sysconf(3C) routine.

getpriority
Gets the process scheduling priority. Returns the highest priority (lowest numerical value) of any of the specified processes. This routine calls the priocntl(2) system call.

getimeofday
Gets date and time. This routine calls the default version of gettimeofday(3C). Note that this version of gettimeofday accepts two arguments for compatibility purposes, although it ignores the second one.

killpg
Sends a signal to a process group. This routine calls the kill(2) system call.

mctl
Applies memory management control functions. This routine calls the memcntl(3) routine.

reboot
Restart system or halt processor. This routine calls the uadmin(2) system call.

sethostname
Sets the name of the host machine. This routine calls the sysinfo(2) system call.

setpriority
Sets the program scheduling priority. The default priority is 0. It is not possible for the process executing setpriority to lower any other process down to its current priority, without appropriate privileges. This routine calls the priocntl(2) and priocntlset(2) system calls.

setregid
Set real and effective group ID’s. This routine calls the setgid(2) or setegid system calls. Exact BSD semantics are not always maintained. The real gid is not always changed. Not all cases can be handled.

setreuid
Sets real and effective user ID’s. Calls SunOS 5.x’s setuid or seteuid. Exact BSD semantics are not always maintained. The real uid is not always changed. Not all cases can be handled.

settimeofday
Sets date and time of day. This routine calls the stime(2) system call. Note that this function accepts two arguments for compatibility purposes, although it ignores the second.

statfs
Gets file system statistics about mounted file systems. Calls statvfs(2).

syscall
Performs indirect system calls.

wait3
Alternative system call to wait(2) for waiting for a process to stop. This routine calls the waitid(2) system call.
wait4
Alternative system call to wait(2) for waiting for a process to stop. This routine calls the waitid(2) system call.

Signals
The SunOS 4.1 signal handlers and default actions are different from the SunOS 5.x signals.
The following routines are in the compatibility package:

<table>
<thead>
<tr>
<th>TABLE 2–5 Signals</th>
</tr>
</thead>
<tbody>
<tr>
<td>Signals</td>
</tr>
<tr>
<td>_longjmp</td>
</tr>
<tr>
<td>_setjmp</td>
</tr>
<tr>
<td>longjmp</td>
</tr>
<tr>
<td>psignal</td>
</tr>
<tr>
<td>setjmp</td>
</tr>
</tbody>
</table>

The SunOS 4.1 style of signal handling is provided in the compatibility package. These routines invoke existing SunOS 5.x signal handling routines. These routines reside in libucb.

_ longjmp, longjmp, _ setjmp, and setjmp

In the SunOS 4.1 release, setjmp and longjmp save and restore the signal mask. This functionality has been preserved with the sigsetjmp and siglongjmp signals, respectively.

psignal

In the SunOS 4.1 release, psignal flushes stdio buffers. The SunOS 5.x version of psignal does not. This behavior has been preserved in the compatibility version.

sigblock and sigsetmask

In the SunOS 4.1 release, sigblock blocks a signal; sigsetmask sets a signal mask. sigblock and sigsetmask call the sigprocmask(2) system call.

siginterrupt, signal, sigvec and sigaction
In the SunOS 4.1 release, siginterrupt is a library function that allows signal interruptions; signal is a library function for simplified signal processing. Both now call sigvec, which itself calls the sigaction routine.

**sigpause**

Releases block signals; waits for interrupts. Implemented in the compatibility package through sigsuspend.

## Library Routines

The Source Compatibility Package includes a set of compatibility libraries. These libraries can be found in `/usr/ucblib`, and they consist of:

- Routines not in the default SunOS 5.x system
- Routines having different interfaces or different system call “wrappers” than their SunOS 5.x counterparts

The routines in Table 2-6 are available with the compatibility package. Routines shown below in bold also exist in the default system:

### TABLE 2-6 Library Routines

<table>
<thead>
<tr>
<th>Library Routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>alphasort</td>
</tr>
<tr>
<td>bcmp</td>
</tr>
<tr>
<td>bcopy</td>
</tr>
<tr>
<td>bzero</td>
</tr>
<tr>
<td>fopen</td>
</tr>
<tr>
<td>fprintf</td>
</tr>
<tr>
<td>freopen</td>
</tr>
<tr>
<td>ftime</td>
</tr>
<tr>
<td>getwd</td>
</tr>
<tr>
<td>index</td>
</tr>
<tr>
<td>initstate</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Library Routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>printf</td>
</tr>
<tr>
<td>rand</td>
</tr>
<tr>
<td>random</td>
</tr>
<tr>
<td>re_comp</td>
</tr>
<tr>
<td>re_exec</td>
</tr>
<tr>
<td>readdir</td>
</tr>
<tr>
<td>rindex</td>
</tr>
<tr>
<td>scandir</td>
</tr>
<tr>
<td>setbuf</td>
</tr>
<tr>
<td>setbuffer</td>
</tr>
<tr>
<td>setlinebuf</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Library Routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>setvbuf</td>
</tr>
<tr>
<td>sleep</td>
</tr>
<tr>
<td>sprintf</td>
</tr>
<tr>
<td>srand</td>
</tr>
<tr>
<td>srandom</td>
</tr>
<tr>
<td>sys_siglist</td>
</tr>
<tr>
<td>times</td>
</tr>
<tr>
<td>ualarm</td>
</tr>
<tr>
<td>usleep</td>
</tr>
<tr>
<td>vfprintf</td>
</tr>
</tbody>
</table>

**Compatibility Package Features** 19
Compatibility library routines are placed in one of the following libraries: libcurses, libdbm, librpcsoc, libtermcap, and libucb.

The descriptions of the routines found in these libraries are divided by the libraries. Many of these routines are not included in the default SunOS 5.x libraries, but are useful enough to be retained for backward compatibility.

The routines in Table 2-6 are included in compatibility libraries for backwards compatibility purposes. At some later date these routines may not be supported and may eventually be discarded.

## Routines in libcurses

The routines listed in Table 2-7 are available in the /usr/ucblib version of libcurses. Those routines shown in bold are also available in the default libcurses. The source compatibility version of libcurses uses termcap, while the base version uses terminfo.

### TABLE 2-7  Curses Routines

<table>
<thead>
<tr>
<th>Curses Routines</th>
<th>Curses Routines</th>
<th>Curses Routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>box</td>
<td>overlay</td>
<td>welrtobot</td>
</tr>
<tr>
<td>delwin</td>
<td>overwrite</td>
<td>welrtoeol</td>
</tr>
<tr>
<td>endwin</td>
<td>plod</td>
<td>wdelch</td>
</tr>
<tr>
<td>fgoto</td>
<td>plodput</td>
<td>wdeleteln</td>
</tr>
<tr>
<td>fullmode</td>
<td>printw</td>
<td>werase</td>
</tr>
<tr>
<td>getcap</td>
<td>scanw</td>
<td>wgetch</td>
</tr>
<tr>
<td>getmode</td>
<td>scroll</td>
<td>wgetstr</td>
</tr>
<tr>
<td>idlok</td>
<td>setterm</td>
<td>winsch</td>
</tr>
</tbody>
</table>
TABLE 2–7  Curses Routines  (continued)

<table>
<thead>
<tr>
<th>Curses Routines</th>
</tr>
</thead>
<tbody>
<tr>
<td>initscr</td>
</tr>
<tr>
<td>longname</td>
</tr>
<tr>
<td>mvcur</td>
</tr>
<tr>
<td>mvprintw</td>
</tr>
<tr>
<td>mvscanw</td>
</tr>
<tr>
<td>mvwin</td>
</tr>
<tr>
<td>mvwprintw</td>
</tr>
<tr>
<td>mvwscanw</td>
</tr>
<tr>
<td>newwin</td>
</tr>
</tbody>
</table>

Routines in libdbm

libdbm contains the following routines: dbmclose, dbminitdelete, fetch, firstkey, nextkey, and store. These routines maintain key/content pairs in a database using database routines. They are packaged in /usr/ucblib/libdbm. For more information, see the appropriate manual pages. These routines are superceded by the libndbm routines in the default system.

Routines in libtermcap

libtermcap contains the following routines: tgetent, tgetnum, tgetflag, tgetstr, tgoto, and tputs.

Routines in librpcsoc

librpcsoc contains the following routines: clnttcp_create, clntudp_bufcreate, clntudp_create, rtime, svcfd_create, svctcp_create, svcudp_bufcreate, svcudp_create, and svcudp_enablecache.
Routines found in this library expect a socket descriptor as an argument.

Routines in libucb

alphasort and scandir

These routines combine the functions of readdir and qsort to process directory information. scandir reads a directory and builds an array of pointers to directory entries, returning the number of entries in the array and a pointer to the array through the parameter, namelist. namelist is a pointer to an arrangement of directory structure pointers. If the directory cannot be opened for reading or if malloc cannot allocate enough memory to hold all the data structures, scandir returns −1. alphasort is a routine that alphabetically sorts an array of pointers to directory entries (built by the routine, scandir).

These routines are not available in the default SunOS 5.x release.

bcmp, bcopy, and bzero

Functionally equivalent to the memcmp, memcpy, and memset routines, see memory(3C). bcopy copies bytes, handling overlapping strings correctly. bcmp compares bytes, returning zero if they are identical, non-zero otherwise. bzero zeros out bytes. These routines are not available in the default SunOS 5.x release.

fopen and freopen

The SunOS 5.x and SunOS 4.1 versions differ in the handling of the a type. If mode a is specified, the SunOS 5.x version opens the file with O_APPEND set, while the SunOS 4.1 version doesn’t. Also, the SunOS 4.1 version always seeks to the end of the file when mode a is specified, while the SunOS 5.x version seeks to the end of the file if update mode is not specified.

ftime

Returns a structure that contains elements showing the elapsed time since the epoch (00:00:00, January 1st, 1970). See ctime(3C). This routine is not available in the default SunOS 5.x release.

getwd

Similar to getcwd. getwd returns the pathname of the current working directory, or NULL if there is an error. The return value of getwd is placed in a character string allocated by the caller. This routine is not available in the default SunOS 5.x release.

index and rindex

index returns a pointer to the first occurrence of a single character in a null-terminated character string. If the character does not occur in the string, index returns a NULL pointer. In the SunOS 5.x release, strchr replaces index, see string(3C).
rindex returns a pointer to the last occurrence of a single character in a null-terminated string, or a NULL pointer if the letter does not occur in the string. In the SunOS 5.x release, strrchr replaces rindex, see string(3C).

These routines are not available in the default SunOS 5.x release.

initstate, random, setstate, and srandom

Random number generator routines relied on by many user programs in the SunOS 4.1 release. These routines are not available in the default SunOS 5.x release.

See rand(3C), srand(3C), and drand(3C) in the SunOS Reference Manual.

nice

The SunOS 4.1 version always returns 0 if successful. The SunOS 5.x version returns the new nice value minus 20.

nlist

If unsuccessful, returns the number of symbols that were not located in the symbol table. (The SunOS 5.x version returns −1 if unsuccessful.) Also, nlist takes a file name as an argument, rather than a file descriptor.

printf, fprintf, sprintf, vfprintf, vprintf, and vsprintf

The following differences exist for these functions:

- In the SunOS 4.1 release, sprintf returns a pointer to its first argument; in the SunOS 5.x release, it returns the count of characters printed.
- In the SunOS 4.1 release, printf and vprintf always return the number of characters printed or EOF, while the SunOS 5.x version returns only EOF in case of an error.
- fprintf and vfprintf share the same differences as printf and vprintf (above); additionally, the buffering scheme for these routines is different for the SunOS 5.x and SunOS 4.1 release.

rand and srand

The SunOS 5.x returns the generated number shifted to the right by 16 and anded with 0x7fff, while the SunOS 4.1 release returns the generated number anded with 0x7fffffff.

re_comp and re_exec

These routines are similar to the regcmp(3G) and regex routines. re_comp compiles the regular expression in a string, returning 0 if it compiled successfully or returning a pointer to an error message if it did not. regcmp, on the other hand, returns a pointer to the compiled form or NULL if given an incorrect argument. re_exec returns 1 if the string s matches the last compiled regular expression, 0 if it fails to match, and −1 if the regular expression is invalid. regex, however, returns NULL on failure or a pointer to the next unmatched character on success.

readdir
Reads a directory entry. The SunOS 4.1 release readdir routine returns a pointer to struct direct, while the SunOS 5.x version returns a pointer to struct dirent.

setbuffer and setlinebuf

These routines are used to provide control over the buffering used for input and output.

Similar to setbuf(3S) and setvbuf, setbuffer may be used after a stream has been opened but before it is read or written. It uses the array pointer instead of an automatically allocated buffer. If the buffer is the NULL pointer, input/output is completely unbuffered.

setlinebuf is used to change the buffering on a stream from block buffered or unbuffered to line buffered. It can be used at any time that the file descriptor is active.

These routines are not available in the default SunOS 5.x release.

sleep

The SunOS 5.x version returns the “unslept” amount, that is, the requested time minus the time actually slept. The SunOS 4.1 version does not return anything.

sys_siglist

An array containing a message string for each SunOS 4.1 signal number.

times

The SunOS 4.1 version returns 0 when successful. The SunOS 5.x version returns the elapsed real time in clock ticks per second from an arbitrary point in the past.

ualarm and usleep

Simplified interfaces to the system calls handling high-resolution timers. These routines are not available in the default SunOS 5.x release.

Header Files

The Source Compatibility Package header files can be found in /usr/ucbinclude. Differences remain in the case of header files, and you should be aware of them; in some cases it was impossible to achieve complete source compatibility. Some SunOS 4.1 header files do not exist in the default SunOS 5.x release. The header files in bold are also available in the default system.
As with the library routines, there are some header files under SunOS 5.x that differ slightly from their SunOS 4.1 counterparts. In those instances, the affected header files are included in this compatibility package with as many additions as possible to keep them compatible with SunOS 4.1. The following list shows those files affected and the reasons for their inclusion in this package:

<table>
<thead>
<tr>
<th>Header Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>a.out.h</td>
</tr>
<tr>
<td>assert.h</td>
</tr>
<tr>
<td>curses.h</td>
</tr>
<tr>
<td>dbm.h</td>
</tr>
<tr>
<td>fcntl.h</td>
</tr>
<tr>
<td>regexp.h</td>
</tr>
<tr>
<td>resolv.h</td>
</tr>
<tr>
<td>setjmp.h</td>
</tr>
<tr>
<td>sgtty.h</td>
</tr>
<tr>
<td>signal.h</td>
</tr>
<tr>
<td>stdio.h</td>
</tr>
<tr>
<td>strings.h</td>
</tr>
</tbody>
</table>

As with the library routines, there are some header files under SunOS 5.x that differ slightly from their SunOS 4.1 counterparts. In those instances, the affected header files are included in this compatibility package with as many additions as possible to keep them compatible with SunOS 4.1. The following list shows those files affected and the reasons for their inclusion in this package:

- **a.out.h**
  - Included for compatibility.

- **assert.h**
  - Included for compatibility.

- **curses.h**
  - Included for compatibility.

- **dbm.h**
  - Used by routines in libdbm and user programs invoking libdbm.

- **fcntl.h**
Included for compatibility.

regexp.h

Includes ctype.h for compatibility.

resolv.h

Included for compatibility.

setjmp.h

Includes additional #defines and #includes for compatibility.

sgtty.h

Included for compatibility.

signal.h

Contains sys/signal.h. Included here for compatibility.

stdio.h

Includes SunOS 4.1-specific definitions.

strings.h

Contains the same type of information as string.h.

struct.h

Contains useful macros.

sys/dir.h

Included for compatibility.

sys/fcntl.h

Includes missing #defines.

sys/file.h

Includes missing #defines such as L_SET, and L_INCR.

sys/ioctl.h

Included for compatibility.

sys/param.h

Includes missing #defines.

sys/reboot.h

Included for compatibility.

sys/resource.h

Includes the rusage structure missing from the SunOS 5.x version of resource.h and used by the getrusage function.