

# Guide to Operations

SUN Workstation<sup>™</sup>

February 15, 1990

P/N: 900-00736

Valid Logic Systems Incorporated  
2820 Orchard Parkway  
San Jose, California 95134  
(408) 432-9400 Telex 371 9004  
FAX (408) 432-9430

Copyright © 1990 Valid Logic Systems Incorporated

This document contains confidential proprietary information which is not to be disclosed to unauthorized persons without the prior written consent of an officer of Valid Logic Systems Incorporated.

The copyright notice appearing above is included to provide statutory protection in the event of unauthorized or unintentional public disclosure.

While every attempt has been made to keep the information in this document as accurate and as current as possible, Valid makes no warranty, expressed or implied, with regard to the information contained herein, including, but not limited to, the implied warranties of merchantability and fitness for any particular application. Valid further assumes no responsibility for any errors that may appear within this document or for any damages, direct or indirect, that may result from using this document.

Allegro, SCALDsystem, ValidGED, ValidCOMPILER, ValidPACKAGER, ValidSIM, ValidTIME, Transcribe, Realchip, and Realmodel are trademarks of Valid Logic Systems Incorporated

Sun Workstation is a trademark of Sun Microsystems, Incorporated

UNIX is a trademark of Bell Laboratories

Ethernet is a trademark of Xerox Corporation

## MANUAL REVISION HISTORY

Rev	Date	Software Release	Reason for Change
A	9/1/87	Valid System Release 1.0	Initial release.
B	12/30/88	Sun Release 4.0	Added Sun 4 and Sun OS 4.0 support, updated <i>Plotting</i> (Section 6) to include Hardcopy 3.4 support.
C	8/30/89	Valid Q4 Release	Revised Valid software installation procedure (Section 2).
D	2/15/90	Valid Q1 Release	Reformatted manual, updated disk partitions and installation (Section 2), updated access control features, revised Section 4, added Appendices B, C, and D.



# Table of Contents

<b>OVERVIEW</b> .....	<b>1-1</b>
<b>INSTALLING VALID SOFTWARE</b> .....	<b>2-1</b>
Installation Requirements .....	2-2
Installation Overview .....	2-3
Determining Disk Requirements .....	2-4
Valid Software Organization .....	2-5
Disk Space Requirements .....	2-6
Valid Default Disk Partitions .....	2-8
Valid Software on Standalone Systems .....	2-9
Valid Software on Servers .....	2-10
Sun Software Installation .....	2-11
Valid Software Installation .....	2-12
First-time Installations .....	2-13
Installing the Valid Applications .....	2-16
Using <i>vscrip</i> t .....	2-17
Installing Sun 3 and Sun 4 Tools on a Server .....	2-21
Access Control .....	2-22
Configuring Servers .....	2-23
Editing the Template Files .....	2-25
Adding and Configuring Clients .....	2-29
Access Control Licenses .....	2-33
Completing the Installation .....	2-34

<b>ADMINISTERING ACCESS CONTROL . . . . .</b>	<b>3-1</b>
Access Control Overview . . . . .	3-2
Starting the License Manager . . . . .	3-2
The Fault Tolerant Mode . . . . .	3-3
Restarting the Daemons . . . . .	3-7
Adding Servers . . . . .	3-9
Access Control Files . . . . .	3-11
The License File . . . . .	3-11
Reinstalling the License File . . . . .	3-12
The <i>license.log</i> File . . . . .	3-13
The <i>Servers</i> File . . . . .	3-14
The <i>Options</i> File . . . . .	3-15
Access Control Utilities . . . . .	3-16
The <i>valid_status</i> Utility . . . . .	3-16
The <i>valid_report</i> Utility . . . . .	3-17
Troubleshooting . . . . .	3-19
 <b>ADDING AND REMOVING USERS . . . . .</b>	 <b>4-1</b>
Creating User Accounts . . . . .	4-2
System Requirements for <i>mkusr</i> . . . . .	4-2
Running <i>mkusr</i> . . . . .	4-3
Problems Running <i>mkusr</i> . . . . .	4-5
Accounts Not Created with <i>mkusr</i> . . . . .	4-7
User Passwords . . . . .	4-8
Forgotten Passwords . . . . .	4-10
Removing Users . . . . .	4-11
Default User Files . . . . .	4-12
The <i>.login</i> File . . . . .	4-13
The <i>.cshrc</i> File . . . . .	4-14
The <i>.profile</i> File . . . . .	4-16

---

---

<b>BACKUPS</b> .....	<b>5-1</b>
Utilities for Backups: Overview .....	5-3
Sample Backup Session .....	5-5
Restoring Individual Files .....	5-6
Restoring Entire File Systems .....	5-10
The <i>tar</i> Utility .....	5-14
Using <i>tar</i> .....	5-15
<b>DESIGN FILE MANAGEMENT</b> .....	<b>6-1</b>
Design Directories .....	6-3
Simple Project Account Security .....	6-4
Strict Project Account Security .....	6-4
Using <i>chmod</i> .....	6-9
File Permissions .....	6-10
Default Permissions .....	6-11
Group Permission .....	6-11
Strict Permissions .....	6-13
Permissions for Programs and Scripts .....	6-13
Special Permissions .....	6-14
Understanding SCALD Directories .....	6-15
Accessing Drawings Over the Network .....	6-17
Libraries .....	6-18
Libraries and File Permissions .....	6-18
The Master Library File .....	6-19
Mounting Valid Libraries .....	6-20
File System Maintenance .....	6-21
Maintaining the Root File System .....	6-22
Maintaining the User File Systems .....	6-23
Users Who Overuse the Disk .....	6-26
Archiving a Design .....	6-27

<b>PLOTTING</b> .....	<b>7-1</b>
Configuring The Printcap File .....	7-3
File Entry Syntax .....	7-3
System Considerations .....	7-4
Versatec Text Device .....	7-6
Local Versatec Graphics Device .....	7-9
Remote Versatec Graphics Device .....	7-10
Entries for Pen Plotters .....	7-14
Local Pen Plotter .....	7-15
Remote Pen Plotter .....	7-16
Entries for Postscript Printers .....	7-17
Local Postscript Printer .....	7-17
Remote Postscript Printer .....	7-18
GED Plotting .....	7-19
Plotting Commands and Utilities .....	7-21
<i>vpr</i> .....	7-21
<i>vpq</i> .....	7-22
<i>vprm</i> .....	7-22
The <i>lpc</i> Utility .....	7-23
Troubleshooting .....	7-24
Configuring the Printcap File .....	7-25
Hardcopy Error Log File .....	7-25
Lpd Error Logs .....	7-26
Versatec Plotter Setup .....	7-29
Setting Up Pen Plotters .....	7-29
Baud Rate .....	7-29

---

---

**APPENDIX A**

<b>IKON 10088 Interface Board Installation . . . .</b>	<b>A-1</b>
Interface Board Switch Settings . . . . .	A-3
Interface Board Jumper Settings . . . . .	A-7
Cabling . . . . .	A-11
VME-VME Adapter Installation . . . . .	A-13
Software Installation . . . . .	A-15
Non-generic Kernel Considerations . . . . .	A-16

**APPENDIX B****Real Products**

<b>Versatec Printer/Plotter Kernels . . . . .</b>	<b>B-1</b>
Installing and Configuring the Kernel . . . . .	B-2
Installing the Valid Kernel . . . . .	B-3
Configuring Custom Kernels . . . . .	B-4
Device and Spooler Scripts . . . . .	B-5

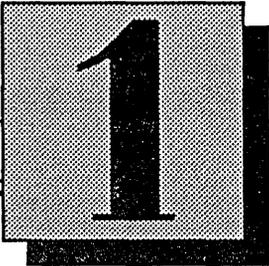
**APPENDIX C**

<b>Using <i>vin</i>stall . . . . .</b>	<b>C-1</b>
The <i>vin</i> stall Utility . . . . .	C-2
Errors and Warnings . . . . .	C-3

**APPENDIX D**

<b>Valid Products . . . . .</b>	<b>D-1</b>
Feature Names . . . . .	D-2





# 1

## *Overview*

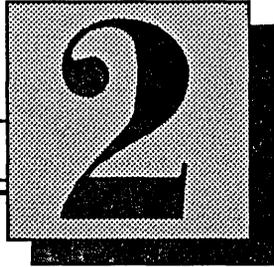
---

**T**his manual is the operations guide for Valid applications software on a SUN Workstation running under the UNIX operating system. Some of the topics covered in this manual include:

- Installing Valid software
- Creating user accounts
- Making tape backups of design data
- Managing Valid design files and directories
- Setting up plotters and printers

This manual assumes that you have a basic working knowledge of UNIX and the Sun workstation hardware. Study the *Sun Beginner's Guides* if you are not familiar with any of these areas of the system's operation.

This manual is a supplement to the *System Administration for the Sun Workstation*. Frequent references are also made to Sun UNIX manual pages in the *UNIX Commands Reference Manual* and the *UNIX Interface Reference Manual* included with your Sun Workstation documentation.



## *Installing Valid Software*

---

**T**his section explains how to configure and install Valid software on a Sun Workstation. Updates to Valid software are also discussed.

*Read this entire section before you start the installation!*

## Installation Requirements

Before you begin, you should be familiar with both the Sun software installation procedure and Sun's system administration documentation. In addition, you should have access to the Sun manual set. In particular, you should read these Sun manuals:

- *Sun System Overview.*
- *System Administration for the Sun Workstation.*
- *SunOS 4.X Release Manual* for the Sun 3 and Sun 4 workstations.
- *Installing UNIX on the Sun Workstation.*

You will need the following tapes and documentation:

- Complete Sun documentation.
- SUNBIN 4.X 68020 release and upgrade tapes for Sun 3 or 4.X SUNBIN SPARC release for Sun 4.
- Valid software installation and application tape(s).
- Valid library tape(s) (optional).
- Valid installation instructions (this manual).
- Any additional Valid installation notes that are packaged with your tapes.

## Installation Overview

*This installation procedure assumes that you are only using your Sun to run Valid applications. If you are running other software, you may need to modify this procedure.*

There are a number of steps involved in configuring and installing Valid software on your Sun workstation. Depending on your exact configuration, for example, whether or not you have already installed the Sun software on the workstation, you may be able to skip some of these steps.

The procedure is:

- Determine your local and network disk space requirements.
- Install the Sun software.
- Install the Valid software.
- Configure network file access.
- Configure your Access License servers and clients.
- Create accounts for users as explained in Section 4.
- Create Real Products devices, and spooler directories, if necessary. Configure any machines that have Real Products and/or Versatec printers and plotters by modifying the kernel, as explained in Appendix B.
- Configure printers and/or plotters. Plotter setup is explained in Section 7; installation of the Ikon 10088 printer interface board is explained in Appendix A.

## **Determining Disk Requirements**

The Network File System (NFS) and disk servers allow many possible disk configurations. If this is the first software installation on your workstation, plan your disk space needs carefully so that you do not run into problems later on.

This section describes how the Valid software and libraries can be organized on standalone systems, servers, and clients. Once you have decided how to organize the software and libraries, you can decide where they are to be installed.

Your disk space needs depend on what Valid application software you have ordered, what other non-Sun software you have installed, and the type of work you will be doing on the workstation. The disk configuration should allow enough room for your software and component libraries and some room for future growth. It should also allow enough space for the design data that will be created on the system.

---

---

## Valid Software Organization

Valid recommends installing all Valid application software and libraries in */home/valid* and creating a symbolic link to */usr/valid*. Application software is located under the directory */usr/valid/tools.sun3* for Sun 3 systems or */usr/valid/tools.sun4* for Sun 4 systems. Most software packages occupy a subdirectory in the “tools” directory. For example, the Graphic Editor (GED) is in either */usr/valid/tools.sun3/editor* or in */usr/valid/tools.sun4/editor*. The directory */usr/valid/tools* is linked to the appropriate “tools” directory (*tools.sun3* or *tools.sun4*). Libraries are common between Sun 3 and Sun 4 workstations and occupy subdirectories in */usr/valid/lib*. For example, the LSTTL library is located in */usr/valid/lib/lsttl*.

All of the Valid executable programs (or scripts to run them) are located in the “bin” directory under “tools.” To run any of these Valid programs, you must add the directory */usr/valid/tools/bin* to each user’s PATH environment variable (the link from *tools.sun3* or *tools.sun4* is followed). The procedure for adding a PATH variable is included in page 2-15 of this manual.

## Disk Space Requirements

The approximate disk space requirements for the most common Valid software configurations are listed in Tables 2-1 and 2-2. Use these lists to determine your disk space needs for Sun3 and Sun4 systems before you start the installation. If space is critical and you need more exact information, see the individual release notes sent with your Valid Software distribution. Both the library and application package sizes generally increase with each new release.

**Table 2-1. Application Package Disk Space Requirements**

Application Package	Required Disk Space		Notes
	Sun3	Sun4	
System Utilities	6.7 Mb	8.0 Mb	Required for all systems
Entry Design (ED)	10.0 Mb	12.0 Mb	Includes System Utilities
Logic Design (LD)	15.5 Mb	19.0 Mb	Includes ED
Validation Design (VLD)	22.0 Mb	27.5 Mb	Includes ED and LD
Transcribe	1.8 Mb	2.0 Mb	
Allegro (PCB)	30.0 Mb	33.0 Mb	Does not include System Utilities
Integrated Chip (IC)			Includes System Utilities
Construct	8.8 Mb	10.5 Mb	
Compose	2.9 Mb	3.4 Mb	
LED	8.7 Mb	9.6 Mb	
Timemill	1.7 Mb	1.9 Mb	
Pathmill	1.7 Mb	1.7 Mb	
Timeview	6.5 Mb	7.6 Mb	
DRC	0.5 Mb	0.7 Mb	

Table 2-2. Library Disk Space Requirements

Library	Required Disk Space *	Library	Required Disk Space *
advpal	630 KBytes	pcbib	7000 KBytes
alsttl	3500 KBytes	sttl	2320 KBytes
asttl	2860 KBytes	ttl	2840 KBytes
cmos	4230 KBytes	10k	2700 KBytes
discrete	300 KBytes	100k	1540 KBytes
fast	3120 KBytes	54alsttl	3280 KBytes
hcmos	5315 KBytes	54asttl	1940 KBytes
ifl	590 KBytes	54fast	2200 KBytes
lsttl	8880 KBytes	54hcmos	5370 KBytes
memory	3780 KBytes	54lsttl	5700 KBytes
mm74c	370 KBytes	54sstl	1099 KBytes
newpal20	810 KBytes	54ttl	1200 KBytes
newpal24	1150 KBytes		

\* Please note that library size remains constant for both Sun3 and Sun4 systems

## Valid Default Disk Partitions

Valid's disk partition default is to install software in the */home* partition on a single disk system, and */home1* on a two-disk system. Valid recommends that the */usr* partition contain SunOS files and that the Valid tools and libraries reside in */home* or */home1* with a symbolic link from */usr/valid* to */home/valid* or */home1/valid*.

Table 2-3 shows Valid's disk partitioning defaults.

**Table 2-3. Valid Default Disk Partitioning on Sun Workstations**

For all systems (excluding Sun-4/60GX-P4)	
<i>/(root)</i>	16 Mb
<i>/swap</i>	40 Mb
<i>/usr</i>	65 Mb
<i>/home</i>	remainder of disk
<i>/home1</i>	second disk
For Sun-4/60GX-P4 (2 x 104 MB disks with floppy)	
<i>/(root)</i>	16 Mb
<i>/swap</i>	40 Mb
<i>/usr</i>	remainder of disk
<i>/home</i>	second disk

\* Note: */usr/valid* is linked to */home/valid* or */home1/valid* on all systems

## Valid Software on Standalone Systems

There are three practical options for a purely stand-alone system with no file systems mounted via NFS.

- Place both application software and libraries on one disk partition.
- Place application software on one disk partition and libraries on another partition.
- Place application software and selected libraries on one disk partition and the remaining libraries on another disk partition.

Wherever the applications and libraries are physically installed, the system must be set up to have them logically installed in the paths described previously in “Valid Software Organization.” Logical installation is done using entries in the */etc/fstab* file and symbolic links across file systems.

For single-disk systems, both the application software and the libraries are on the same disk. For two-disk systems, Valid recommends that software is installed in */home/valid* which is symbolically linked to */usr/valid*. If your libraries also fit on the first disk, put them there under */usr/valid/lib*. If you must split the libraries between two disks, you must create symbolic links for each library on the second disk back to */usr/valid/lib* on the first disk. Another option for two-disk systems is to partition the second disk into two or more file systems during setup, mount one partition on */usr/valid/lib*, and use the other partition(s) as user file systems.

## Valid Software on Servers

*For more information, see Creating the Links, p. 2-14.*

Many different configurations are possible if you have several systems linked with Ethernet, or servers with diskless clients.

If possible, install all of the Valid applications and libraries on a single server. Mount the appropriate tools directory (*/usr/valid/tools.sun3* or */usr/valid/tools.sun4*) from the server on each client as */usr/valid/tools.sunX* and make a local link to */usr/valid/tools* using NFS. Sun 3 and Sun 4 executables can reside on the same server as long as care is taken when mounting the clients (See Valid Software Installation for more information). Since the libraries are identical for both a Sun 3 and a Sun 4, */usr/valid/lib* can be mounted on either a Sun 3 or Sun 4.

If none of your systems has enough disk space to hold both the applications and libraries, you can install the applications in */usr/valid/tools* on one server and install the libraries in */usr/valid/lib* on another server. Clients would then mount both of these file systems in */usr/valid*. This arrangement might also make sense on a system with a small disk; you could install the applications on your local disk and mount the libraries from a server.

If you have a large number of workstations, you should consider setting up additional, duplicate servers for the Valid applications and libraries. This redundancy prevents problems on one server from stopping work on all of your workstations. While it is possible to divide the libraries among several servers, Valid recommends against this practice, as this can cause compatibility problems.

## Sun Software Installation

*If you are using a Versatec plotter, see Section 7 and the Appendices of this manual for plotter information.*

To install or upgrade your Sun system software, follow the instructions supplied by Sun. For a Sun3 or Sun4 system, Valid recommends installation of the SunOS4.X operating system. If a Sun3 client is served by a Sun4, they must both have SunOS4.X installed.

The Sunview User, Sunview Programmer, and Networking packages must be installed in order for the Valid graphics applications to run properly. Valid also recommends that you include the Install package in order that additional operating systems can be installed at a future date. (The `setup_exec` utility is required for all updates.)

## Valid Software Installation

*To determine if a disk partition is already mounted, use the **mount** command without options or arguments.*

All Valid application software and libraries are installed with *vscript*, the Valid software installation script.

Your Valid installation tape includes this script and two installation utilities, one for software installation on a Sun3 and one for software installation on a Sun4. System configuration information is prompted by *vscript* which then uses the appropriate utility to install the software. This section explains how to run *vscript* to install Valid application software and libraries for initial installations and software updates on standalone systems and single-server systems.

A server for the Valid applications must be installed before any of its clients can be configured. Do not install Valid applications software on a diskless system or any other system that does not have the applications on the local disk. Instead, mount */usr/valid* on the client using NFS.

When you install the Valid software, the disk partitions to contain the Valid software must be mounted by entries in the */etc/fstab* file.

## First-time Installations

The following procedures are for first-time installations only. (If you already have Valid System Tools on your system, move to the section *Installing the Valid Software*.)

Whenever Valid application packages and libraries are installed, the system must be set up to have them logically installed in the paths */usr/valid/tools* and */usr/valid/lib*. Logical installation is accomplished using symbolic links across file systems and with entries in the */etc/fstab* file. For first-time installations, Valid recommends that the applications be physically installed in */home/valid* with the appropriate links to */usr/valid*. See *Creating the Links* in this section for more information.

## Creating the Valid Directories

To install the Valid application software, create the directory */home/valid* and the sub-directories */home/valid/lib* and */home/valid/tools.sun3* and *tools.sun4* using the *mkdir* utility. Login as root and, at the UNIX prompt, type:

```
cd /home
mkdir valid
mkdir valid/lib
mkdir valid/tools.sun3
mkdir valid/tools.sun4
```

### Creating the Links

To run the Valid applications, you must create a symbolic link between */home/valid* and */usr/valid*, and a second link between */usr/valid/tools.sun3* or *tools.sun4* and */usr/valid/tools*.

To create these links, type:

```
ln -s /home/valid /usr/valid
ln -s /usr/valid/tools.sunX /usr/valid/tools
```

### Creating the Lib Account

Valid libraries are owned by the account “lib.” It is likely that a lib account does not exist on your system. If this is the case, create a “lib” account by adding entries to the password file */etc/passwd* and the group file */etc/group*.

To add a lib account, insert the following line to the password file anywhere above the plus symbol (+):

```
lib:*:8:8::/usr/valid/lib:
```

Next, edit the group file */etc/group* and add the following entry for lib:

```
lib:*:8:
```

If user ID 8 is already being used, choose any other value. However, the user ID for lib must be the same for every system on the network.

## Changing Permissions

The Valid directories will be owned by root. To change the ownership, group, and permissions of the Valid directories to *bin*, type:

```
chown bin valid
chgrp bin valid
chmod 755 valid
```

Change the ownership and group of the *tools* and *lib* directories by typing:

```
cd /home/valid
chown bin tools*
chgrp bin tools*
chown lib lib
chgrp lib lib
```

## Setting the PATH Variable

Once you have created the necessary directories and links, you must add */usr/valid/tools/bin* to the PATH variables in root's *.profile* and *.cshrc* files.

If you are using the Allegro (PCB) application, you must also add */usr/valid/tools/pcb/bin* to the PATH variables in root's *.profile* and *.cshrc* files.

Now that you have created the necessary directories and symbolic links, you are ready to begin the actual software installation.

## Installing the Valid Applications

The following procedures apply to both first-time Valid installations and to Valid software updates. Your installation tape includes the *vscript* installation script and two *vinstall* utilities, one for installation on Sun3 workstations and one for installation on Sun4 workstations.

Valid software is password protected to prevent unauthorized use. The individual passwords are keyed to the 8-digit host ID number of the machine chosen to be the Access server. Before proceeding with the installation, make sure that the server's host ID matches the number on the label of your installation tape. To determine your 8-digit host ID number for use during installation, type `host id` at the prompt.

If these numbers are different, quit the installation and call your Valid representative.

## Using vscrip

If you have not already done so, load the installation tape and log onto your system as root.

- 1 Change directories by typing:

```
cd /usr
```

- 2 The installation script and utilities are extracted from the installation tape using the `tar` command. This command automatically creates the directory `/usr/valid/install` if it does not already exist on your system.

To extract this information from a 1/4-inch cartridge tape, type:

```
tar xvpf /dev/rst0
```

To extract this information from 1/2-inch tape, type:

```
tar xvpf /dev/rmt0
```

- 3 Change directories by typing:

```
cd /usr/valid/install
```

- 4 Run the installation script by typing:

```
vscrip
```

You are then prompted for the following user-specific information:

What kind of tape are you installing:

- 1) 1/4" cartridge tape
- 2) 1/2" tape

What kind of executables are you installing?

- 1) SUN3
- 1) SUN4

Respond to both questions by entering the correct number and pressing **Return**. To quit the installation procedure at any time, type q.

*NOTE: If you are installing Valid applications in a mixed architecture environment, you must run vscript once for Sun3 executables and a second time for Sun4 executables.*

You are then prompted to insert the appropriate application tapes and to verify that you are ready to begin the installation process. A message appears to inform you that the tape is being positioned.

---

---

The following is a sample list of packages from Tape 1 of the Sun3 tape set. Depending upon your system, your selections may be different.

```
*****
1) Valid System Tools 3.0 SUN3-P4
2)  ged 9.1 SUN3-P2
3)  hardcopy 3.4 SUN3-P2
4)  stdlib 9.0 SUN3-P1
5)  libutil 9.0 SUN3-P3
6)  validcompiler 1.4 SUN3-P4
7)  valid packager 2.0 SUN3-P9
*****
```

You are asked to select the packages to be installed. Type a (for all) to install all of the applications included on the tape. (If you are only installing Valid System Tools, type "1".) The installation script verifies your selections and lists each package to be installed:

You have selected the following packages:

```
package 1
package 2
.
.
package n
```

Do you want to proceed?

Enter "y" or "n":

Type y. A series of status messages is displayed, and *vscript* asks if you want to continue. Type y. The following message appears:

```
Proceeding with the installation
- TAKE A BREAK
```

When all the selected packages have been installed, you are instructed to insert the next application tape, if necessary.

Extracting the software from tape for an initial installation requires approximately 30 minutes; a complete software applications update requires approximately 60 minutes. Library installation can require more than an hour depending on the size and number of the libraries being installed.

If problems occur during the installation, *vscript* usually describes the nature of the problem before exiting. Check the message file */tmp/vinstall.msg* for error messages if you run into problems. If the problems are severe and you need to call Valid for assistance, save and print a copy of this file for reference.

---

## Installing Sun 3 and Sun 4 Tools on a Server

Use *vscrip*t to install both the Sun3 and Sun4 packages if you are setting up a server to serve both Sun3 and Sun4 clients. To do so, complete the *vscrip*t procedure, selecting Sun 3 at the prompt. When *vscrip*t has completed the installation, run *vscrip*t a second time and select Sun 4 at the prompt.

Please note that the appropriate System Tools package must reside on the license server, regardless of the location of the Valid applications. For this reason, Valid System Tools must be installed for both Sun 3 and Sun 4 systems in a mixed architecture environment.

## Access Control

Access Control is an administrative tool that manages the security and the number of copies of an application running in a networked environment.

Access Control monitors the number of licenses purchased for a given design group and distributes them on demand. Access to an application is provided on a first-come/first-serve basis until the allowed number of copies of that application are in use. Please note that a specific number of licenses are needed to run each copy of a Valid application. For Sun3 workstations, four licenses are required for each application. For Sun4 workstations, five licenses are required for each application.

The following section explains how to install Access Control for both single server and multiple server configurations. For an overview of Access Control and information about fault tolerant mode, see *Administering Access Control*, Section 3.

---

## Configuring Servers

Once you have installed the Valid software on your system, the following procedures are used to configure Access Control. Servers are configured first, followed by the client systems for each server. Please note that configuring a standalone system is the same as configuring a single server.

Once the necessary directories and symbolic links have been created and the servers have been properly configured, the license manager daemon (*lmgrd*) can be invoked. During installation, the license file is installed in the directory *usr/valid/tools/adm*. The name of this file is "license" with an 8-digit hexadecimal extension that is the same as the host ID of the primary (first listed) license server.

Check to make sure that the entries in the license file conform to the following syntax, and that there is an entry for each server:

```
SERVER host_machine hostid 5200
```

If the name of the host machine is not entered correctly, edit this entry to include the correct name. If the license file does not exist after installation or if the host ID for the machine is incorrect, contact Valid.

*For more information concerning the license file, see Access Control Files, p. 3-11.*

In addition to entries for each server, the license file contains entries for the *valid1* and *valid2* daemons, and *FEATURE* entries for each package you have purchased. In each *FEATURE* entry, previous to the encrypted 16-character field, you will find a reference to the number of licenses purchased for that *FEATURE*.

In order to reserve the TCP/IP 5200 communication port for exclusive use of Valid's Access Control, Valid recommends the following procedures.

- 1** Login as root and edit the */etc/services* file.
- 2** Move the cursor to the end of the file and add the following line:

```
license          5200/tcp
```

For additional information concerning the */etc/services* file, please refer to the appropriate Sun documentation.

## Editing the Template Files

Several template files are created during installation. These template files contain examples of files required for Access Control to function properly.

Once the server has been configured, the license manager daemon *lmgrd*, and the *valid1* and *valid2* daemons are automatically started whenever the system is rebooted. In order for the license manager daemon to start automatically, a line must be added to the existing */etc/rc.local* file to read the special Valid startup file */etc/rc.valid*. This special file is created from a template file and modified to include the server's actual 8-digit host ID.

To setup the server to automatically startup the license daemons:

- 1 Change directories to */usr/valid/tools/adm* by typing:

```
cd /usr/valid/tools/adm
```

- 2 Copy and rename the *rc.local.template* file:

```
cp rc.local.template /etc/rc.valid
```

- 3 Change directories to */etc* and change permission of the new file by typing:

```
cd /etc
chmod 755 rc.valid
```

- 4 Using *vi*, edit the *rc.valid* file and replace "HOSTID" with the actual 8-digit host ID.

**5** Save the existing *rc.local* file by typing:

```
cp rc.local rc.local.old
```

**6** Edit the existing *rc.local* file and add the following line at the end of the file:

```
/etc/rc.valid
```

**7** Start the daemons by typing:

```
/etc/rc.valid
```

**8** Check to make sure that the *lmgrd* daemon is running by typing:

```
ps -aux.
```

If the process is not running, please refer to *Troubleshooting* in Section 3 for instructions on starting the daemons manually.

**NOTE**

If you are running a multiple license server configuration, the license manager daemon must be invoked (Step #7) at each server and run concurrently.

---

Once you have edited the *rc.local.template* file, you must also include the 8-digit host ID number of the Access server in your */usr/valid/tools/adm/servers* file.

- 1 Change directories and copy the *servers.template* file by typing:

```
cd /usr/valid/tools/adm
cp servers.template servers
```

- 2 Change permission of the new file by typing:

```
chmod 755 servers
```

- 3 You must now edit the new servers file and, for each client running Valid software, include the clientname and the correct path to the license file. This must also be done for a standalone system.

The *servers.template* file contains two entries, a template entry to be edited and an example. Once the file has been copied and the first line has been edited, the example line may be deleted from this file. Edit the first line to include the required user-specific information:

```
clientname /usr/valid/tools/adm/license.servershostid
```

If you are running Access Control in a mixed architecture environment, you must copy the edited *servers* file to update each server's *servers* file to the corresponding *adm* directory in the other tools directory (*tools.sun3* or *tools.sun4*).

You must now edit your */etc/exports* file:

- 1 Include the following line in the server's */etc/exports* file:

```
/usr/valid
```

- 2 Update the "exports" list with the new information included in the *exports* file by typing:

```
/etc/exportfs -a
```

If an *exports* file does not exist on your system, create an executable */etc/exports* file and include */usr/valid*. Then reboot your system in order to restart the daemons.

## Adding and Configuring Clients

In order to configure your system for multiple clients, you must mount the server's */usr/valid/tools.sunX* and */usr/valid/lib* directories with NFS entries in the client's */etc/fstab* file and include each of the clients in the server's *servers* file. To configure multiple clients:

- 1 Login as root and edit each client's */etc/fstab* file to include the following lines:

```
hostname: /usr/valid/tools.sunX /usr/valid/tools.sunX nfs ro, hard
hostname: /usr/valid/lib /usr/valid/lib nfs ro, hard
```

- 2 Edit the */usr/valid/tools/adm/servers* file of the server to include an entry for each client. Each entry must specify clientname, the correct path to the license file, and the license file extension number of the license server's host ID.
- 3 For each client, you must make a *tools* and *lib* directory. To make these directories, at the client machine, login as root and type:

```
cd /usr
mkdir valid
cd valid
mkdir tools.sunX
mkdir lib
```

- 4 Create a symbolic link between the tools directories by typing:

```
ln -s tools.sunX tools
```

- 5 Mount the server's */usr/valid/tools* and */usr/valid/lib* directories from the client by typing:

```
/etc/mount -a
```

*For more information about mounting file systems, see the Sun Beginner's Guide: Using the Network.*

Once you have created the necessary directories and properly mounted the server, you will be able to run the Valid applications on each client machine for which you have licenses available.

**NOTE**

In fault tolerant mode, should the mounted (active) license server become inaccessible, permission for continued use of Valid applications automatically transfers to a secondary (standby) license server. Please note that the new license server must be mounted in order to maintain access to the Valid application.

For information about Access Control's fault tolerant mode, see Section 3.

## Separate License and File Servers

Each client machine must be able to read the *license.HOSTID* file and the *servers* file located in the access license server's */usr/valid/tools.sun3/adm* or */usr/valid/tools.sun4/adm* directory.

If the client is mounting applications from a file server (or local disk) that is not the access license server, you must:

- 1 On the client, change directories by typing:

```
cd /usr/valid/tools
```

- 2 Create an *adm\_license* directory by typing:

```
mkdir adm_license
```

- 3 Edit the client's */etc/fstab* file and add the following line to mount the license server's *adm* directory on *adm\_license*:

```
license_server:/usr/valid/tools/adm /usr/valid/tools/adm_license nfs ro,hard
```

In the above line, *license\_server* is the hostname of your license server.

- 4 Mount the license server's *adm* directory by typing:

```
/etc/mount /usr/valid/tools/adm_license
```

- 5 On the client, change directories by typing:

```
cd /usr/valid/tools/adm
```

- 6 Create symbolic links between the *license.HOSTID* and *servers* files in the *adm\_license* directory and the *adm* directory by typing:

```
ln -s /usr/valid/tools/adm_license/license.HOSTID license.HOSTID
ln -s /usr/valid/tools/adm_license/servers servers
```

---

---

## Access Control Licenses

A specific number of licenses are needed to run each copy of a Valid application, four licenses for a Sun3 workstation and five licenses for a Sun4 workstation.

When all available licenses for a particular application are in use, or when too few licenses have been purchased to run a copy of a particular application, the following error message is displayed:

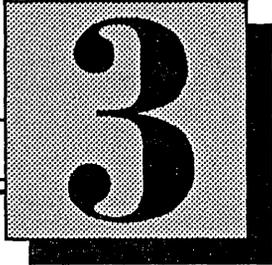
```
Sorry, no license available at this time for XXX.  
The MAXIMUM ALLOWABLE USERS ARE USING XXX  
Exiting.
```

Access Control automatically quits and returns you to the system prompt. For information about node-locked software or reserving licenses for specific users, see Section 3. To add more licenses, contact your Valid representative.

## **Completing the Installation**

Once installation is complete and Access Control is functioning, you are ready to create new user accounts and configure the system for plotting and printing.

See Section 4 to create accounts and Section 7 for information on plotting and printing.



# 3

## *Administering Access Control*

---

**T**his section provides information about the administration of Valid's Access Control. In particular, this section explains:

- Fault tolerant mode
- Access configuration options
- Reserving licenses
- Adding clients and servers
- Access Control Files and Utilities
- Troubleshooting Access problems

## Access Control Overview

Access Control manages the security and the number of copies of an application running in a networked environment. With Access Control, applications are distributed on a first-come, first-serve basis until the allowed number of licenses for the application are in use.

In the simplest multiple-client configuration, Access Control gives all users in a design group equal access to the available copies of an application. You can also configure Access Control to reserve licenses for certain users or machines. This section provides an overview of the Access Control tool in the fault tolerant mode.

## Starting the License Manager

Three Access Control daemons are invoked on license server machines and must be present for Access Control to function properly: the license manager daemon (*lmgrd*), and the *valid1* and *valid2* daemons. The *valid1* and *valid2* daemons distribute licenses to users of specific Valid applications and are managed by the license manager daemon. Please note that only one set of daemons (*lmgrd*, *valid1*, and *valid2*) can run on a license server at any given time.

Once the server machine has been configured, these daemons are automatically started whenever the system is rebooted. The procedure for invoking the daemons is explained in Section 2, *Configuring Servers*.

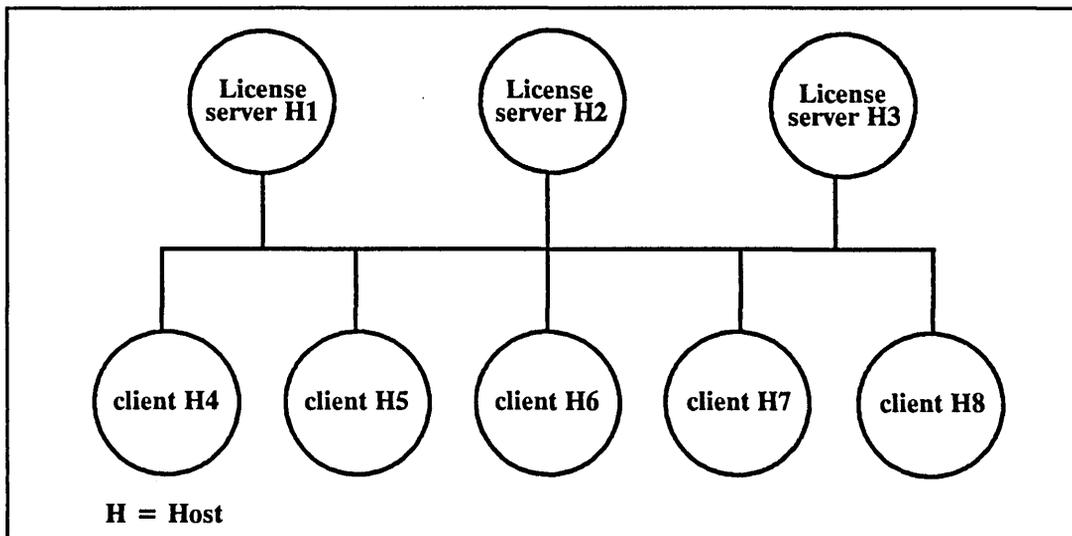
**The Simple Mode**

In the simple mode, one machine is a license server for a group of licenses. The simple mode is recommended for networks with only one file server. In this environment, the file server is configured as the license server.

**The Fault Tolerant Mode**

In the fault tolerant mode, multiple license servers are chosen to manage a group of licenses that is shared by all servers. With redundant file servers, fault tolerance provides continued access to applications when one of the license servers becomes dysfunctional. See Figure 3-1 for a diagram of a possible fault tolerant mode configuration.

In both Simple Mode and Fault Tolerant Mode with only one Valid file server, please note that the applications are unavailable whenever that server is down.



**Figure 3-1. Hypothetical Fault Tolerant Mode Configuration**

In a multiple server environment, users obtain licenses to use Valid applications while a simple majority of the license servers are running. For example, with three license manager daemons on three different servers, at least two servers must be functioning properly for licenses to be distributed to clients. With the hypothetical configuration in Figure 3-1, if server H1 crashes or is taken off-line, licenses continue to be issued from the remaining servers. However, if two out of the three servers become inaccessible, licenses are revoked.

For this reason, Valid recommends that you have an odd number of servers configured as license managers. Fault tolerance is enhanced when more than an even number of servers are available. (You do not gain fault tolerance by adding a fourth server to the configuration in Figure 3-1, because licenses would be unavailable with two servers down whether or not a fourth server was configured.) The following table lists optimal configuration for fault tolerance.

**Table 3-1. Fault Tolerance**

Number of Servers	Servers Down	Licenses Granted
3	1	Yes
3	2	No
4	1	Yes
4	2	No
5	1	Yes
5	2	Yes
5	3	No

## Reserving Licenses

It is possible to reserve licenses for specific machines or specific users. By reserving licenses, immediate access to Valid's application software is provided from the total pool of licenses purchased.

To reserve licenses for a specific user, you must create an "options" file and add a RESERVE line for each license to be reserved.

### NOTE

In order to run one copy of a Valid application, four licenses are required for Sun 3 workstations and five licenses are required for Sun 4 workstations. For this reason, the RESERVE line of the *options* file must be copied four times for a Sun3 and five times for a Sun4 to reserve the correct number of licenses.

- 1 Change directories to the active license server's */usr/valid/tools/adm* directory. Login as root on the license server and type:

```
cd /usr/valid/tools/adm
```

- 2 Copy the template options file:

```
cp options.template options
```

- 3 Edit the new */usr/valid/tools/adm/options* file and add the following line for each license you want to reserve, where *feature* is the application to be reserved and *name* is the specified user:

```
RESERVE feature USER name
```

EXAMPLE

For example, if you want to reserve a GED application for user Karen on a Sun3, enter the following line four times:

```
RESERVE ged.ex USER karen
```

- 4 To reserve a license for a specific machine, add the following line to the options file:

```
RESERVE feature HOST name
```

where *feature* is the application software to be reserved and *name* is the name of the machine for which you are reserving the license.

EXAMPLE

For example, if you want to reserve a GED application for a Sun4 named "sunlight," enter the following line five times:

```
RESERVE ged.ex HOST sunlight
```

- 5 If you are running Access Control in a mixed architecture environment, you must copy the edited *options* file on all servers and to the corresponding *ladm* directory in the other tools directory (*tools.sun3* or *tools.sun4*).

## Restarting the Daemons

Once you have copied and modified the */usr/valid/tools/adm/options* file to reserve the licenses, the *lmgrd*, *valid1* and *valid2* daemon entries in each license file must be modified and the *lmgrd* daemon must be restarted on each server.

Complete the following procedures on each license server to modify and restart the license daemons.

**1** Change directories by typing:

```
cd /usr/valid/tools/adm
```

**2** Edit the *license.HOSTID* file and modify the license file DAEMON entries to read:

```
DAEMON valid1 /usr/valid/tools/bin/valid1 /usr/valid/tools/adm/options
DAEMON valid2 /usr/valid/tools/bin/valid2 /usr/valid/tools/adm/options
```

**3** If you are running Access Control in a mixed architecture environment, copy the edited *license.HOSTID* file on all servers and to the corresponding *adm* directory in the other tools directory (*tools.sun3* or *tools.sun4*).

**4** Review the status of all Valid applications and determine if users are accessing a license with the *valid\_status* utility. To use this utility, type:

```
/usr/valid/tools/bin/valid_status -a
```

*Do NOT kill the license manager daemon while licenses are in use.*

All users presently accessing a Valid application are listed. Please notify each user that that you are terminating the license manager.

- 5** Kill the license manager daemon (*lmgrd*) using the command sequence

```
ps waux | grep lmgrd  
kill -9 pid
```

where *pid* is the process number for the *lmgrd* process.

- 6** Restart the license manager daemon (*lmgrd*) on each license server. Type:

```
/etc/rc.valid
```

- 7** Use the **ps waux** command to verify that the *lmgrd* process is running. If the */etc/rc.valid* file did not restart the license manager daemon, see *Troubleshooting* at the end of this section for instructions on restarting the license manager daemon manually.

## Adding Servers

In order to add an additional server to your existing environment or to update your application software, a new license file is required.

When you receive a new set of tapes from Valid, load the new installation tape and log in as root on the active license server.

- 1 Change directories to */usr*.

```
cd /usr
```

- 2 Extract the installation script and utilities from the installation tape with the **tar** command. To extract this information from a 1/4-inch cartridge tape, type:

```
tar xvfp /dev/rst0
```

To extract this information from 1/2-inch tape, type:

```
tar xvfp /dev/rmt0
```

- 3 Change directories by typing:

```
cd /usr/valid/install
```

- 4 Use the installation script to install the new license file. The **-p** option extracts only the license file:

```
vscript -p
```

- 5 If you are running Access Control in a mixed architecture environment, run **vscript -p** again for the second set of executables.

- 6 Edit the *license.HOSTID* file and check to make sure that there is an entry for each server and that each entry in the license file conforms to the following syntax

```
SERVER host_machine hostid 5200
```

where *host\_machine* is the name of the license server and *hostid* is the host ID of that machine.

- 7 Verify that the new license file is available in the */usr/valid/tools/adm* directory of each of the servers. In a mixed architecture environment, copy the new *license* file on each server and to the corresponding *adm* directory in the other tools directory (*tools.sun3* or *tools.sun4*).

If a separate installation tape for each license server was not provided, please contact your Valid representative.

- 8 Repeat Steps 5, 6, & 7 on page 3–8 to restart the license manager daemon (*lmgrd*) on each server.

A new *license.HOSTID* file must be received from Valid, and the previous procedure followed, in these situations:

- Adding license servers to your network
- Removing license servers from your network
- Changing server–client configuration
- Adding Valid applications to your file server

*Valid strongly recommends that all servers be reconfigured at the same time in order to avoid possible differences between new and old license file permissions.*

## Access Control Files

This section briefly describes the Access Control files used to configure your system: the *license.HOSTID* file, the *license.log* file, the *servers* file, and the *options* file.

### The License File

The license file is located in the */usr/valid/tools/adm* directory on the license server machine and contains the authorization codes and encryptions for all applications managed by Access Control.

```
SERVER sunrise a15328b3 5200
DAEMON valid1 /usr/valid/tools/bin/valid1 /usr/valid/tools/adm/options
DAEMON valid2 /usr/valid/tools/bin/valid2 /usr/valid/tools/adm/options
FEATURE actomd valid1 1.000 31-dec-99 500 EB53E499EB2E8CCFCF4A ""
FEATURE advantage valid2 1.000 31-dec-99 500 5B13B41975599E666D54 ""
FEATURE arouter valid1 1.000 31-dec-99 500 EB736449D18E43BFB268 ""
FEATURE calayV valid1 1.000 31-dec-99 500 FB2334191C9E5CD39430 ""
FEATURE cif2led.ex valid1 1.000 31-dec-99 500 CB23B40919DFCCDDE445 ""
FEATURE comp valid1 1.000 31-dec-99 500 6BF394C9ACF93C07B348 ""
```

Figure 3-2. Sample *license.HOSTID* file

#### NOTE

Be careful when modifying your license file. The server hostname and host ID on the SERVER line and the options file and daemon executables locations on the DAEMON lines are the only portions which you may alter. If other areas of the license file are changed or have become corrupted, the license manager daemon will not function correctly. This will prevent access to the Valid applications.

## Reinstalling the License File

If the *license* file is accidentally deleted or becomes corrupted, the Valid applications cannot run. To reinstall the *license* file from the installation tape, login as root on the active license server and use the *vscript* installation script as follows:

- 1 Change directories to */usr*. Type:

```
cd /usr
```

- 2 To extract the installation script from the tape, use the **tar** command. Extract this information from a 1/4-inch cartridge tape by typing:

```
tar xvp /dev/rst0
```

Extract this information from 1/2-inch tape by typing:

```
tar xvp /dev/rmt0
```

- 3 Change directories by typing:

```
cd /usr/valid/install
```

- 4 Use the installation script's *-p* option to install the new license file by typing:

```
vscript -p
```

- 5 If you are running Access Control in a mixed architecture environment, run `vscript -p` again for the second set of executables.

- 6 Repeat Steps 5, 6, & 7 on page 3-8 to kill and restart the license manager daemon (*lmgrd*) on each server.

## The license.log File

*You may want to periodically remove old files to save disk space.*

If you need to change the location of the license file, modify the */etc/rc.valid* file to include the new path to license file before restarting the license manager daemon (*lmgrd*). You must also edit all copies of the server's */usr/valid/tools/adm/servers* file and copies of the *rc.valid* file on all servers to reflect any change of license file location.

All access to the license server is logged in the */usr/tmp/license.log* file. Information recording checkouts, access denials, and network problems can be found in this file. A sample *license.log* file is shown in Figure 3-3.

Whenever the license server is rebooted, the *license.log* file is automatically renamed with a *.month.day* extension (for example: *license.log.Nov.28*) and a new *license.log* file is started. Please note that data accumulates in these files.

```
11/28 11:42 (license daemon) FLEXlm (v1.30) started on sunrise
11/28 11:42 (license daemon) CONNECTED, master is sunrise
11/28 11:42 (license daemon) Started valid1
11/28 11:42 (license daemon) Started valid2
11/28 11:42 (valid1) Server started on sunrise for: "verifier" "ucpfsim" "tvad"
"tscr.ex" "timemill" "thermostat" "tgilocal" "test_man" "swap" "stream2led.ex" "ucpsim"
"sim" "signoise" "rcfsim" "quanticout" "plot.ex.color" "placement" "pcb_review"
"pcb_prep" "pcb_engineer" "pcb_editor"
11/28 13:47 (valid1) OUT: gbom v1.000 by stu@zulu (4 licenses)
11/28 13:47 (valid1) IN: gbom by stu@zulu (4 licenses) (used: 0:00:00:01)
```

Figure 3-3. Sample *license.log.Nov.28* file

## The Servers File

The *servers* file must be present in the */usr/valid/tools/adm* directory to allow access to Valid's applications. The *servers* file must include the names of all clients running Valid applications and the path to the license file for each client.

Entries in this file must be edited to include the required user-specific information, according to the following syntax:

```
clientname /usr/valid/tools/adm/license.servershostid
```

Please note that if a client is not listed in the *servers* file, the default path */usr/valid/tools/adm/license.dat* is followed.

For procedures to create and modify your *servers* file, please refer to *Editing the Template Files* in Section 2.

## The Options File

The *options* file is located in the */usr/valid/tools/adm* directory on the license server and is used to reserve licenses for specific users or machines.

A sample options file, in which a license for Valid's Graphic Editor has been reserved for user Karen on a Sun4 and for a Sun3 workstation named "sunlight," is shown in Figure 3-4.

```
# Reserve ged.ex for "karen"
RESERVE ged.ex USER karen
# Reserve ged for the machine "sunrise"
RESERVE ged.ex HOST sunlight
RESERVE ged.ex HOST sunlight
RESERVE ged.ex HOST sunlight
RESERVE ged.ex HOST sunlight
```

**Figure 3-4.** Sample */usr/valid/tools/adm/options* file

To reserve use of a Valid application for a specific user or specific machine, see *Reserving Licenses*, page 3-5.

## Access Control Utilities

### The `valid_status` Utility

#### SYNTAX

```
valid_status -option
```

This section provides an overview of two Access Control utilities: `valid_status` and the `valid_report`.

The `valid_status` utility lists all current users of an application. It also lists the users waiting for license access and licenses that are specifically reserved. See Figure 3-5 for a sample `valid_status` display.

There are three options available, individually and in combination, to extract the information you need:

- `-a` lists all users for all applications running. Use this option for a complete listing.
- `-l path` specifies the pathname of a specific license file; without this option, `valid_status` defaults to the license file used by your machine. This option can be used in conjunction with the `-f` option.
- `-f feature` lists all users of a specific feature or Valid application. For a complete list of Valid Products and their corresponding feature names, see Appendix D.

```
Users for the feature ged.ex are:
karen at sunlight, started on Friday 12/15 at 11:26, 5 licenses

Users for the feature drc.ex are:
karen at sunlight, started on Friday 12/15 at 11:26, 5 licenses,
gary at sunstorm, started on Friday 12/15 at 11:28, 5 licenses
```

Figure 3-5. Sample `valid_status` display

## The `valid_report` Utility

### SYNTAX

The `valid_report` utility provides a complete account of users' access to the various applications. By default, a complete list of applications, usage, time of access, and CPU utilization is provided for the specified `license.log` file in `/usr/tmp`.

```
valid_report [-options] path_to_license.log
```

Specific reports, containing only requested information, can be generated using one or more of the following options:

- p** reports prime time usage between 8am – 5pm.
- C** suppresses counting of duplicate entries; if more than one copy of an application is running on a machine at the same time, this option only records one copy. Please note this option must be entered in *uppercase*.
- Y** displays the last timestamp message in the `license.log` file.
- u name** reports all usage for a specific user.
- h name** reports all usage for a specific host.
- f name** reports all usage for a specified feature (Valid application).
- P start\_time end\_time** reports usage between specified times. Please note that this option is designated in military format, and provides information only for hours specified.

Figure 3-6 shows a sample report using the `-f` option as requested in the following `valid_report` request:

**EXAMPLE**

```
valid_report -f ged.ex /usr/tmp/license.log
```

```
License server running from (12/15 11:23) to (12/15 11:38) 00:15:00
```

FEATURE "ged.ex" of valid2						
user name@host	Requests			Feature time		
	Granted	Denied	Queued	Running	/ Queued	
-----						
-----						
gary@sunstorm	15	0	0	00:37:08/	00:00:00	
karen@sunlight	5	0	0	00:00:02/	00:00:00	
-----						
-----						
Totals:	20	0	0	00:37:08/	00:00:00	
Averages:				00:01:51/	00:00:00	
Misc. info:						
	20% idle time	00:03:00				
	0% denied time	00:00:00				

**Figure 3-6. Report Generator Utility – sample report**

## Troubleshooting

If your Valid application malfunctions, investigate these possible problems with Access Control:

- ✓ Are there any error messages in the *license.log* file?

The */usr/tmp/license.log* file contains information concerning any errors logged by the *valid1*, *valid2*, and *lmgrd* daemons.

- ✓ Does the license server's *servers* file have a correct entry for the malfunctioning machine?

All entries should appear as:

*clientname path\_to\_license\_file.*

If the entries are correct, check all mounts and links as described in Section 2.

- ✓ Is the license file located under */usr/valid/tools/adm* and does the license server daemon know this location?

Use the **ps waux** command to locate a *lmgrd -c license\_file\_path* process where *license\_file\_path* is the correct pathname to your *license.HOSTID* file.

- ✓ Is TCP/IP functioning properly?

Refer to your Sun documentation for debugging information.

- ✓ Is the application (for example, GED) available for access to an inadequate number of licenses? Use the *valid\_status* utility to determine license status.

✓ Is your machine able to communicate with the active license server?

To determine this, access the license server by typing `/usr/etc/ping hostname` where *hostname* is the name of the license server. If the two machines are communicating, you will see a message similar to: `xxx is alive`. If the machines are not communicating, this message appears: `no answer from xxx (Or /usr/etc/ping: unknown host xxx)`.

✓ Do you have the environment variable `LM_LICENSE_FILE` set? This may indicate possible conflicts between complimentary security systems.

To determine if the environment variable `LM_LICENSE_FILE` is set, type:

```
printenv | grep LM_LICENSE_FILE
```

If the environment variable is set, the output is: `LM_LICENSE_FILE = path_to_license_file`. When you have determined that this file is set, contact your Valid representative for instructions.

✓ Is there more than one license file indicated during the vscrip installation process?

If more than one `license.HOSTID` file exists on the installation tape, you are prompted to select the correct file for installation.

*Please note that in both examples below, the backslash symbol (\) is included in the text to indicate a continuous command line. This symbol is not entered in the command itself.*

✓ Are the *valid1*, *valid2*, and license manager (*lmgrd*) daemon running?

Use the **ps waux** command to check all running processes. If *valid1* or *valid2* is not running, restart *lmgrd* as described in the *Configuring Servers*, Section 2.

✓ Are the Valid applications inaccessible following the execution of */etc/rc.valid*?

You can restart the license manager daemon manually by entering the following commands. If you are running a multiple license server configuration, this command must be entered at each server and run concurrently.

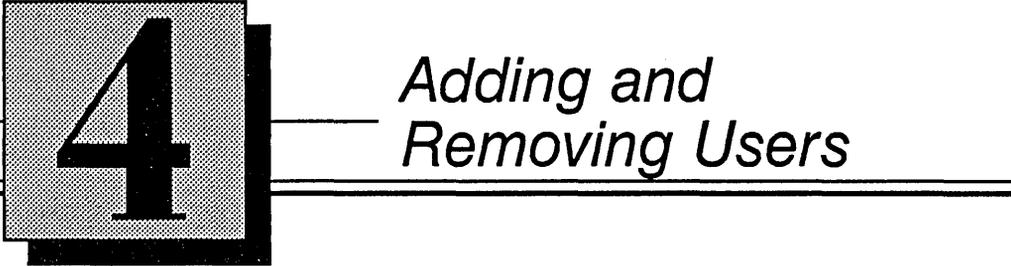
In a C shell, start the *lmgrd* daemon with the following command and substitute “HOSTID” with the 8-digit license file extension number:

```
/usr/valid/tools/bin/lmgrd -c /usr/valid/tools/adm/license.HOSTID >& \
/usr/tmp/license.log &
```

In a Bourne shell, start the *lmgrd* daemon with the following command and substitute “HOSTID” with the 8-digit license file number:

```
/usr/valid/tools/bin/lmgrd -c /usr/valid/tools/adm/license.HOSTID \
1>/usr/tmp/license.log 2>/usr/tmp/license.log &
```





## *Adding and Removing Users*

**T**his section explains how to add and remove users for the Valid software on your Sun workstation. Some of the topics discussed are:

- Creating new user accounts
- Assigning user passwords
- Deleting accounts
- Understanding default user files

## Creating User Accounts

Only the super-user (root) can create accounts for new users, and each account should be created on the machine where the user normally logs in. If you are running the Yellow Pages, always create the new account on the Yellow Pages master server.

Valid provides the *mkusr* utility to automate the creation of new accounts. *Mkusr* adds entries to the password and group files, creates the user's login directory, and then adds a set of default files for UNIX and Valid applications software to the user's login directory.

## System Requirements for Mkusr

The Valid software must be installed on and accessible from the system where you run *mkusr*.

To run *mkusr* on a Yellow Pages server, the Valid software must also be installed on the client system and the user's home directory must be located on the client system.

In this environment, Ethernet must be installed and running between the client and server, and the Yellow Pages must be running on both client and server. In addition, the client's */etc/hosts* file must contain entries for the server so that the server can create the user's directory on the client.

## Running mkusr

Follow this procedure to create a user account:

**1** Log in as root.

**2** Type:

```
mkusr
```

The system prompts for the additional user information including the user name, group name, the person's full name, the filesystem for the user's home directory and the default shell (Bourne or C).

**3** Enter a name for the user, such as "stuart." If you enter a name in all capital letters, UNIX assumes that the user's terminal can only display all caps, which can be annoying. To avoid this problem, use lowercase or mixed-case login names.

**4** Enter the name of the project group for the user if you are using project groups (see *Design File Management* in Section 6.) The default project group name is *user*.

**5** Enter the person's full name, such as "Stuart Jones."

*Press Ctrl-D to exit from mkusr without creating a new account.*

**6** The default file system is */home*. Press  if you want the user's home directory to be on the */home* file system: */home/stuart*. Enter the name of a file system (for example, "*/home1/stuart*") if you want the home directory located elsewhere.

**7** Select the user's UNIX shell. The Bourne shell is the default, and slightly faster, but the C shell is more versatile. Type B if you want the Bourne Shell. Type C if you want the C shell.

*Mkusr* repeats the information you enter, and displays the user and group ID numbers that the system creates. The user ID number is a unique number for each user, and the group ID number is an encoded number for the group name entry.

**8** To add the user account, type:

y

If you enter *y*, *mkusr* edits the password file (*/etc/passwd*) and group file (*/etc/group*), and adds the new information.

If there is an error in the user information displayed, type:

n

If you enter *n*, the system prompts you to re-enter the user information.

Next, *mkusr* copies the contents of the */usr/valid/tools/adm/ALL* directory to the new user's directory. This directory contains a set of user startup files (*.profile*, *.login*, and *.cshrc*), a *startup.ged* file, and a set of SCALD command (or *directive*) files for running the other SCALD analysis tools. These files are described in the next part of this section.

After you create a new account, update the Yellow Pages password and group databases. Run *mkusr* on the Yellow Pages server if you use the Yellow Pages.

If you create the user's password on the Yellow Pages server, but put the user's home directory on a client system, the user cannot log in to that client system unless the server and the Yellow Pages are running. To avoid this, you may want to create an entry in the client's own password file. You can do this on the client with the Valid utility *mknewpwenry*.

*Mkusr* uses *mknewpwenry* to create passwords. Simply log in as root and enter the command:

```
mknewpwenry
```

You are prompted for all of the necessary information, as shown previously for *mkusr*.

If you try to create a directory on a client system but the client is down, *mkusr* pauses and eventually times out with an error message. If this happens, when the client comes back up, log on to the client and run the utility that creates user directories, *mkscaldusr*.

## Problems Running mkusr

SYNTAX

***mkscaldusr username owner group dirname***

The *owner* is the owner of the directory, almost always the same as the *username*. For example:

```
mkscaldusr joe joe user /home/joe
```

Alternatively, you can manually delete the user's entries from the server's password and group files, and then re-run *mkusr* on the server.

If a user cannot log on, be sure that the Yellow Pages password and group databases have been updated on the Yellow Pages server.

If you make a user on a diskless client, you may need to add the user's home directory to the server's */etc/exports* file to allow the user to access the directory. Also, the user's directory (or the file system containing it) must, in general, be mounted on the client with write permission.

## Accounts Not Created with *mkusr*

*Mkusr* chooses user ID numbers based on login names. If you have two users on your Ethernet with the same login, they will receive identical user ID numbers. Consequently, even if they have accounts on different systems, they may be able to gain access to each other's files without knowing the other person's password. This may be a problem if you are in a security-conscious environment. To fix the problem, change one of the user's ID numbers by editing one user's password file entry on the Yellow Pages server. If the user also has a password file entry on the user's own system, change that password file entry as well.

Users with accounts not created by *mkusr*, especially root, must add the directory */usr/valid/tools/bin* to their search path in order to use the Valid software. This requires changes to the *.login* and *.cshrc* files (C shell) or to the *.profile* file (Bourne shell). Examples of these files containing the correct path can be found in the directory */usr/valid/tools/adm/ALL*.

Copy the Valid SCALD software startup files (*startup.ged*, etc.) in */usr/valid/tools/adm/ALL* to the user's home directory. Be careful not to overwrite the user's existing *.login*, *.cshrc*, or *.profile* files with the generic versions in the ALL directory, since the user may have customized them. Make sure that each user's path variable gets changed, either by you or by the user.

## User Passwords

New users should set their passwords with the **passwd** command and/or the **yppasswd** command (if you are running the Yellow Pages) the first time they log in.

The exact procedure depends upon whether you are running the Yellow Pages and whether the user has a password file entry on the YP client system as well as on the YP server.

The procedure each user should follow is:

- 1** Log in to the YP server or to a client and enter the **yppasswd** command:

```
yppasswd
```

- 2** The system prompts for a password. Type in a password. The password is not echoed on the screen.
- 3** The system asks the user to repeat the password to guard against typing errors. Retype the password. The system then updates the Yellow Pages server's password database with the encrypted version of the new password.

If the user also has a local password file entry on a YP client, the user should log in to that client and enter the command:

```
passwd
```

The system prompts for a password. (Users can have different passwords on their home system and on the rest of the network.) Type a password at the prompt, and repeat it when prompted to do so.

Users can change their passwords with the *passwd* and *yppasswd* commands whenever they like.

If there is no password entry for a user's account, the system does not demand a password when someone logs in with that user name.

## Forgotten Passwords

Root can change any user's password with the command:

**`passwd username`**

When users forget their password, you can give them a new one with the `passwd` command. Since the `yp-passwd` command does not let root change other user's passwords unless root knows their current password, you'll need to use the `passwd` command on the YP server and then update the YP password database.

If you forget the root password, `rlogin` from a machine listed in your `.rhosts` file and change the root password to one you can remember.

## Removing Users

Follow this procedure to remove a user from the system:

- 1** Delete the line containing the user's name from the */etc/passwd* file on the Yellow Pages server and on any clients where the user has password file entries.
- 2** Delete the user's name and the trailing comma from the */etc/group* file on the YP server and all clients where the user had password entries.
- 3** Rebuild the YP password and group databases on the YP server.
- 4** Archive the user's directory to tape.
- 5** After creating the tape, determine which of the user's files you should keep on hand, and copy them to other directories.
- 6** Delete the user's login directory with the **rm** command using the **-r** option. For example:

```
cd /home
pwd    check that you're in the right place
rm -r suzy
```

### NOTE

The **-r** argument to *rm* recursively removes */home/suzy* and all the files and directories it contains. **Be extremely cautious with the *rm -r* command.** You don't want to accidentally remove everything in */home!*

## Default User Files

This section describes the default files that *mkusr* puts into each new user's home (login) directories.

- *.login*
- *.cshrc*
- *.profile*
- *compiler.cmd*
- *packager.cmd*
- *simulate.cmd*
- *startup.ged*
- *td.cmd*
- *verifier.cmd*
- *master.local*

The **ls** command does not list the files that begin with a dot (.) unless you use the **-a** flag.

All of these files except for *.login*, *.profile*, and *.cshrc* are for the Valid SCALD applications tools, and are described in the various SCALD reference manuals. These three "dot" files customize the user's UNIX environment when logging in or starting up a shell.

---

## The .login File

The system reads a user's *.login* file whenever the user logs in under the C shell. The file contains commands that set up the shell environment for the user. The default *.login* file is shown in Figure 4-1.

```
# rcsid[]="$Header: .login,v 800.1 85/09/12 15:05:23 root Exp $";  
# sccsid[]="%W% %Y% %Q% %C%";  
setenv MAIL /usr/spool/mail/$USER
```

Figure 4-1. Default .login File

The first two lines are comments that identify the revision level of the file.

The *setenv* command sets the MAIL variable that determines the file in which the user's mail is spooled.

## The .cshrc File

The system reads the *.cshrc* file whenever a C shell is started for the user. Unlike the *.login* file, which is read only once (when the user logs in), the *.cshrc* file is read each time a sub-process is started. When you log in, the system reads the *.cshrc* file first, and then the *.login* file. Consequently, the value set for *PATH* (or any other variable) in the *.login* file overrides the value in *.cshrc*. Figure 4-2 shows the default *.cshrc* file.

```
# rcsid[]="$Header: .cshrc,v 800.1 85/09/12 15:05:19 root Exp $";
# sccsid[]="%W% %Y% %Q% %G%";
set path=(~/bin /usr/local /usr/ucb /usr/bin /bin /usr/valid/tools/bin .)
set mail=/usr/spool/mail/$USER
if ($?prompt) then
    # this is login shell
    set ignoreeof
    set history=20
endif
```

Figure 4-2. Default *.cshrc* File

The first *set* command sets the user's search path. The second *set* command specifies the file in which the user's mail is spooled.

The *if ... endif* checks whether the variable "prompt" is set. If it is, the login shell is being initialized and the two *set* commands are executed.

The variable “ignoreeof” is set to cause the system to ignore end of file characters (**Control**-D) so that pressing **Control**-D does not log the user out.

The value of “history” is set to 20. The C shell history mechanism keeps track of the last 20 commands this user has entered, and allows the user to repeat them with or without modifications. See the *cs*h man page for more information.

## The .profile File

The system reads the *.profile* file in a user's top directory whenever the user logs in under the Bourne shell. Figure 4-3 shows the default *.profile* file.

```
# rcsid["$Header: .profile,v 800.1 85/09/12 15:05:27 root Exp $"];
# sccsid["$W% %Y% %Q% %G%"];
MAIL=/usr/spool/mail/$USER
PATH=:/usr/$USER/bin:/usr/local:/usr/valid/tools/bin:/usr/ucb:/usr/bin:/bin
PS1="% "
export HOME USER PATH MAIL TERM PS1
```

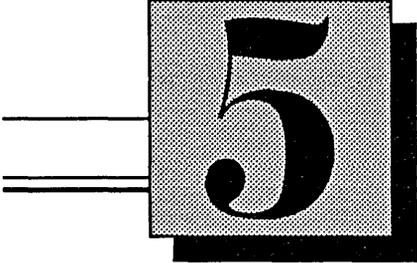
Figure 4-3. Default .profile File

The *PATH* *must* be on a single line.

The *MAIL* and *PATH* variables are set as explained for the *.cshrc* file.

The *PS1* variable, the primary prompt, is set to display '*%*'.

The *export* command makes the *HOME*, *USER*, *PATH*, *MAIL*, *TERM*, and *PS1* variables global to the user's shell and available to any program or command the user executes.



# 5

## *Backups*

---

**T**his section reviews procedures for backing up and restoring design data.

The most valuable part of your Sun Workstation is the data created by your users. To facilitate the task of maintaining backup copies of their design data, this chapter reviews your standard backup utilities.

This material is intended to be a supplement to the Backup and Restore sections of *System Administration for the Sun Workstation* that came with your workstation.

The exact procedure for backup depends upon several factors at your site, including the number of systems involved, the responsibilities of your System Manager, and your site's policy regarding backups. Choose the procedure that works best for you, but be sure to check that you can actually restore from the backup tapes you make.

## Utilities for Backups: Overview

UNIX provides several utilities that can be used for backing up and archiving files. For specific information on command syntax, see the appropriate UNIX manual page. In addition, Valid supplies three utilities: *backup* (8V), *custombkup* (8V), and *cronbkup* (8V). Below are brief descriptions of these utilities.

### **tar: tape archiver**

This is the easiest backup utility to use. Incremental backups are difficult with *tar*, but if you want a full backup of a small user file system or a design directory, or if you just want to backup a few specific files, *tar* is very convenient. *Tar* is also the best utility for users who want to make backup tapes themselves. Because *tar* does not support multi-volume tapes, backups of large files or file systems must fit on a single tape. *Tar* is the most portable backup UNIX utility. A user has the best chance of moving a file between different vendors, such as SunOS, SCALD S320, and ULTRIX with the *tar* format.

### **cpio: copy input to output**

This utility is good for incremental backups of design directories or file systems. The syntax is complex, but useful.

### **custombkup**

This utility is an interactive front end to *cpio* that creates full or incremental back-ups of directory trees. Use *custombkup* to backup part of a file system.

### **cronbkup**

This utility is a script file that performs automatic *cpio* backups of directory trees to a remote server-disk. *Cronbkup* is executed by the *cron* daemon from the *crontab* file.

**dump**

This utility creates full or incremental dumps of entire file systems. *Dump* supports multi-volume tapes. Use *restore* to read a *dump* tape.

**backup**

This utility is an interactive, front end for *dump* that backs up every file system on each of your system's disks.

**dd**

This is the generic UNIX copying utility. It can be used to read and write tapes for non-UNIX systems. For example, the block size and the order of data in bytes may be different, or the other system may use the EBCDIC rather than the ASCII data format. The *dd* utility can handle these types of conversions. However, the *dd* utility is not very convenient for making system backups, and, like *tar*, is limited to a single tape.

## Sample Backup Session

Figure 4-1 is an example of a session using *backup*.

```
Do you want to back up to:
  1) a local tape device
  2) a remote tape device
  3) a file in a remote file system which is currently mounted via NFS
[local|remote|file]: local
Is UNIX running single user? [y|n]: y
Do you wish to check the file systems? [y/n]: y
  {normal fsck messages}

The "dump" program with the "9u" option will now be used to back up all
the files created or changed since the last daily, weekly, or monthly
backup.

Daily backup (dump 9u) of /dev/rsd0g (/usr)
-----

What is the path name of the file you want to backup to?
/nfs/sudan/mnt/backups/kenya/dailyusrmonday
Type <ctrl>D when tape is loaded and on line
^D

DUMP:   Date of this level 9 dump: Mon Apr 20 17:39:42
DUMP:   Date of last level 0 dump: the epoch
DUMP:   Dumping /dev/rsd0g (/usr) to
/nfs/sudan/mnt/backups/kenya/dailyusrmonday
DUMP:   mapping (Pass I) [regular files]
DUMP:   mapping (Pass II) [directories]
DUMP:   estimated 6620 tape blocks on 0.34 tape(s).
DUMP:   dumping (Pass III) [directories]
DUMP:   dumping (Pass IV) [regular files] . . .
DUMP:   estimated 49437 tape blocks on 0.14 tape(s).
```

Figure 4-1. Daily Backup to Remotely-Mounted System

## Restoring Individual Files

The easiest way to restore a few files is to use *restore* with the *i* flag. This flag tells *restore* to read the directory information on the tape (or on the first tape, if it is a multi-volume backup) and then create a shell-like environment in which you can examine the directories and files. See *restore* in the *UNIX Reference Pages* for more information.

The basic procedure to restore individual files follows.

- 1 Bring the system down to single-user mode and run a file system check.
- 2 Mount the tape for the file system that contains the file you want. If the file system spans several tapes (volumes), mount the first tape (volume 1).
- 3 Use *cd*(1) to change the working directory to be the same as that of the mounted file system (for example: *cd /usr*).
- 4 Enter the appropriate command:

**restore iv**

*{For local 1/4 inch cartridge tape}*

**restore ivf [host:]/dev/rmt0**

*{For 1/2 inch remote tape device}*

**restore ivf filename**

*{For remote dump file mounted via NFS}*

The **l** flag sets up the interactive shell, and the **v** flag tells *restore* to be verbose. The **f** flag says to use an input device other than the default, */dev/rst0*. The optional host argument reads a tape in a remote system's tape drive; your system must be in the */.rhosts* file on the remote system.

- 5 The system displays a series of messages, reads the directory information, and then displays a prompt:

```
verify tape and initialize maps
Dump date:      (date tape was made)
Dumped from:   (date of previous dump)
Extract directories from tape
Initialize symbol table

restore>
```

- 6 From the *restore>* prompt, you can use the *cd*, *pwd*, and *ls* commands to move through the directory tree and list the names of the files. While in this shell-like environment, */* is the top directory of the file system you backed up (for example, */usr/joe* becomes */joe*).

- 7 Use the *add* command to select a file for extraction:

```
restore> add filename filename2...
```

The selected file is marked for extraction.

- 8 After selecting the files and directories you want, use the *extract* command to extract the files from the tape:

```
restore> extract
```

The system responds:

```
Extract requested files
You have not read any tapes
yet. Unless you know which
volume your file(s) are on,
you should start with the
last volume and work towards
the first.
```

```
Specify next volume #:
```

- 9 Enter the tape number. If there is only one tape in the backup, type:

```
1
```

If there are several tapes and you don't know which tape contains the file(s), load the last volume into the tape drive and then enter its number.

---

---

The system extracts the files. When finished, the system prompts:

Set directory mode, owner,  
and times. Set owner/mode  
for `.` [yn]:

- 10 Answer “y” if you want to restore the original ownership and permissions. Answer “n” if you want the restored files to be owned (and have the default permissions of) the person running restore.
- 11 The `restore>` prompt reappears. You can select more files or exit from *restore* with the *quit* command.

## Restoring Entire File Systems

To recover a lost user file system, you must rebuild the file system using *newfs* and then restore the files onto the clean file system. Even if the file system you are restoring has not been totally destroyed, you should use *newfs* to get a clean disk with contiguous files.

**Do not restore a file system unless you are running in single-user mode.**

Restore your full backup first, and then each necessary incremental backup, working from the oldest tape to the newest. Valid recommends the following backup procedures: first restore the most recent full (monthly) backup, then the current weekly backup, and finally, each of the daily backups, beginning with the oldest.

### NOTE

Everything in the file system is irretrievably deleted when you run *newfs*. Always backup the entire file system with *dump* or *rdump* before rebuilding it.

---

To restore a full file system, follow this procedure:

**1** Bring the system down to single-user mode. Halt all diskless clients if the system is a Network Disk server. You can do this by rebooting the server.

**2** Rebuild the file system with the *newfs* command:

```
/etc/newfs /dev/rdevice
```

where *device* is the name of the disk partition (for example, *xy2h*). *Newfs* is a front-end to *mkfs* that uses disk partition information stored in the file */etc/disktab*. This command essentially says, "erase this file system and make a new file system with all blocks free."

**3** After running *newfs*, check the new file system:

```
/etc/fsck -y /dev/rdevice
```

**4** Mount the file system:

```
/etc/mount /dev/device filesystem_name
```

For example:

```
/etc/mount /dev/device /home
```

**5** Change directory to the top directory of the file system:

```
cd /home
```

**6** Load the first volume (tape 1) of the most recent full (monthly) backup into the tape drive.

**7** Restore the file system with the appropriate command:

**restore rv**

*{For local 1/4 inch cartridge tape}*

**restore rvf [host:]/dev/rmt0**

*{For 1/2 inch remote tape device}*

**restore rvf filename**

*{For remote file mounted via NFS}*

The system asks you:

Set directory mode, owner,  
and times. Set owner/mode  
for `.` [yn]:

**8** Answer “y” if you want to restore the original ownerships and permissions. Answer “n” if you want the restored files to be owned (and have the default permissions of) the person running restore. You probably want to restore the original information.

**9** After restoring all volumes of the monthly backup, repeat the *restore* command for the most recent weekly backup, and again for each of the daily backups, beginning with the oldest.

*The -a option unmounts  
all files.*

**10** Change to the root directory:

```
cd /
```

**11** Dismount the file system:

```
/etc/umount -a
```

**12** Run a file system check on the restored file system:

```
/etc/fsck /dev/name
```

**13** Bring the system backup to multi-user mode by entering **Control**-D.

This should leave you with a restored file system that is complete to the time of your last backup.

## The tar Utility

*Tar* is the tape archive utility for writing multiple files or directories to a single magnetic tape or restoring multiple files from a single magnetic tape. The *tar* utility operates on individual files and directories rather than entire file systems and can be used both to read and to write tapes. *Tar* recursively follows directory trees to the bottom of the directory hierarchy. It can therefore be used to archive entire directory trees. *Tar* is relatively simple and has a number of uses:

- *Tar* is used for backing up user files and application programs when updating your system to a new release.
- *Tar* is used for backups of individual files and directories.
- Designers who want to make a backup tape for themselves often use *tar*.
- Designers who need to maintain their own tape archives often use *tar*.

*Tar* is not very useful for incremental backups because:

- There is no way to tell *tar* not to overwrite files that have been changed since the tape was created.
- Multi-volume archives are impossible to create because *tar* does not recognize the end of the tape.
- *Tar* is relatively slow for archiving or restoring large numbers of files.

---

## Using tar

The *tar* command has several options, or “flags.”

- c** creates a new archive on tape or write tapes. This flag will cause *tar* to write at the beginning of the tape. Please note that the **c** option will overwrite any previous files that may exist on the tape.
- x** extracts files or directories from tapes.
- v** tells *tar* to be verbose. This means that *tar* will print the name of each file that it reads or writes, together with the file’s size in bytes. By default *tar* does not tell you which files it has selected.
- t** displays the names and other information on the tape without loading the files. The user is encouraged to check the contents of a tape before loading it in order to avoid overwriting a more current version of a file.
- f** specifies the name of the archive tape device instead of the default */dev/rst0* and is used in most situations requiring *tar*.

The example below creates an archive to the default tape drive (*/dev/rst0*):

```
tar cv filename
```

To append the file or files to the end of an existing archive tape, use the **r** option, as follows:

```
tar rv filename
```

If you give *tar* a full pathname for a file or directory as you create the tape, that file or directory is restored to its original directory regardless of the directory location when you read the tape. This kind of tape is known as a “rooted” *tar* tape.

The example below creates a rooted tape:

```
tar cvf /dev/rst0 /home/sue/proj1
```

If you want to *tar* several files on one tape, you must use the device file *nrst0* as the device. This file stops the tape at the end of the *tar* cycle without rewinding the tape, and is used as follows:

```
tar cvf /dev/nrst0
```

Often you do not want to restore files to their original directories. For example, when archiving a design directory, you may want to restore it to a different directory later. To create an “unrooted” tape, change to the directory containing the files or directories you want to archive and use a period (.) to represent the pathname. To create an “unrooted” tape, the *tar* command is entered as follows:

```
cd /home/stuart/project1
tar cvf /dev/rst0 .
```

*Using a period (.) after the tar command places the restored files in your present directory location.*

---

The period (.) tells *tar* to archive both the current directory and each of its files and subdirectories. The advantage of an unrooted tape is that you can restore the files to any directory you choose. The disadvantage is that, if you do not move to the correct directory before reading the tape, you might restore the files in the wrong directory.

The following example shows how *tar* is used to read the contents of an unrooted tape into a user's directory.

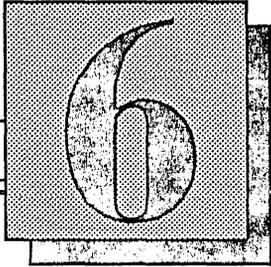
```
cd /home/stuart/project1
tar xvf /dev/rst0
```

**NOTE**

When unrooted files are extracted, file ownership is automatically transferred to the directory's owner. This can be helpful when you want to change the ownership of a directory tree. Rooted files retain their original ownership. In order to retain the original ownerships and protections, use the **p** flag.

*Tar* works with file systems rather than the raw disk. Accordingly, *tar* is significantly slower than the *dump* and *restore* utilities. Also, the messages printed by the **v** flag further decrease *tar*'s speed.





## *Design File Management*

---

This section describes the files and directories that make up a Valid SCALD design project on a Sun Workstation. In particular, this section explains:

- Organizing files into design directories
- Protecting designs using UNIX file permissions
- Using SCALD directories and SCALD libraries
- Maintaining the user file systems
- Archiving a design

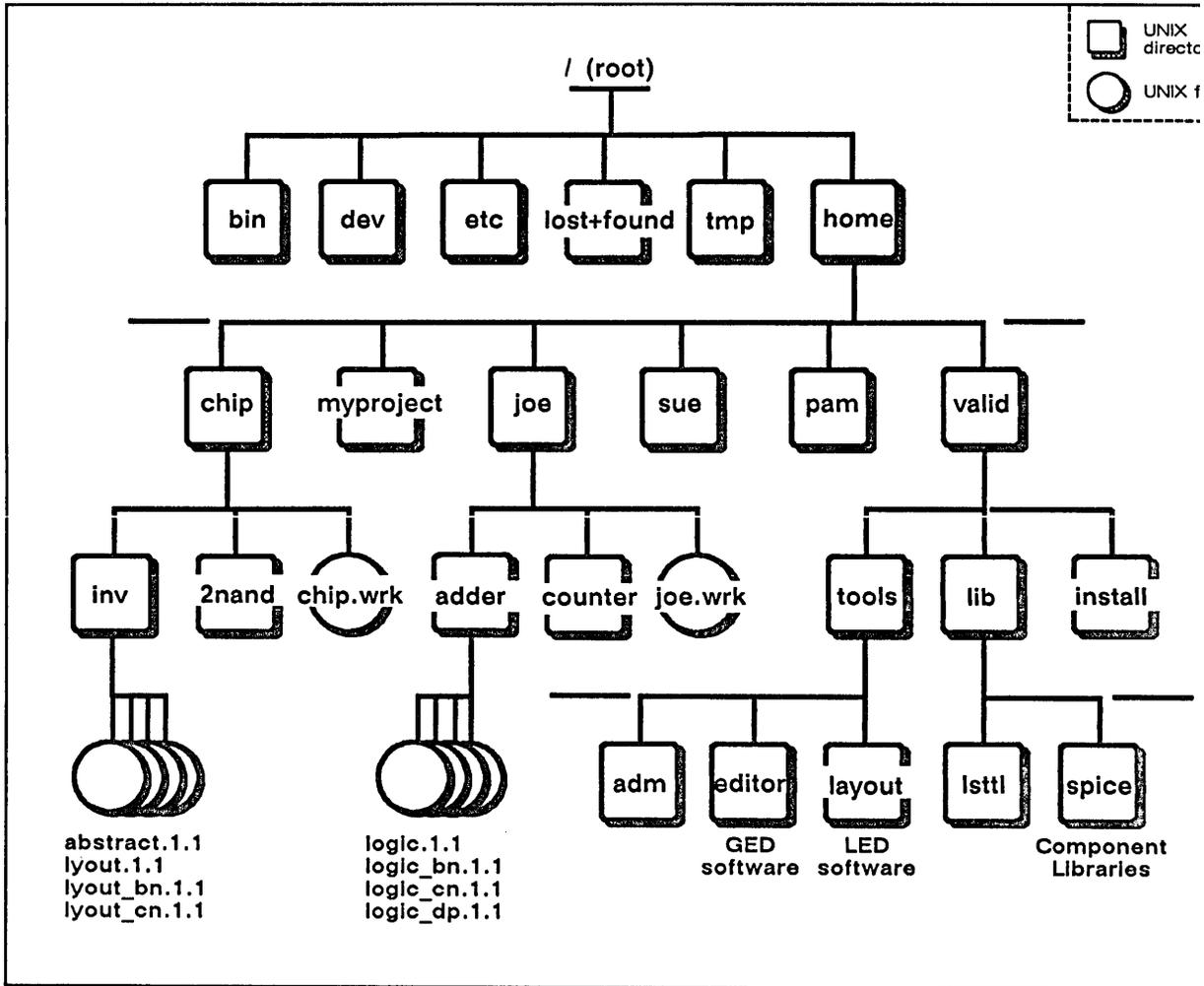


Figure 6-1. Directory Hierarchy

Figure 6-1 shows where the Valid software and libraries are located, and how design project directories are arranged.

## Design Directories

For more information on file permissions, see File Permissions, page 6-10.

The files used in a design need to be organized in order that users can easily find the files they need, have permission to change the appropriate files, and are denied permission for files they shouldn't change. For these reasons, it is best to keep all of the drawings that make up a design in a single *design directory*. This directory can be the login directory for an account you create for the design. The sub-directories that GED creates in a design directory, which contain the files for individual drawings, are called *drawing directories*. Figure 6-2 displays a sample design directory.

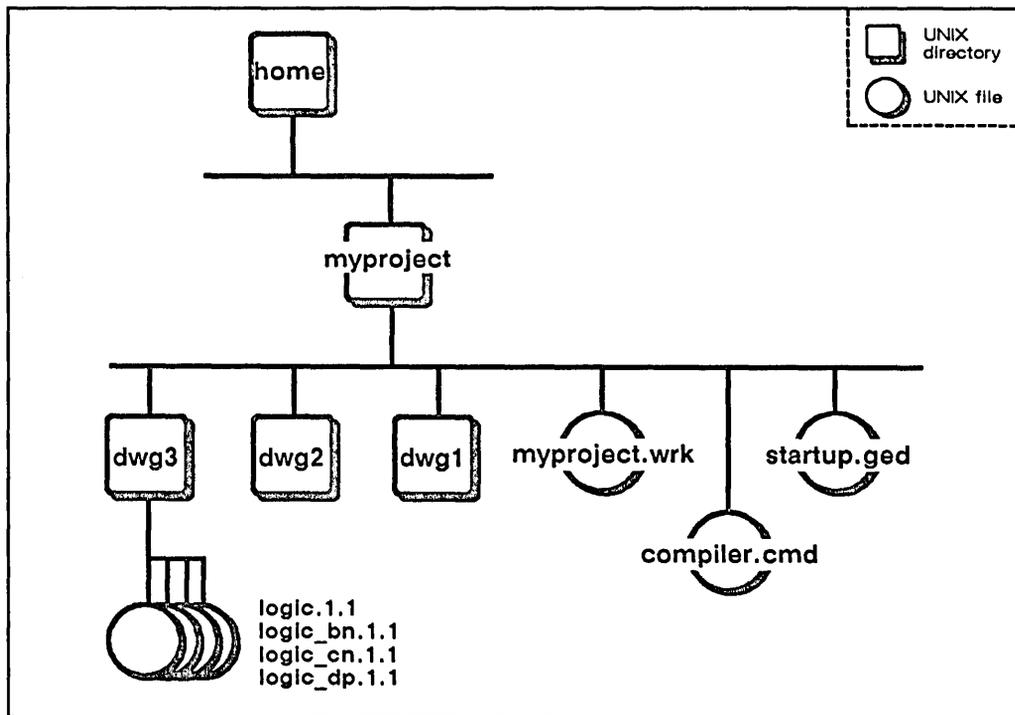


Figure 6-2. Sample Design Directory

## **Simple Project Account Security**

To set up the design directory, run *mkusr* and give the project name as the user name. This creates a UNIX directory for the project.

All users working on an individual project can log in with the project name. A project account allows each member of the group convenient access to, and *write* permission for, the files in the design. There are two disadvantages to project accounts: problems can occur when two users try to simultaneously access the same command or data files. With this approach, there is also a lack of password confidentiality.

## **Strict Project Account Security**

A second, more secure approach, and one that requires more work from the System Manager, is to create the design directory as indicated above, but have each user log in with their own user name, instead of the project name.

For this approach, the System Manager must change the default file permissions on the design directory files. By default, only the user who creates a drawing has write permission on that drawing's files and in that drawing's directory. Anybody can copy the drawing to their own directory, or use it unchanged from the original directory, however, they cannot modify it unless they first copy it. Problems can occur when several different users need to have write permission in the same directory.

Separate access can also limit user capacity to run the Logic Simulator, Timing Verifier, or Packager on drawings they do not “own.” If everyone on the system is given write permission, this may solve the permission problems, but the security problem remains.

A safer solution to the file permission problem is to assign the people working on a project to a group (by modifying the */etc/group* and */etc/passwd* files), and then give write permission in the design directory only to that group.

Figure 6-3 shows how this arrangement works. The figure shows the top directories for users ken, pam, joe, and toaster, who belong to the project group “toast.” Ken, Pam, and Joe would have the command **use /home/toaster/toaster.wrk** in their *startup.ged* files in their home directories.

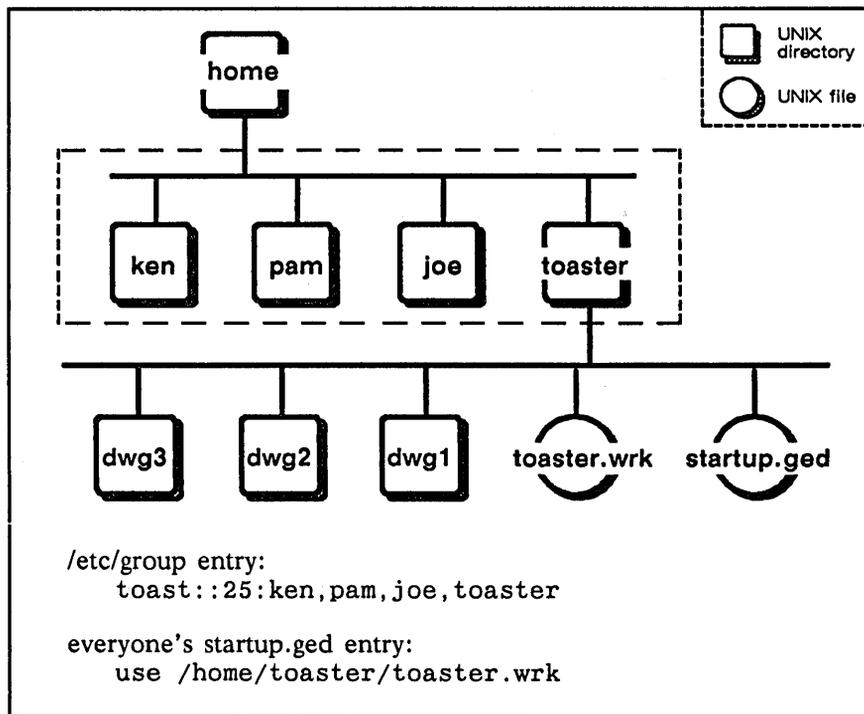


Figure 6-3. Project Account Directory

For more information on file permissions, see File Permissions, page 6-10.

The following procedure explains how to set up file protections so that the members of a project group, and no one else, can modify the group's files.

- 1 Edit the */etc/groups* file and create groups for each of your project teams. Pick unique group ID numbers for each group. Do not use group ID 0; this is the ID for the group containing root, sys, and other privileged users.

**EXAMPLE**

Avoid using group ID 10, which is the default “user” group. See *mkusr* in the Valid supplement to the UNIX manual pages for more information. Do not use group passwords; they are a potential security loophole.

For a group designing three different projects, the */etc/group* file might look like this:

```
wheel:*:0:
daemon:*:1:
kmem:*:2:
bin:*:3:
news:*:6:
lib:*:8:
staff:*:10:
other:*:20:
user::2791:fe,scald,user
proj1::20:gary,joe,pam,mary
proj2::21:carol,lee,stuart
proj3::23:paula,kevin,hubert
+::0:
```

- 2** Edit the */etc/passwd* file with *vi* and change each user’s group ID number as necessary. Carefully check the file format. Then rebuild the Yellow Pages password database.
- 3** Add the command **umask 002** to each group member’s *.login* and *.profile* files. This gives the group write permission for new files and directories created by users in the group.

**4** Add the **umask 002** command to the default *.login* and *.profile* files in the directory */usr/valid/tools/adm/ALL*. These are files provided to all new users when accounts are created.

**5** Change the file permissions on the existing files that the groups will share. To do this, use the commands

`chmod 664 filename`

and

`chmod 775 dirname`

to set permissions on files and directories. The *chmod* command accepts wildcard arguments.

**6** When you create accounts for new users, assign them to the proper existing or new group.

**7** Instruct all users to work in subdirectories of their own login directories, not in the project account's directory.

## Using chmod

You can use the *find* command to recursively search a directory tree and execute *chmod* on all of the files and directories. The C shell command

```
find . -type d -exec chmod 775 {} \;
```

recursively changes the permissions on all directories under the current directory. The C shell command

```
find . -type f -exec chmod 664 {} \;
```

recursively changes the permissions on all regular files under the current directory. Warn your users that they must restore execute permissions on any files that should be executable in the directory trees you modified.

Files and directories should have the following permissions:

```
drwxrwxr-x 19 ken 512Nov 14 17:20 Mydir
```

```
-rw-rw-r-- 1 ken 7135Nov 14 10:55 Myfile
```

## File Permissions

File permissions are used to prevent users from gaining access to files that they are not authorized to read or modify.

By default, any user can read any other user's files. Also by default, only the "owner" of a file (the person who created it) can edit or delete the file. The default permissions for new files are set with the *umask* command. As you have already seen, permissions on existing files are changed with the *chmod* command.

Each file and directory in UNIX carries a three-digit octal code that defines read (r), write (w), and execute (x) permissions.

### Read permission

Allows you to look at the file names in a directory (with *ls*) or the contents of a file (with *more*).

### Write permission

Allows you to edit a file (with *vi*) or delete a file (with *rm*). Write permission for a directory allows you to write files into the directory or delete files from that directory.

### Execute permission

Allows you to execute the file as a program. Execute permission on a directory allows you to *cd* to the directory and to specify it in path names.

Only a file's owner, or the super-user (root), can change the permissions on a file. However, the owner of a directory can override the permissions on any file in that directory regardless of the file's ownership and its subsequent permissions.

## Default Permissions

### EXAMPLES

The three permissions — read, write, and execute — are specified for three different types of users: the owner of the file, the owner's group, and everyone else.

To examine the permissions on a file or directory, use the `ls -l` command. For example, listing a user's name directory would show a *startup.ged* file similar to:

```
-rw-r--r-- 1 steve 4659 Aug 5 09:12 startup.ged
```

The entry shows that the file has standard default permissions: its owner (steve) can read, edit, or delete the file. Anyone else can read but cannot edit or delete the file. If this file was a program, no one could execute it without steve or root first changing the execute permission on the file. A directory entry for such a program would be similar to:

```
drwxr-xr-x 7 steve 1024 Sep 16 15:11 project1
```

This entry shows the default permissions on a directory. The `d` in the first column identifies the file as a directory. The owner (steve) can read or write files in the directory and list the contents of the directory. Anyone else can `cd` to the directory and list the files in the directory, but they cannot write new files into the directory or change any files.

## Group Permission

By default, all users belong to the same group. The group memberships on your system are listed in the file */etc/group*, which you can modify with `vi` at any time.

To provide members of a group unrestricted read and write access to files while restricting access to users outside the group, include the following *umask* command in each group member's *.login* file:

**umask 007**

With this *umask* value, files and directories are created with the following permissions:

```
--rw-rw----  
drwxrwx---
```

To change existing files to match these permissions, use both of the following *chmod* commands:

```
chmod o-rwx filename  
chmod g+w filename
```

If read permissions are allowable to users outside the group, yet they are not allowed to write or delete those files, use **umask 002** rather than **umask 007**.

---

---

## Strict Permissions

In order to prevent anyone but a file's owner from reading or modifying a file, use *chmod* as follows:

```
chmod og-rwx filename
```

If you want to apply this permission to all new files created by a specific user, put the following *umask* command in the user's *.login* file:

```
umask 077
```

Files and directories will have the following permissions:

```
-rw-----  
drwx-----
```

## Permissions for Programs and Scripts

Valid applications software is stored in directories for which all users have read and execute permissions, but not write permission. Similarly, most of the SCALD and UNIX programs can be read or executed by anyone. There are a few powerful programs, such as *shutdown*, which can only be executed by root.

If you create shell scripts or particular programs for your users, remember to give them execute permission for those files. To do so, use the following command:

```
chmod uog+x filename
```

---

---

## Special Permissions

In addition to the three permission “bits” described previously, there is a fourth bit that allows special permissions: set-user-ID permission, set-group-ID permission, and the “sticky bit.” While it is unlikely that you will need them, a brief review follows:

### Set-user-ID Permission

A set-user-ID program temporarily sets the user ID of the person executing the program to that of the program’s owner. Several system programs require root privileges so that ordinary users can perform necessary tasks with that program. An `ls -l` of a file with this permission looks like this:

```
-rwsr--r-- 7 root 1024 Sep 16 15:11 mycode
```

### Set-group-ID Permission

Set-group-ID programs work much the same way: the person executing the program temporarily carries the group ID of the program’s owner. An `ls -l` of a file with both set-user-ID and set-group-ID permissions looks like this:

```
-rwsr-sr-- 7 root 1024 Sep 16 15:11 mycode
```

### The Sticky Bit

Setting the “sticky bit” causes the system to save a program’s image in the swap area on the disk. It also shortens the amount of time required to read frequently-used programs, such as *vi*. Be careful to set this permission only as necessary; setting this bit on several programs can quickly fill the swap space and slow system performance. The sticky bit permission looks like this:

```
-rwxr-xr-t 5 root 153886 Jan 2 86 /usr/ucb/vi
```

## Understanding SCALD Directories

A SCALD directory is a file that maps names given to drawings by users to their corresponding UNIX directories. By convention, a SCALD directory file ends with a *.wrk* extension.

SCALD directories are automatically created and updated by the Graphics Editor (GED).

There should only be one SCALD directory in each UNIX design directory. More than one makes it difficult to keep track of which drawings belong to which SCALD directory.

While users are cautioned against modifying the contents of their SCALD directories, if a SCALD directory file is accidentally deleted or altered, you may be required to edit a user's SCALD directory file to correct or resolve the situation. Figure 6-4 shows a sample SCALD directory.

```
FILE_TYPE = LOGIC_DIR;
"UNIX INODES" `unixinodes`;
"SYSTEM BOOT PROC" `systembootproc`;
"ETHERNET" `ethernet`;
"UNIX INODE STRUCTURE" `nxndestructure`;
"BACKPANEL" `backpanel`;
END.
```

Figure 6-4. Sample SCALD Directory

The first line of the file is the file type. Unless you are doing library development work, the file type is almost always LOGIC\_DIR. The last line must be "END."

Each entry in the file maps the name given to a drawing in GED with the name of the corresponding UNIX directory.

The UNIX directory name is derived from the drawing name according to the following rules:

- All upper case letters are converted to lower case.
- Spaces and special characters are deleted.
- The name is abbreviated to 14 characters by eliminating vowels.

According to these rules, the drawing named “UNIX INODE STRUCTURE” described in Figure 6-4 is mapped to the UNIX directory “nxdestructure.”

**NOTE**

If you must add an entry to a SCALD directory with *vi*, be careful when naming directories. Make sure that the real UNIX directory name matches exactly the name placed in the file.

## Accessing Drawings Over the Network

The System Manager, as super-user (root), can mount a remote design directory on the local machine. Remote mounting allows access to the drawings on another system as though they were actually on the local disk. There are two system requirements for mounting a remote design directory: both machines must support NFS, and you must be logged in as root.

The */etc/exports* file on the remote machine controls which file systems can be mounted. The design directory you want to mount must be listed in the server machine's */etc/exports* file.

### NOTE

If you find yourself executing the same **mount** command frequently, include a **mount** entry in your local */etc/fstab* file.

The following procedure explains mounting a remote directory:

- 1 Create an empty directory, using *mkdir*, on which to mount the remote directory.
- 2 Use the **mount** command to mount the remote directory.

### EXAMPLE

To mount a remote directory from a machine named "hegel" to a directory named */home2/proj4*, type:

```
mount -o soft hegel:/home/joe/proj4 /home2/proj4
```

The **-o soft** option prevents the system from halting if the remote machine goes down.

## Libraries

In addition to those supported by Valid, libraries can include your company's standard borders, in-house libraries, standard cell libraries, or any other set of drawings or cells to which your users need quick, read-only access.

SCALD libraries for parts, drawings, or cells are structured like SCALD directories. By convention, library SCALD directory names end in *.lib*.

The use of a "master library" file makes libraries more convenient to use than SCALD directories since users do not have to specify the path to the library directory. For example, it is much easier to type **lib lsttl** than it is to type the full command **lib /usr/valid/lib/lsttl/lsttl.lib**.

In a networked environment, a library server provides convenient, transparent access to component parts via NFS. Remotely mounting libraries allows any number of systems to share a single set of libraries rather than using important disk space on each individual system.

## Libraries and File Permissions

Library directories and the files below them should be owned by the user "*lib*." These library files and directories are write-protected against all other users. In order for the Valid application programs to use a library, the parts must be owned by *lib*, and *lib* must have write-permission in the subdirectories for each part. In order for an application program to use a remote library, local user "*lib*" must have *rsh* privileges on the remote machine.

## The Master Library File

The system maps library locations to alias names in the file */usr/valid/lib/master.lib*. When you install a new library with *vininstall*, the master library file is updated automatically. However, if you add libraries of your own, you will need to add entries to this file. A sample *master.lib* file is shown in Figure 6-5.

```
file_type = master_library;
'sim'  '/usr/valid/lib/sim/sim.lib';
'time' '/usr/valid/lib/time/time.lib';
'standard' '/usr/valid/lib/standard/standard.lib';
'phantom' '/usr/valid/lib/phantom/phantom.lib';
'help'  '/usr/valid/tools/editor/doc/help.wrk';
'spice' '/usr/valid/lib/spice/spice.lib';
'tutorial' '/usr/valid/lib/tutorial/tutorial.lib';
end.
```

Figure 6-5. Sample *master.lib* File

The first line of the file specifies the file type. The last line must be “end.” Each entry in the file has: a quoted alias, beginning in the first column, one or more spaces, a quoted UNIX pathname to the library’s SCALD directory, and a semicolon terminator. Please note that you must use quotation marks (‘) as indicated, not the grave accent (`).

## Mounting Valid Libraries

Mounting Valid libraries from a server is essentially the same as mounting any file system. The only additional consideration occurs when the libraries are located in different directories on the server and client. On a Sun Workstation, libraries are logically located in */usr/valid/lib*.

When mounting libraries on either identical directory structures or dissimilar directory structures, the following general rules should be considered:

- Keep the library directory structure intact. The installation program (*vinstall*) installs libraries at designated directory locations. See *Valid Software on Servers* in Section 2 of this manual for more information.
- Minimize the number of mount points and symbolic links to simplify system management and make NFS easier to use.

### NOTE

When mounting libraries on dissimilar directory structures, you must also maintain the *master.lib* file on the client. The *master.lib* file, as demonstrated in Figure 6-5, is referenced by several Valid applications.

## File System Maintenance

The *df* command reveals the amount of space available on each file system. Figure 6-6 shows a sample *df* command response.

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/sd0a	7495	5677	1068	84%	/
/dev/sd1c	129727	87887	28867	75%	/mnt
/dev/sd0g	83623	66745	8515	89%	/home

Figure 6-6. Sample *df* Command Output

### NOTE

Using *df* at the beginning of each work day allows you to notice fast-growing file systems. When most of your maintenance time is occupied freeing small amounts of disk space, the addition of another disk drive is recommended.

## Maintaining the Root File System

The amount of free space in the root file system rarely changes. You should keep approximately one-half megabyte free in the root file system, if possible. Diminishing space in the root file system is most likely caused by one of the following:

- New programs and scripts added to the root file system rather than to the */usr/local* directory.
- Program running in the root file system causing a core dump (“core” files can be very large).
- Growing accounting files; large output files can be created if the system accounting utilities are running and the files are not periodically purged.
- Files not deleted from */tmp*. Programs such as *vi* create temporary files in the */tmp* directory which may not be automatically deleted upon completion of their tasks.

When a file system is full, the system displays the message `File System Full` each time a user tries to do something that requires a “write” on the root file system. Since *vi* and other programs use the */tmp* directory for temporary files, these programs also complains when the root file system is full.

## Maintaining the User File Systems

Over time, your user file systems fill up with design data, output files from the SCALD analysis tools, memos and documentation that your users write. Here are some ways to eliminate unnecessary disk usage:

- The optional on-line UNIX manual pages require 4.5 Megabytes in the */usr* file system. If you no longer require them, you can delete the manual pages from your system and use NFS to mount the manual pages from another system in your own */usr/man* directory. To do this, first delete your system's manual pages with the commands:

```
cd /usr/man
rm -r *
```

Then add the following line to the */etc/fstab* file:

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

Replace *host* with the name of the server machine. When you reboot the system, the manual pages from the server will automatically be mounted on */usr/man*.

The server must include */usr* or */usr/man* in its */etc/exports* file in order for you to mount the manual pages.

- By default, GED writes both an ASCII and a binary copy of each drawing. In most cases, two files are unnecessary.

To eliminate the slower ASCII file, place the command:

```
set noascii
```

in each user's *startup.ged* file. This tells GED to write only binary format output files.

- Whenever a UNIX command or program crashes, including the SCALDsystem tools, UNIX creates an image of memory in the current directory, in a file named "core". Periodically you should delete all the "core" files on the system.

Using *find* and *rm* to locate and remove all "core" files also deletes any user files coincidentally named core. To avoid deleting any user files by this name, use the command:

```
find / -name core -print | mail root &
```

This command generates a list of every core file on the system and then mails the list to root. Check with your users if there are core files in their directories, and remind them not to use the name "core" for any permanent files.

- *vw.spool* files from the GED **hardcopy** command and the SCALDstar plotting utilities are often left in working directories. Periodically locate all of these files and ask their owners to delete the ones they no longer need. If a user wants to keep a large *vw.spool* file, suggest putting it on a *tar* tape.

The command

```
find / -name vw.spool -print | mail root &
```

generates a list of all the *vw.spool* files on the system, and mails the list to root.

- Files that were spooled can accumulate in the directories under */usr/spool*. Periodically check the contents of the spool directories:

```
ls -l /usr/spool/*
```

If the */usr/spool/mqueue* directory contains old files, use **ps -ax** to check that the *sendmail* daemon is running. Then check to make sure that your network and uucp connections are operating properly.

*Use the lpc utility to clean the plotter spooling directories; see Section 7, Plotting, for more information.*

## Users Who Overuse the Disk

**EXAMPLE**

The *du* command reports disk usage, and can identify those users who have not deleted unnecessary files from their systems. With *du*, you can determine the number of kilobytes in specified directories and sub-directories. To check the disk usage in the */home* file system, use the command:

```
du /home > du.log &
```

The command *du* lists every directory under the one you specify. After listing each file in a directory, *du* prints the total for the directory. Look for the summary entry for each of the top directories in the file system:

In the following sample of a *du* file, user Fred has 5.681 Megabytes under his directory.

```
5681    /home/fred
1200    /home/gary
4       /home/maureen
2200    /home/catie
335     /home/stuart
11      /home/julie
9431    /home
```

Use the *du* command regularly to identify those users who utilize more space than needed on the disk. You will also spot directories that are growing rapidly, which may indicate a problem program.

## Archiving a Design

Most users prefer to keep designs on-line, however, your disk cannot maintain an increasing number of files indefinitely. Once a design has been finished, it should be archived on tape and deleted from the disk. When creating an archive, always assume that someone other than yourself will be restoring the design to the system. Be systematic, and document what you do. Here are some points to keep in mind:

- Put **everything** on the tape. Include all of the design directories, and all user-created library directories. Also include all user or site-specific *property.dat*, *textmacro.dat*, and physical part table files.
- Put the component libraries on the tape. Since you might delete one of the libraries from your system at a future date, saving the libraries on the tape guarantees that you will have them available if you ever work on the design again.
- Put all supplemental documentation for the design on the tape.
- Archive the command files and final output files for all SCALD tools, particularly the Packager output files.
- Make two copies of each tape, store one off-site and label all tapes clearly.
- Check each tape after you make it, and be sure you can read the contents by extracting the first and the last file.

The easiest way to archive a design is to use *tar*.

Move to the design directory and delete everything you do not want to archive. This may include old memoranda and *vw.spool* files.

Next, look at the *startup.ged* or *startup.led* file and check which libraries are used in the design. You should put the libraries on the tape.

Careful planning of the *tar* command line will make it much easier to restore the design in the future. It is best to name the design directory in the command line, rather than using **tar cv \*** from inside the directory. Naming the design directory ensures, when you restore the design, that you end up with one new subdirectory under the current directory, rather than a large number of files.

Move to the directory above the design directory by typing:

```
cd ..
```

For a simple design, such as one that was created in a single design directory and that uses only parts from the 74LSTTL library, the *tar* command might look like:

```
tar cvf /dev/rst0 proj1 -C /usr/valid/lib lsttl
```

This command writes the design directory, "proj1" onto the tape, performs a *cd* to */usr/valid/lib*, and then writes the *lsttl* library onto the tape.

After you create the tape, use **tar tvf** to check that the tape was written successfully:

```
tar tvf /dev/rst0 >& archive.log
```

The file *archive.log* contains a list of every file and directory on the tape. Look through the log file and make sure the tape is complete, then print *archive.log* and store it with the tape.

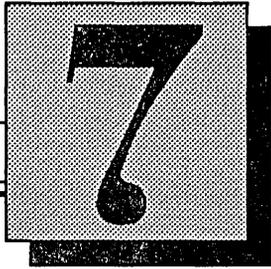
When you read this tape with **tar x**, two directories are created under the current directory: *lsttl* and *proj1*. Once you've read the tape, you can move the library or the design directory using **cp -r** if necessary.

For a more complicated design, you may need to put several design directories and user libraries on the tape:

*Note that the backslash (\) allows a long UNIX command to be continued onto a second line.*

```
tar cvf /dev/rst0 bigchip -C /usr/joe pads -C\  
/usr2/mary cmogates
```





## *Plotting*

**T**his section describes:

- The *printcap* file
- Valid supported plotters/printers
- Plotting commands and utilities
- Troubleshooting plotting problems

The UNIX printer and plotter spooling mechanisms are described in the *Sun System Administration for the Sun Workstation* manual and the following UNIX manual pages: *lpc* (8), *lpd* (8), *lpq* (1), *lpr* (1), *lprm* (1), *printcap* (5), and *vpr* (1V). Read the UNIX documentation before configuring your hardware.

Tables 7-1 through 7-4 list the plotters/printers supported by the Valid applications. If you need hardware and installation information on a plotter, see "Adding Additional Hardware" in the *Sun Installing UNIX* manual.

If you are using a plotter that is not supported by Valid, Valid's optional PLOTTALK software can be used to create an interface to virtually any plotter. PLOTTALK is a Plotter Interface Library that is made up of a set of C library routines and C data structures. PLOTTALK is explained in detail in the *PLOTTALK Reference Manual*.

---

---

## Configuring The Printcap File

The file */etc/printcap* configures the system to spool printing and plotting jobs.

Valid provides a prototype printcap file in the directory */usr/valid/tools/adm/printcap.proto*. The comments in this file explain how to generate printcap file entries for the plotters supported by Valid. Append this file to */etc/printcap* and modify it, if necessary, to fit your needs.

The prototype printcap file allows you to do local plotting on an 11-inch Versatec V-80, an HP7580 pen plotter, or an A-size postscript printer. If you want to support any other arrangement of plotters or printers, you must modify the printcap file. If your plotter or printer is shared by several machines on the Ethernet, each machine's */etc/printcap* file must be set up correctly.

### File Entry Syntax

The following syntax rules must be followed for the individual entries in the */etc/printcap* file:

- The first configuration parameter on a line must be preceded by a tab; individual parameters must be separated by colons.
- More than one configuration parameter can be included on a line (if separated by a colon).
- All but the last line must end with a line-continuation character ('\'), and the last line must be terminated with a colon.

## System Considerations

The following system considerations must be observed when setting up an *etc/printcap* file entry:

- The directory *usr/spool/hpd* and the directory referenced by the “:sd=” configuration parameter in the corresponding *etc/printcap* file entry must exist to support plotting.
- There must be one entry in the *etc/printcap* file (and one spool directory) for each different type of printer or plotter you want to use.
- If there are two entries with the same name in the file (e.g., two “gedplot” entries), the first one is used.
- If you want to print remotely over the Ethernet, there must be a corresponding local printer entry on the remote host, and the remote host’s *etc/hosts.equiv* file must include an entry for your host.
- Only the spool directories for plotting GED drawings to 11-inch Versatec plotters and plotting to the HP7475 pen plotter are created automatically during Valid’s installation procedure. (These directories, under *usr/spool*, are *vpd*, *hpd*, *vers11*, and *hp7475*.) For any other plotter, including wider Versatecs and laser printers, you must create the spool directory manually in *usr/spool*. The *printcap* file examples show you the directory name to use.

- The following entry must exist. GED hardcopy will not work without this entry:

```
gedplot | Used by GED Hardcopy:\
:mx#0:\
:lp=/dev/null:\
:if=/usr/valid/tools/editor/lib/hpf:\
:sd=/usr/spool/hpd:\
:st=/usr/spool/hpd/status:\
:if=/usr/spool/hpd/errorlog:
```

- In the `gedplot` entry, *mx* specifies the maximum size of the plot file (the value “#0” allows unlimited file sizes), and *if* is the output filter. Since the *hpf* filter program spools the plotter input file instead of sending it directly to the plotter, *lp* must be set to “/dev/null.”

## Entries for Versatec Plotters

If you have a Versatec (or comparable) electrostatic plotter, there must be at least two entries for the Versatec in */etc/printcap*: one entry for Sun utilities (*/dev/vp0*) and at least one entry for GED graphics. The GED graphics device entry is needed for plotting from within GED and for the UNIX *Vpr* command. These entries are in addition to the *gedplot* entry.

## Versatec Text Device

Valid supplies a *printcap* entry for a Versatec text device. This entry must be present for GED hardcopy to work. Also, the Sun supplied optional Versatec software must be installed when you run *setup* during system installation. Text is sent to the Versatec using the *lpr* command with the **-Pversatec** (or **-Pvp**) option.

---

---

The entry for a local 11-inch Versatec printer is:

```
versatec|vp|Local Versatec printer:\
:lp=/dev/lpv0:\
:mx#0:\
:pw#264:\
:pl#66:px#2112:py#1700:tr=\f:\
:of=/usr/lib/vpf:if=/usr/lib/vpf:\
:tf=/usr/lib/rvcat:cf=/usr/lib/vdump:\
:vf=/usr/valid/tools/bin/vplotv:\
:sd=/usr/spool/vpd:\
:st=/usr/spool/vpd/status:\
:lf=/usr/spool/vpd/errorlog:
```

The entry for a remote Versatec printer is:

```
versatec|vp|Remote Versatec printer:\
:lp=:\
:rm=remotehost:\
:rp=versatec:\
:sd=/usr/spool/vpd:\
:st=/usr/spool/vpd/status:\
:lf=/usr/spool/vpd/errorlog:
```

---

---

For other Versatec sizes, change the *px#* and *pw#* entries according to the following table.

Table 7-1. Versatec Px# and Pw# Entries

Printer/Plotter	Px# Entry	Pw# Entry
22-inch Versatec	4224	528
36-inch Versatec	7040	880
42-inch Versatec	8160	1020
72-inch Versatec	14336	1792
24-inch Color Versatec	4512	564
36-inch Color Versatec	6848	856
42-inch Color Versatec	8000	1000
24-inch Benson (9424)	5952	744

## Local Versatec Graphics Device

A local graphics device entry must have two names in the first line. The first name is the name that GED and *hpf* recognize for the specific printer/plotter model, and the second name is the generic name “vplot.” The above entry is for an 11-inch Versatec; if you ask for an 11-inch Versatec in GED, the rasterization program (*hpf*) looks for a “vers11” entry. If it cannot find one, no plot is output.

When adding or creating a printcap entry, make sure that there are no spaces between the vertical bars and the printer names themselves. Only the spooling directories for 11-inch Versatec plotters are created automatically by the Valid installation procedure. You must manually create spooling directories for any of the other plotter models. Supported black and white and color vplot plotters and their corresponding GED/*hpf* names are listed in the following tables.

The local Versatec entry for plotting looks like this:

```
vers11|vplot|Versatec plotter:\
:lp=/dev/null:\
:mx#0:\
:of=/usr/valid/tools/bin/vplotf:\
:sd=/usr/spool/vers11:\
:st=/usr/spool/vers11/status:\
:lf=/usr/spool/vers11/errorlog:
```

---

---

**Remote Versatec  
Graphics Device**

The entry for the graphics device for remote plotting follows the previous pattern:

```
vers11|vplot| remote 11" Versatec plotter:\
:lp=:\
:rm=remotehost:\
:rp=vers11:\
:sd=/usr/spool/vers11:\
:st=/usr/spool/vers11/status:\
:lf=/usr/spool/vers11/errorlog:
```

You can have more than one remote Versatec plotter if there is more than one size of Versatec on your network. You must have a separate spooling directory for each size or type of printer or plotter. The entry for a second Versatec might look like:

```
vers36|vplot| remote 36" Versatec plotter:\
:lp=:\
:rm=remotehost:\
:rp=vers36:\
:sd=/usr/spool/vers36:\
:st=/usr/spool/vers36/status:\
:lf=/usr/spool/vers36/errorlog:
```

Table 7-2. Black and White Plotters

Name	Plotter
vers11	Versatec V-80, Spectrum 2552 in B/W mode
vers22	Versatec 7222 and 8222
vers24	Versatec 7224 and 8224
vers25	Versatec 7225 and 3224* in B/W mode
vers36	Versatec 7236 and 8236, 3236* in B/W mode
vers42	Versatec 8242 and 9242 in B/W mode
vers44	Versatec 7224, 8244, and 3244* in B/W mode
vers52	Versatec 8252
vers59	Versatec 8259
vers72	Versatec 8272
versv81	Versatec V-81
vers111	Versatec V-82
vers122	Versatec 8122
vers124	Versatec 8124
vers136	Versatec 8136
vers142	Versatec 8142
vers172	Versatec 8172
vers422	Versatec 7422
	*Support not verified

Table 7-3. Black and White Plotters (Continued)

Name	Plotter
vers424	Versatec 7424
vers425	Versatec 7425 and 3424* in B/W mode
vers436	Versatec 7436 and 3436* in B/W mode
vers444	Versatec 7444 and 3444* in B/W mode
vers2554	Versatec Spectrum 2554 in B/W mode
vers2556	Versatec Spectrum 2556 in B/W mode
vers2558	Versatec Spectrum 2558 in B/W mode
vers2562	Versatec Spectrum 2562 in B/W mode
vers2564	Versatec Spectrum 2564 in B/W mode
vers2566	Versatec Spectrum 2566 in B/W mode
vers2568	Versatec Spectrum 2568 in B/W mode
b9424	Benson 9424 in Versatec compatibility mode
b9436	Benson 9436 in Versatec compatibility mode
b9444	Benson 9444 in Versatec compatibility mode
b9624	Benson 9624 in Versatec compatibility mode
b9636	Benson 9636 in Versatec compatibility mode
b9644	Benson 9644 in Versatec compatibility mode
calcomp5744	Calcomp 5744 (Using 907/PCI interface)
	*Support not verified

Table 7-4. Color Plotters

Name	Plotter
Cvers24	Versatec 3224 in color mode
Cvers36	Versatec 3236 in color mode
Cvers42	Versatec 9242 in color mode
Cvers44	Versatec 3244
Cvers424	Versatec 3424
Cvers436	Versatec 3436
Cvers444	Versatec 3444
Cvers2700a	Versatec C2700 series (C2736) A size paper, 4-color transfer
Cvers2700b	Versatec C2700 series (C2756) B size paper, 4-color transfer
Cvers2700a3	Versatec C2700 series (C2766) A3 size paper, 4-color transfer
Cvers2700a4	Versatec C2700 series (C2726) A4 size paper, 4-color transfer
C3vrs2700a	Versatec C2700 series (C2736) A size paper, 3-color transfer
C3vrs2700b	Versatec C2700 series (C2756) B size paper, 3-color transfer
C3vrs2700a3	Versatec C2700 series (C2766) A3 size paper, 3-color transfer
C3vrs2700a4	Versatec C2700 series (C2726) A4 size paper, 3-color transfer

## Entries for Pen Plotters

Pen plotters require both a printer entry and a gedplot entry in the printcap file. The first line of the printer entry requires the name that GED and *hpf* have for the various pen plotters. Supported pen plotters are listed in the following table.

Table 7-5. Pen Plotters Supported by Valid Applications

Name	Plotter
hp7580	Hewlett-Packard 7580B, 7585B, and DraftMaster
hp7580ex	Hewlett-Packard 7580B, 7585B, and DraftMaster in "expand mode"
hp7475	Hewlett-Packard 7475A
hpdrafrf	Hewlett-Packard Draftmaster with roll-feed option
hpdrafrfex	Hewlett-Packard Draftmaster with roll-feed option in "expand mode"
b1624sr	Benson 1624 (roll paper)
b1624	Benson 1624 (cut sheet)
b1625sr	Benson 1625 (roll paper)
b1625	Benson 1624 (cut sheet)
b1834sr	Benson 1834 (roll paper)
b1834	Benson 1834 (cut sheet)
b1835sr	Benson 1835 (roll paper)
b1835	Benson 1835 (cut sheet)
calcomp1043	Calcomp 1043 (Using 907/PCI interface)

---

---

**Local Pen Plotter**

An entry for a local pen plotter is:

```
hp7580| local E size HP pen plotter:\
:lp=/dev/device:\
:mx#0:\
:br#9600:\
:sh=:\
:sd=/usr/spool/hp7580:\
:st=/usr/spool/hp7580/status:\
:lf=/usr/spool/hp7580/errorlog:
```

In the above example, *sh* (suppress header) is required for all pen plotter entries. Also, the *device* specified by the *lp* parameter is the tty port attached to the plotter. *Device* can either be a specific port (e.g., "ttyb") or logical port (e.g., "hp" with */dev/hp* linked to a specific tty device such as "*/dev/ttyb*").

---

---

## Remote Pen Plotter

A remote entry for a pen plotter is shown below.

```
hp7475| remote B size HP pen plotter:\
:lp=:\
:rm=remotehost:\
:rp=hp7475:\
:sh=:\
:sd=/usr/spool/hp7475:\
:st=/usr/spool/hp7475/status:\
:lf=/usr/spool/hp7475/errorlog:
```

In the above entry, the *lp* parameter must be empty, and parameters must be added for *rm* (remote machine) and *rp* (remote printer name).

## Entries for Postscript Printers

### Local Postscript Printer

GED plotting is supported by local and remote A- and B-size laser printers with a postscript interface. These printers work with any RS-232 serial plotter or laser printer that understands Postscript.

The printcap entry for a local A-size postscript (*postscripta*) printer is:

```
postscripta| local A-size laser printer:\
:lp=/dev/device:\
:mx#0:\
:br#9600:\
:sd=/usr/spool/postscripta:\
:st=/usr/spool/postscripta/status:\
:lf=/usr/spool/postscripta/errorlog:
```

In the above entry, the *device* specified by the *lp* parameter is the tty port attached to the printer. *Device* can either be described specifically (e.g., “ttyb”) or logically (e.g., “psa” with */dev/psa* linked to a specific tty device such as “/dev/ttyb”).

To plot to the laser printer, use the command **set plotter postscripta** or **set plotter postscriptb** while in GED.

---

---

## Remote Postscript Printer

The printcap entry for a remote B-size postscript printer (*postscriptb*) is:

```
postscriptb| local B-size laser printer:\
:lp=\
:rm=remotehost:\
:rp=postscriptb:\
:sd=/usr/spool/postscriptb:\
:st=/usr/spool/postscriptb/status:\
:lf=/usr/spool/postscriptb/errorlog:
```

In the above entry, the *lp* parameter must be empty, and parameters must be added for *rm* (remote machine) and *rp* (remote printer name).

---

---

## GED Plotting

GED hardcopy in `set local_plot` mode works as follows:

- 1** GED sends *hpf* header information (plotter type, scale, etc.) and the binary or body drawing file to *hpr*.
- 2** *hpr*, a shell script wrapper, calls *lpr* with the `-Pgedplot` option.
- 3** The *lpd* daemon is contacted, and the data is sent to the spool directory listed in */etc/printcap* (in the */usr/spool/hpd* directory).
- 4** The */etc/printcap* entry for *gedplot* instructs *lpd* to run the *filter* program (in the directory */usr/valid/tools/editor/lib/hpf*) on the data.
- 5** *hpf* transforms the input data into a format acceptable to the plotter and calls *lpr* with the `-Pplotter_name` option to send the formatted plotter file to the correct directory for the type of plotter specified.
- 6** Another daemon then spools the plotter-specific files to the plotter.

The double spooling (Steps 5 and 6) allows other programs to access plotters in an orderly fashion, and keeps GED users from having to wait for their plots to be processed.

When GED is in **set spooled\_plot** mode, *hpf* is used with the **-f** option (in stead of *hpr*) to create a *vw.spool* file. Since the *lpr* spooling system is not involved, the GED user must wait while *hpf* creates the *vw.spool* file.

The following files and programs are involved in plotting:

- **/usr/valid/tools/editor/lib/hpr**  
*script that calls lpd*
- **/usr/ucb/lpr**  
*program to enter job in spool directory*
- **/usr/lib/lpd**  
*line printer daemon*
- **/etc/printcap**  
*plotter setup configuration file*
- **/usr/spool/spool\_dir**  
*spool directory*
- **/usr/spool/spool\_dir/lock**  
*status lock file*
- **/usr/valid/tools/editor/lib/hpf**  
*plot data filter*

## Plotting Commands and Utilities

This section gives a brief overview of the UNIX plotting commands and plot queue monitoring facilities.

In addition to the generic UNIX print and plot spooling commands *lpr*(1), *lprm*(1), and *lpq*(1) there are three commands specifically for plotting and printing on Versatec plotters:

### **vpr**

The *vpr*(1V) command spools a plot file for plotting on the Versatec plotter. It can be used to plot the *vw.spool* files created by GED Hardcopy or the *Instaplot* and *Makeplot* utilities.

If you are plotting output files from Makeplot, Valid recommends that you not use the *lpr* spooling mechanism as the intermediate files would probably fill up your disk. Instead run the *vplotv* utility directly using the */dev/lpv0* option:

```
cat vw.spool | /usr/valid/tools/bin/vplotv\  
[-wwidth] > /dev/lpv0
```

*Note that the backslash (\) allows a long UNIX command to be continued onto a second line. The backslash is not entered in the command itself.*

The example above works for the Ikon 10088 VME plotter interface. If you have an Ikon or Systech Multibus plotter interface board, use the utility *vplotm* and the device *vp0*:

```
cat vw.spool | /usr/valid/tools/bin/vplotm\  
[-wwidth] > /dev/vp0
```

The *-w* option specifies the number of bytes per line (264 for vers11). Additional *vplot* values for other plotters are as follows:

Table 7-6. Vplot Width Values

Plotter	Bytes/Line
11-inch Versatec	264
22-inch Versatec	528
36-inch Versatec	880
42-inch Versatec	1020
72-inch Versatec	1792
24-inch Color Versatec	564
36-inch Color Versatec	856
42-inch Color Versatec	1000
Benson 9424	744

**vpq** The *vpq*(1V) command lists the files in the Versatec plotting queue.

**vprm** The *vprm*(1V) command deletes a file from the plotting queue.

## The lpc Utility

**EXAMPLE**

The line printer control utility *lpc*(8) allows the System Manager to monitor and control the plotting queues. To start up the utility, enter the command:

```
lpc
```

The system displays the following prompt:

```
lpc>
```

To check the status of the plotting and printing queues, use the *status* command, as follows:

```
lpc> stat
versatec:
  queuing is enabled
  printing is enabled
  no entries
  no daemon present
gedplot:
  queuing is enabled
  printing is enabled
  no entries
  no daemon present
vers11:
  queuing is enabled
  printing is enabled
  1 entry in spool area
  no daemon present
lpc> quit
```

See the *lpc* (8) manual page for more details on this useful utility.

## Troubleshooting

**EXAMPLE**

The first thing to do if you have trouble with plotting is to determine whether the problem is with the hardware or the software. The easiest way to check whether the hardware is operating correctly is to bypass the spooling software and send a *vw.spool* file, preferably one you have plotted before, directly to the plotter filter.

The following example invokes the versatec plot decoder, sending the output directly to the device. The `-w` option specifies the number of bytes per line; the default is 264 for an 11-inch Versatec.

```
cat vw.spool | /usr/valid/tools/bin/vplotv \
[-wwidth] > /dev/lpv0
```

For pen plotters and postscript plotters, you can test the hardware with the command:

```
cat vw.spool > /dev/ttya
```

If necessary, substitute *ttyb* for *ttya*.

If the plot comes out, you know that the hardware works. If the plot does not come out, check the cabling for the plotter and verify the plotter interface installation.

---

## Configuring the Printcap File

The file */etc/printcap* configures the system to spool printing and plotting jobs.

If your */etc/printcap* file is correct and your system is configured properly for plotting, rebooting the system fixes most plotting problems and starts plotting of any jobs that were “stuck” in the plotting queues. However, rebooting the system is not always convenient. The steps below show how to diagnose and fix common plotting problems.

## Hardcopy Error Log File

If you are having problems with *hardcopy* you may be able to determine the problem by looking at the error log.

The error log is in the file */usr/spool/hpd/hpferrorlog*. This file contains the last 3K bytes of error messages from the *hpf* program. Each invocation of *hpf* starts with the message:

```
New invocation of /usr/valid/tools/editor/lib
/hpf: date
```

After the header, error messages (if any) appear. Common problems include missing bodies and permission problems. Also, if you have discovered a bug in *hpf* or GED, the internal error messages in the log file will help pinpoint the problem. Print out a copy of this file and include it with the Software Problem Report when you report a plotting software problem to Valid.

## Lpd Error Logs

The master *lpd* daemon has an error log file, */usr/spool/lpd-errs*, for error messages. If you are trying to fix remote printing, look at *lpd-errs* on the remote machine as well. Each copy of *lpd* for individual spool directories opens an error file if there is an *lf* entry in */etc/printcap* (usually *spool\_dir/errorlog*.) Check these for any error messages.

Here are a few additional things to check:

- The owner of all the spool directories must be *daemon*; the group of all spool directories must also be *daemon*.
- The mode of all the spool directories must be *drwxrwxr-x* (i.e., 775)
- The *rp* (remote printer) capability on the local machine must have an exact match with a name of a printer in the */etc/printcap* file of the remote machine.
- The files within the spool directories (i.e., *lock*, *status*, *errorlog*, and *.seq*) must be owned by *root* and have *daemon* as the group.
- No spaces are permitted between the vertical bars that separate printer names and the printer names themselves in the */etc/printcap* file.

- If you are trying to plot to a remote plotter, use the *lpc* utility's **status** command to check whether the plot is being successfully spooled to the remote system. If the plot shows up in the queue on the remote machine, the problem is not on your local machine. If you use *lpc* to stop printing, you can examine the print queues before the files are processed.
- If you are trying to get hardcopy in GED to work, remember that the name of the printer in */etc/printcap* must exactly match one of the names in the tables in the first part of this section.
- Every machine that uses *lpr* must have a "printer" service entry in the file */etc/services*, even for local printing. Also, the local and remote */etc/services* files must agree on the service number for "printer" in order for remote printing to work. The default */etc/services* file works fine, but you might have trouble if someone has modified the file.

The default entry is:

```
printer 515/tcp spooler  
#Line Printer Spooler
```

- Make sure there is at least one *lpd* daemon running. If not, log in as root and enter the command:

```
/usr/lib/lpd
```

If you change the *printcap* or *services* files, or change the ownerships of spool directories, use the **ps -ax** command to get the process ID numbers of all of the *lpd* daemons, kill them, and then restart the daemon:

```
kill -15 process_id  
/usr/lib/lpd
```

## Versatec Plotter Setup

See Appendix A for information on the installation of the Ikon 10088 VME bus plotter interface board. Also see the Sun manual *Card Slot Assignments and Backplane Configuration Procedures* (Sun P/N 813-2004-05) when checking the installation of the plotter interface board and your workstation's *Hardware Installation Manual*.

## Setting Up Pen Plotters

The correct back panel switch settings to use with the HP7580 pen plotter are:

Parity off	
Parity odd	
Full Duplex	
Hardwire	
DTR Normal	
9600 Baud	
expand	<b>normal</b>
emulate	<b>normal</b>
standalone	eavesdrop
monitor mode	<b>normal</b>
local	<b>normal</b>

## Baud Rate

In 4.2BSD UNIX, when a non-login port closes, the baud rate is reset to the default: 9600 baud. This creates a problem if you are trying to hook up a pen plotter that cannot accept 9600 baud. Baud rate is set automatically by the System Utilities *sleeper* program located in */usr/valid/tools/bin*. If you need to reset the Baud Rate, use the following command:

```
sleeper -b baudrate -f /dev/ttya &
```

where *baudrate* is the required baud.

The file */etc/gettytab* shows the baud rate codes to use in */etc/ttytys*. The most popular settings are:

0 cycles between 100, 1200, 150, 110

1 150 baud

2 9600 baud

3 cycles between 1200 and 300

4 Decwriter (300 Baud)

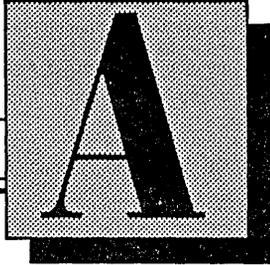
5 cycles between 300 and 1200

6 2400 baud

7 4800 baud

8 cycles between 9600 and 300

With **sleeper** running, the *lpr(1)* command should function at the selected baud rate.



## *IKON 10088 Interface Board Installation*

---

**T**his section explains how to configure and install an IKON 10088 Interface Board on a Sun Workstation. Some of the topics covered in this section are:

- Interface Board Switch Settings
- Cabling
- VME-VME Adapter Installation
- Non-generic Kernel Considerations

The Ikon 10088 interface board provides both TTL and differential outputs for a Versatec printer/plotter. The interface board is premounted on a Sun VME-VME adapter board to simplify installation.

Hardware installation of the interface board involves verifying/setting the switches and jumpers, installing the interface cable between the appropriate interface board or VME-VME rear panel connector and the printer/plotter, and installing the Sun VME-VME adapter into an expansion slot in the chassis.

---

---

## Interface Board Switch Settings

Individual switches are used to set the interface board's base address, the driver type (TTL or differential), and the word length (16 or 32 bits). These switches are shown in figure A-1. Before proceeding, make sure that toggle switch S2 (located between the two interface connectors) is in its neutral (center) position.

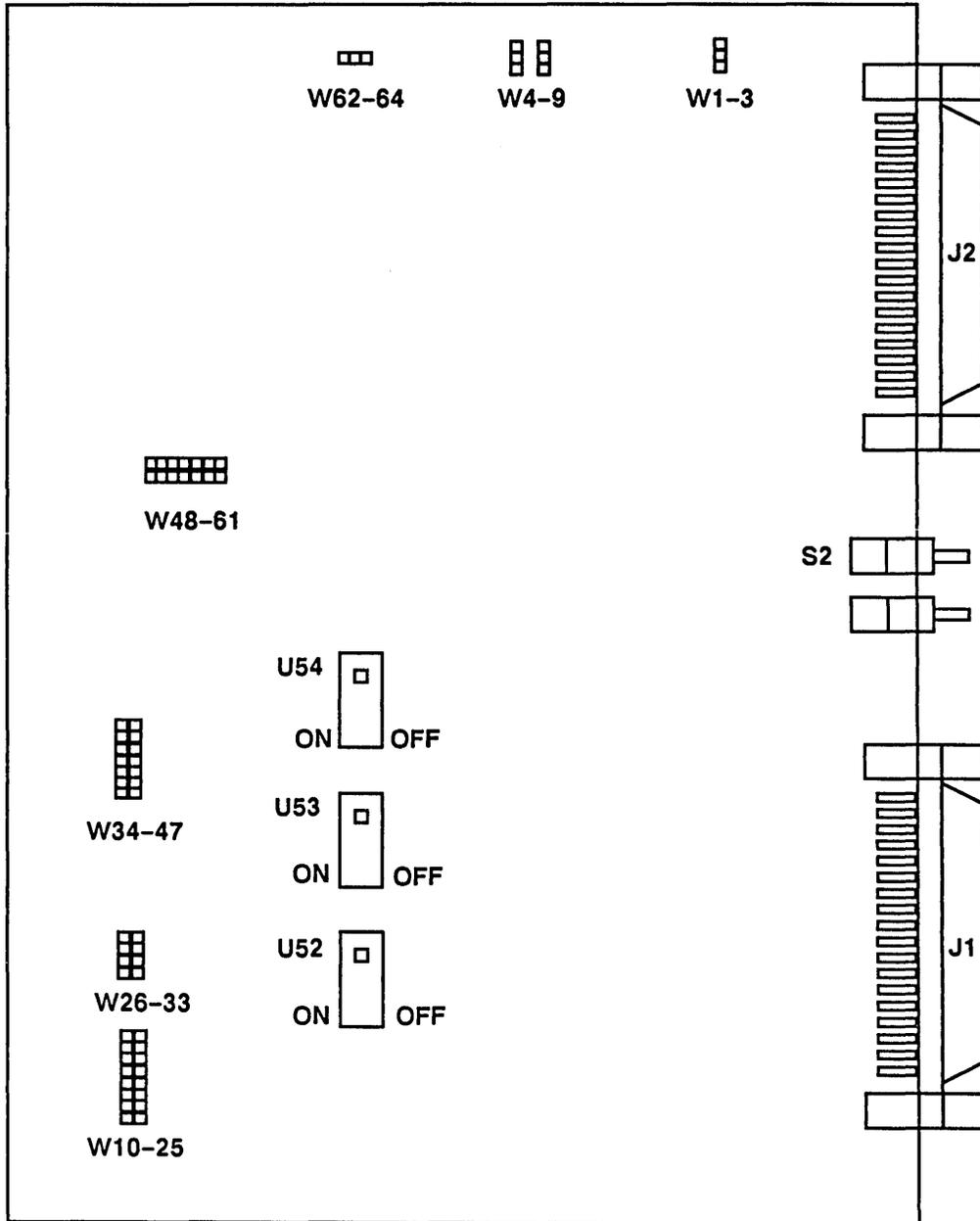


Figure A-1. Switch/Jumper Locations

**Base Address**

Switches U52, U53, and the first four positions of switch U54 determine the base address of the IKON interface board on the VME bus. These switches are set to base address E0000x (hexadecimal) according to the following table:

**Table A-1. Switch Settings**

U52 Setting			U53 Setting			U54 Setting		
1	A23	OFF	1	A15	ON	1	A07	ON
2	A22	OFF	2	A14	ON	2	A06	ON
3	A21	OFF	3	A13	ON	3	A05	ON
4	A20	ON	4	A12	ON	4	A04	ON
5	A19	ON	5	A11	ON	5	32 BITS	
6	A18	ON	6	A10	ON	6	BYTE SWAP	OFF*
7	A17	OFF	7	A09	ON	7	DIFF	ON
8	A16	ON	8	A08	ON	8	I-O MAP	OFF*

\*Always set to OFF

**Driver Type**

The type of driver (TTL or differential) is set by switch U54-7. If you are using a Versatec printer/plotter with a TTL interface, set switch U54-7 to OFF. If you are using a Versatec printer/plotter with a differential (long-line) interface, set switch U54-7 to ON.

**Word Length**

The word length (16- or 32-bit) is set by switch U54-5. If you are installing the interface board in a Sun3, set switch U54-5 to ON (32-bit). Note that on the Sun3 or Sun4, switch U54-5 can be in either position; setting the switch ON improves the throughput by transferring data to the board in 32-bit increments.

## Interface Board Jumper Settings

### Bus Grant Level

Jumpers are used to set the bus grant, bus request, interrupt request, and interrupt acknowledge levels. The locations of these jumpers are shown in Figure A-1.

The bus grant level is determined by the jumper configuration at the bus grant pin matrix (pins 10 through 25). Jumpers are installed to select bus level 3 (the Sun CPU only responds to bus level 3) and to not pass through any of the other bus grant levels. To select bus level 3, install jumpers between pins 11-13, 15-17, 19-21, 22-23, and 24-25 as shown in Figure A-2.

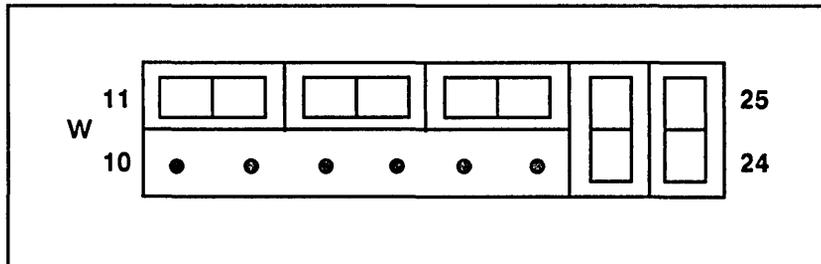


Figure A-2. Bus Grant Pin Matrix

### Bus Request Level

The bus request level is determined by the jumper configuration at the bus request pin matrix (pins 26 through 33). A single jumper is installed between pins 32 and 33 to select bus request level 3 as shown in Figure A-3.

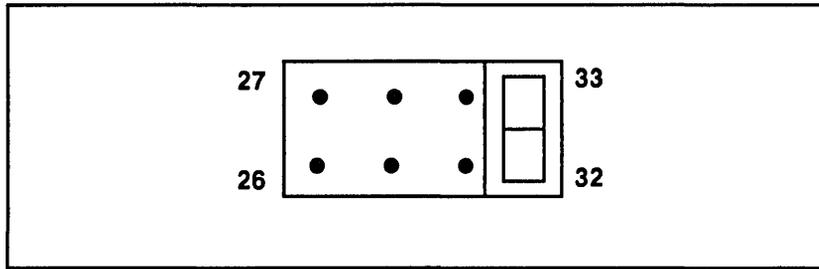


Figure A-3. Bus Grant Pin Matrix

**Interrupt Request Level**

The interrupt request level is determined by the jumper configuration at the interrupt request pin matrix (pins 34 through 47). A single jumper is installed between pins 44 and 45 to select interrupt request level 2 as shown in Figure A-4.

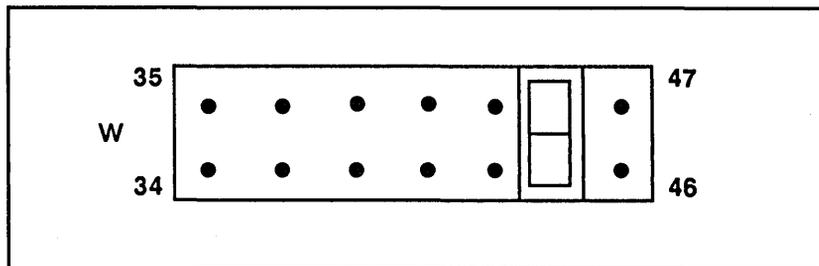


Figure A-4. Interrupt Request Pin Matrix

## Interrupt Acknowledge Level

The interrupt acknowledge level is determined by the jumper configuration at the interrupt acknowledge pin matrix (pins 48 through 61). A single jumper is installed between pins 58 and 59 to select interrupt acknowledge level 2 as shown in Figure A-5.

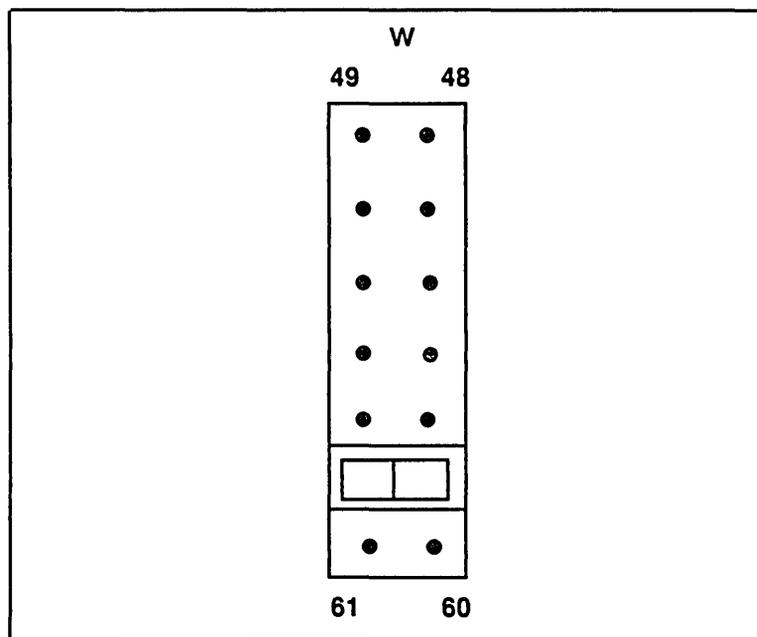


Figure A-5. Interrupt Acknowledge Pin Matrix

## Miscellaneous Jumpers

In addition to the bus and interrupt jumpers, several other jumpers must be installed for the interface board to function properly. Refer to Figure A-6 and verify that jumpers are installed between pins 2-3, 4-5, 7-8, and 63-64 along the left side of the board.

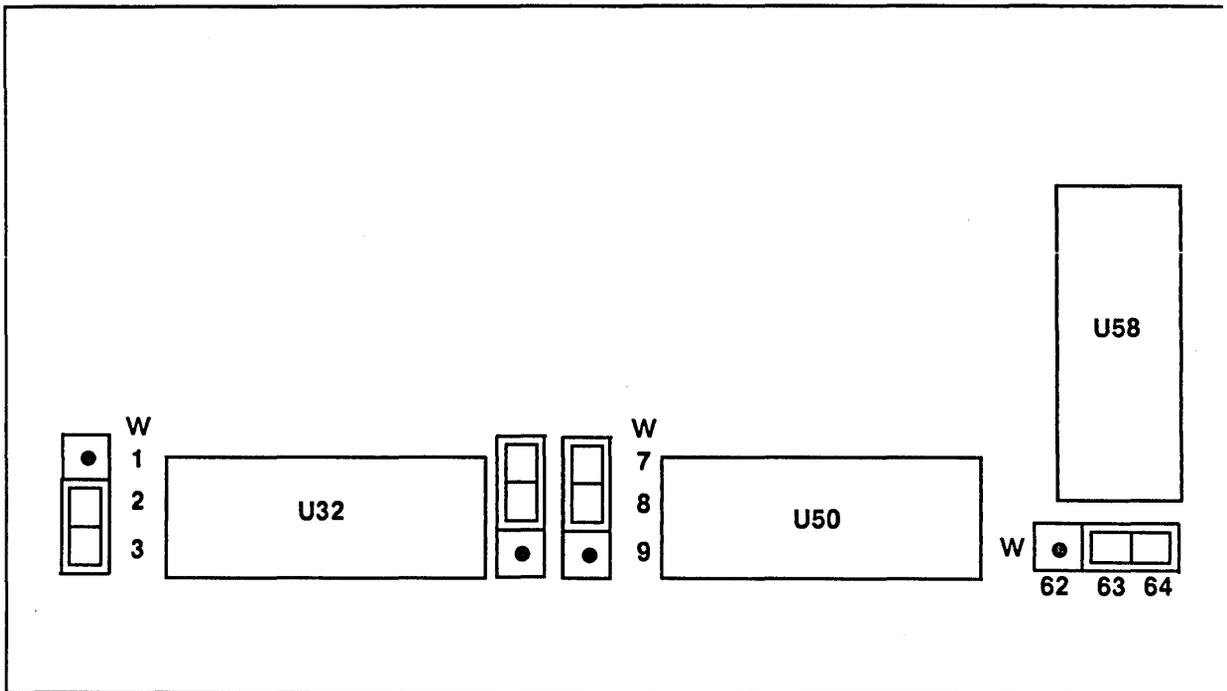


Figure A-6. Miscellaneous Jumpers

---

---

## Cabling

The Ikon 10088 interface board supports Centronics-compatible interface connectors for both a TTL and a differential (long line) cable interface. There are two VME-VME adapter cable configurations: internally connected directly on the Ikon board's interface connectors and externally connected to the interface connectors on the the VME-VME adapter's rear panel.

## Internal Cabling

With internal cabling (i.e., no cabling between the Ikon interface board and the VME-VME adapter's rear panel), the plotter cable is connected directly to the interface board's J1 or J2 connector. The plotter interface cable must be installed before installing the VME-VME adapter into the expansion chassis. The TTL cable is installed in the interface's J2 connector, and the differential cable is installed in the interface's J1 connector. The type of interface is determined by switch U54-7 (see Interface Board Switch Settings).

Insert the cable interface mating connector into the appropriate board connector (J1 or J2) and secure it in place with the mounting screws. To reduce strain on the connector, secure the cable to the VME-VME adapter with a cable tie.

## External Cabling

With external cabling (i.e., interface connectors mounted on the VME-VME adapter's rear panel), the plotter interface cable is connected to either the J1 or J2 connector on the rear panel of the VME-VME adapter. The plotter interface cable should be installed after installing the VME-VME adapter into the expansion chassis (see the next section). The TTL cable is installed in connector J2, and the differential cable is installed in connector J1. The type of interface is determined by switch U54-7 (see Interface Board Switch Settings).

Insert the cable interface mating connector into the appropriate rear panel connector (J1 or J2) and secure it in place with the mounting screws.

## VME-VME Adapter Installation

The VME-VME adapter is installed in slots 6-9 of the Sun back plane (slots are numbered from 1-12 with the Sun CPU board installed in slot 1). If you are using an adapter with internal cabling, a slot with an unused adjacent slot must be used. Install the VME-VME adapter into the selected slot and secure it in place with the top and bottom mounting screws.

After installing the VME-VME adapter in its slot, verify that jumpers Px03 and Px04 on the front of the backplane are removed. To check these jumpers:

- 1** Make sure that the system is powered down and disconnect the ac line cord.
- 2** Remove the front panel bezel. The bezel is held in place with six internal wire-spring fasteners that release when the bezel is pulled away from the chassis.
- 3** Remove the four screws that hold the power supply pull-down drawer to the chassis.
- 4** Pull the drawer out from the top and carefully lower it into its fully open position on its supports. It may be necessary to pry the top of the drawer out from the chassis due to the tight fit of the RFI strip.

- 5 Refer to Figure A-7 and make sure that the Bus Grant and Interrupt Acknowledge jumpers at Px03 and Px04 on the front of the backplane connector where the VME-VME adapter is installed have been removed. (x is the number of the slot where the adapter is installed).
- 6 Raise the power supply drawer back into the chassis and secure it in position with the four screws removed in step 3.
- 7 Replace the front panel bezel and the ac line cord and apply power to the system.

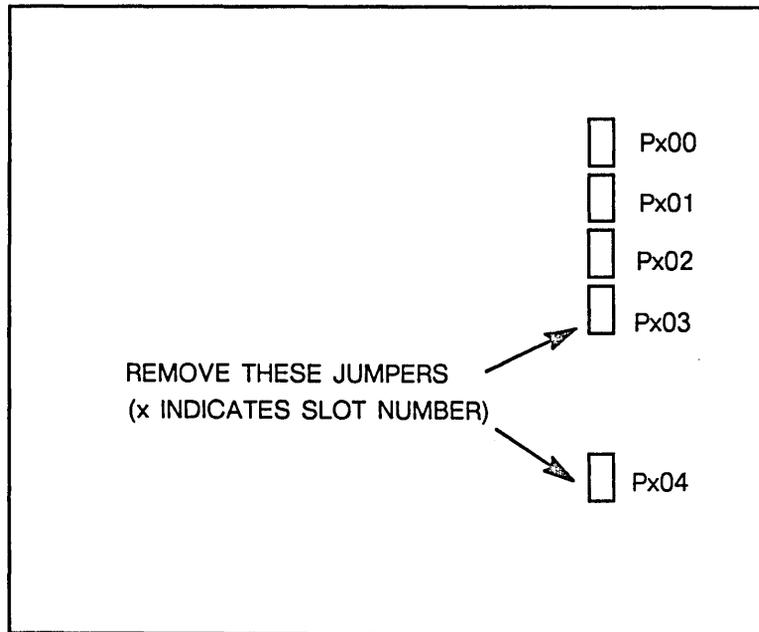


Figure A-7. Backplane Jumpers

## Software Installation

Software installation is described in Section 2 of this manual. During setup, be sure to install the Sun-supplied optional Versatec printer software and then use *vscrip*t to install the Valid System Tools package. After *vin*stall completes, execute the *mkvaliddevices* script:

```
/usr/valid/tools/bin/mkvaliddevices
```

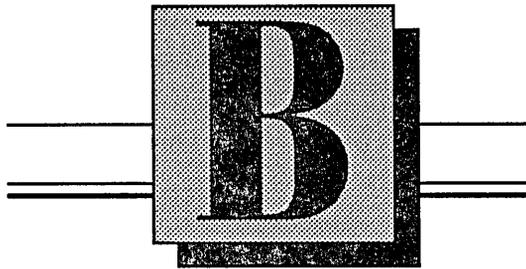
This script runs *mknod*(8) to create the appropriate character special devices in */dev*.

**NOTE** The *mkvaliddevices* script overwrites all devices of the same name.

Edit the */etc/printcap* file and merge in the Valid supplied *printcap* entry for the Versatec as described in Section 7 of this manual. After editing the *printcap* file, be sure to kill and restart the line printer daemon (*lpd*).

## **Non-generic Kernel Considerations**

The IKON 10088 driver requires a special (non-generic) kernel. Valid supplies modified kernels to support both the IKON 10088 driver and the Real Products drivers. See Appendix B for more information.



## *Real Products and Versatec Printer/Plotter Kernels*

---

**T**his section explains how to configure a custom kernel for Real Products or the Versatec printer/plotter on a Sun Workstation. Topics covered in this section are:

- Installing and configuring your kernel
- Configuring your custom kernel

## Installing and Configuring the Kernel

If you have a Versatec plotter or Valid Real Products, you need to configure the kernel on the machine to which the hardware is attached in order for the kernel to recognize the hardware. Valid provides a kernel that contains the drivers for these particular products.

If you are using Sun's generic kernel, you can simply replace the Sun kernel with Valid's modified kernel. If you are using a modified Sun kernel, you must make a new kernel that includes the Valid device drivers.

To check if your kernel has been modified, use the command:

```
strings /vmunix | grep Release
```

The result should be something similar to:

```
SunOS Release 4.0.3 (Generic) #4:date
```

Check the Sun release number (4.X) and the kernel type. If your kernel is not generic, see "Configuring Custom Kernels" later in this section.

## Installing the Valid Kernel

If you have a generic Sun kernel, perform the following steps to replace your Sun kernel with the appropriate Valid kernel.

- 1 Determine which Valid kernel to use as the replacement kernel by entering:

```
/usr/valid/tools/bin/kversion
```

The `kversion` command reports the release level of the Sun kernel and identifies which Valid kernel should be used.

- 2 Change to the Valid kernel directory and identify the kernel reported by the `kversion` command (e.g., `vmunix.sun3.os4.X`)

```
cd /usr/valid/tools/kernel
ls
```

- 3 Copy this kernel onto the root file system:

```
cp vmunixXX.Y /vmunix.valid
```

- 4 Shutdown the system by typing:

```
sync
```

```
sync
```

```
shutdown +n
```

```
(n is time in minutes)
```

- 5 When the PROM monitor prompt (>) appears, reboot the system with the Valid kernel:

```
b vmunix.valid
```

## Configuring Custom Kernels

**6** After the system boots up, execute a few simple commands to make sure that the system is functional. Use the *df* command and make sure all of your disk drives appear. (Note that commands like *ps* that read the kernel's symbol table will not work properly until you complete the next step.)

**7** Backup the old kernel (*/vmunix*) to tape using *tar(1)*. Save the tape. Then rename the kernels:

```
mv /vmunix /vmunix.old
```

```
mv /vmunix.valid /vmunix
```

If you are using a non-generic or "custom" Sun kernel, and if you are using Versatec plotters or Valid Realproducts, you will need to add the necessary device drivers to your kernel. The procedures for adding the Realproducts driver are found in the README file in one of the following directories:

```
/usr/valid/tools/kernel/realproducts/4.0
```

```
/usr/valid/tools/kernel/realproducts/4.0.3
```

Instructions for adding the Versatec Ikon10088 plotter driver can be found in the file *Installation* in the directory */usr/valid/tools/kernel/ikon10088*.

---

---

## Device and Spooler Scripts

Next, you must create the Real Products and Versatec plotter devices by running the script *mkvaliddevices*.

To run the script, log in as root and enter the command:

```
/usr/valid/tools/bin/mkvaliddevices
```

### NOTE

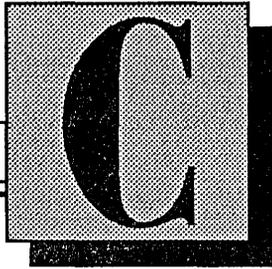
The *mkvaliddevices* script overwrites all devices of the same name.

After you run the *mkvaliddevices* script, you must run the spooler installation script on each system that has a local disk. To run the *spoolsetup* script, log in as root, and enter the following command:

```
/usr/valid/tools/bin/spoolsetup
```

The spooler installation script sets up your system for GED plotting to 11-inch Versatec plotters, HP7580 pen plotters, and postscripta plotters.





## Using *vinstall*

---

**T**his section provides additional information about the *vinstall* utility.

## The *vin*stall Utility

### SYNTAX

The *vin*stall utility is called automatically by the *vs*cript installation script during installation of the Valid applications. A copy of *vin*stall is included in */usr/valid/tools/bin* when the Valid System Utilities package is installed.

```
vin
```

stall [-options] feature\_name

Several options to *vin*stall can be used directly to verify or delete Valid application packages. By default, all packages are reported if a specific application is not indicated in the command line.

- v reports the version of an installed application.
- q reports the latest version of an installed application.
- h reports the installation history of an application.
- d deletes a previously installed application.
- r resumes a failed or stopped installation.

-f devicename installs from a tape drive other than */dev/rst0*. Please note that the -f option is only used for special installations, if instructed by Valid.

---

---

## Errors and Warnings

If problems occur during the installation, *vinstall* usually describes the nature of the problem before exiting. Check the */tmp/vinstall.msg* file for error messages.

## Incomplete Installation

The following message appears when incomplete installations occur:

```
Warning!! According to the lock file /usr/
valid/tools/adm/vinstalllock, an incomplete
installation was done of feature_name
```

The *vscript* installation script is used to reinstall a specific application.

**1** Change directories by typing:

```
cd /usr/valid/install
```

**2** Run the installation script by typing:

```
vscript
```

**3** You are prompted for user-specific information regarding the kind of tape and the kind of executables you are installing. You are then asked to select the package to be installed. Enter the number that corresponds to the *feature\_name* included in the warning message.

For complete information regarding installation, see *Using Vscript*, Section 2.

---

---

## Corrupt History File

If for any reason the *vinstall* history file *vinstallhist* becomes corrupt, the following error message is displayed:

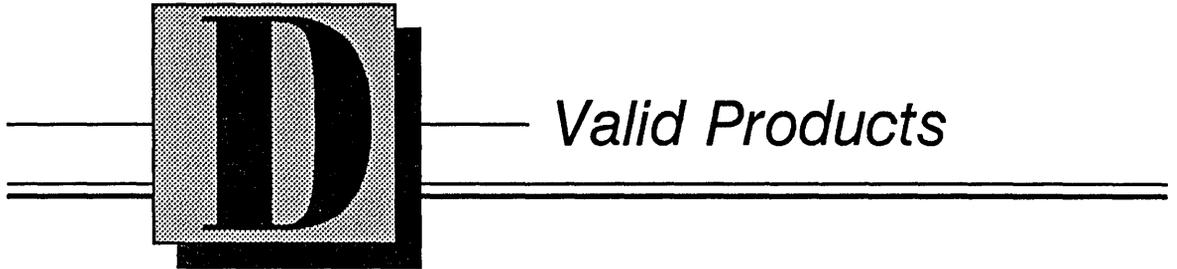
```
Vinstall: Fatal error in /usr/valid/tools/adm/vinstallhist on line N
This problem might occur if you try to do an installation with a full
file system. To fix the problem, delete the indicated line from the
/usr/valid/tools/adm/vinstallhist file, make room on your disk for the
new software or libraries, and run vinstall again. Note that if the
problem persists, restore the original vinstallhist file from your
backup tape and again run vinstall.
```

## Deleted Files from Previous Installation

During an update installation, if files from the previous version of an application are missing, *vinstall* reports:

```
Warning: The file filename does not exist on
this system
```

The missing file is replaced by *vinstall* if it exists on the current tape. If the *license* file has been deleted or corrupted, use *vscript* to reinstall the file.



## *Valid Products*

**T**his section provides the corresponding feature names for all Valid products.

## Feature Names

The following table lists all Valid products and the correct feature name for each product. When using Valid utilities that require *feature\_name* in the command line, please use the exact names provided below.

PRODUCT NAME	FEATURE NAME
allegro_bd	arouter
	intrroute.o
	pcb_editor
	placement
	swap
allegro_bi	tgilocal.o
	pcb_editor
	tgilocal.o
allegro_br	arouter
	placement
	swap
allegro_engineer	pcb_engineer
	placement
	swap
	tgilocal
allegro_md	mdin
	mdout
allegro_multiwire	multiwire
allegro_prep	pcb_prep
	tgilocal.o
allegro_review	pcb_review
	placement
	swap
	tgilocal.o
allegro_therm	thermal.o
allegroif	gallegro

PRODUCT NAME	FEATURE NAME
applicon	gappfeed gapplicon
cadds	gcadds
cadds4x	gc4xfeed gcadds4x
calay	gcalay gcalfeed
calaypcb	gcalaypcb gcalpfeed
cbds	gcbds gcbdsfeed
cif	cif2led.ex led2cif.ex
compare	compare.ex
compiler	comp expgen pcomp
compose	compose.ex
cadam	gcadam
led_drc	drc.ex
confirm_drc	drc.ex
construct	construct
flattener	flat xrefer
fsim	fsim rcfsim.o test_man ucpfsim.o
ged	ged.ex
hardcopy	hpf
hdl	ghdl
hilo	ghilo

PRODUCT NAME	FEATURE NAME
intgraph	gintgraph
lasar	glasar
led	led.ex
logcap	glogcap
makeplot	plot.ex.color
mcldl	gmcdl
packager	packager
philpac	gphilpac
redac	gredac
	gredfeed
rinf	grinf
	grinfeed
scicards	gscicards
	gscifeed
secmai	gsecfeed
	gsecmai
sim	sim
	ucpsim.o
spiceif	gspice
standardif	gbom
	glib
	gphysdly
	gscald
	gspares
stream	led2stream.ex
	stream2led.ex
tegas	gtegas5
time	verifier
timeview	tvad
timemill	timemill
pathmill	pathmill
gtimemill	gtimemill.ex

---

---

PRODUCT NAME	FEATURE NAME
transcribe	tscr.ex
wirewrp	crefer
	gwirewrp



# Index

## A

- Access Control
  - adding servers, 3-9
  - configuring servers, 2-23
  - fault tolerant mode, 3-2, 3-3
  - number of licenses required, 2-22
  - overview, 3-2
  - starting the daemons, 3-2
  - troubleshooting, 3-19
- Access Control files
  - license.HOSTID*, 2-16, 2-23, 3-11
  - license.log*, 3-11
  - options*, 3-11, 3-15
  - servers*, 3-11
- Access Control utilities
  - valid\_report*, 3-17
  - valid\_status*, 3-16
- access, group, 6-4
- active license server, 2-30
- add command, 5-8
- adding servers, 3-9
- Allegro (PCB) PATH variable, 2-15
- archiving, using tar, 5-14
- archiving designs, 6-27
  - using tar, 6-28

## B

- backslash (\) symbol, 3-21
- backup
  - See also* dump
  - design archive tapes, 6-27
  - utilities, UNIX, 5-3
- baud rate for printers, 7-29
- Bourne Shell, selection, 4-4
- Bourne shell command, starting the daemons, 3-21

## C

- C Shell, selection, 4-4
- C shell command, starting the daemons, 3-21
- cabling, A-11
- cd command, 5-6
- changing permissions, using chown, 2-15
- chmod command, 6-8, 6-9, 6-10
- chown command, 2-15
- commands, 6-13
  - add, 5-8
  - cd, 5-6, 5-7
  - chmod, 6-8, 6-9, 6-10, 6-12
    - chmod g+w, 6-12
    - chmod o-rwx, 6-12
    - chmod og-rwx, 6-13

## commands, con't.

- chgrp, 2-15
- chown, 2-15
- df, 6-21
- du, 6-26
- dump, 5-5, 5-10
- export, 4-16
- find, 6-9
- hardcopy, 6-25
- hostid, 2-16
- kill, 3-8
- kversion, B-3
- ls, 4-12, 5-7, 6-10, 6-11, 6-14
- mkdir, 6-17
- more, 6-10
- mount, 2-12, 2-30, 6-17
- mount -a, 2-30
- newfs, 5-10
- passwd, 4-8, 4-10
- ps, 2-26, 3-8
- ps waux, 3-21, 6-25
- pwd, 5-7
- quit, 5-9
- rdump, 5-10
- restore, 5-4, 5-6
- rm, 4-11
- set, 4-14
  - set noascii, 6-24
  - setenv, 4-13
- shutdown, 6-13
- status, 7-23
- tar, 2-17, 3-9
  - tar -f, 5-15
  - tar c, 5-15
  - tar x, 5-15

- umask, 6-10, 6-12

- umask 002, 6-7, 6-12

- umask 007, 6-12

- umask 077, 6-13

- yppasswd, 4-8, 4-10

communication port, reserving, 2-24

configuring custom kernel, B-2

configuring disk space, 2-4

configuring servers, 2-10

copying designs, backup, 6-27

creating accounts

- for users, 4-2

- with Yellow Pages, 4-2

creating the lib account, 2-14

creating the links, 2-14

creating the Valid directories, 2-13

creating user accounts, selecting shell

- Bourne, 4-4

- C, 4-4

.cshrc file, 2-15

custom kernel configuration, B-2

## D

daemons

- license manager (*lmgrd*), 2-23, 3-21

- valid1*, *valid2*, 2-24

daemons line printer, 7-27

daily backup, example, 5-5

dd, backup utility, 5-3

default disk partitions, 2-8

---

---

default file permissions, 6-11

default files

- .cshrc*, example, 4-14
- .login*, example, 4-13
- .profile*, example, 4-16

default user files, 4-12

design directories, mounting, 6-17

design directory, example, 6-3

*dev/vplot* termcap entry, 7-7

*dev/vprint* termcap entry, 7-7

df command, 6-21

directories

- bin*, 2-5
- creating, 6-4, 6-17
- design, 6-3
- drawing, 6-3
- /home/valid*, 2-5
- remote mounting, 6-17
- SCALD, 6-15
- spooling, 7-9
- using mkdir, 6-17

directory hierarchy, 6-2

directory name, 6-16

disk partitions, 2-8

disk space

- file system maintenance, 6-21
- requirements, 2-4

disk space requirements

- applications, 2-6
- libraries, 2-7

disk usage, checking, 6-26

drawing name, rules, 6-16

du command, 6-26

dump, backup utility, 5-3

dump command, 5-5, 5-10

## E

editing templates, 2-25

editing the DAEMON entries, 3-7

editing the *servers* file, 2-27

encryption, 2-24

error file, hardcopy, 7-25

error messages, message file, 2-20

errors in installation, 2-20

case-sensitivity, 6-10

*etc* files

- /exports*, 2-28
- /fstab*, 2-12, 2-29
- /gettytab*, 7-30
- /group*, 2-14
- /passwd*, 2-14
- /printcap*, 7-3
- /rc.local*, 2-25
- /rc.valid*, 2-25, 3-8
- /services*, 2-24

export command, 4-16

external cabling, A-12

**F**

fault tolerance, 3-2  
  license distribution, 3-4

fault tolerant mode, access control, 3-3

FEATURE entry, license file, 2-24

feature names, 3-16, 3-17  
  table of, D-2

file ownership, security, 6-5

file restoration. *See* restore

file servers (non-access) 2-31

file system maintenance, 6-21  
  disk space, 6-21  
  maintaining root, 6-22  
  user files, 6-23

files  
  .cshrc, 2-15  
  errorlog, 7-25  
  /etc/exports, 2-28  
  /etc/fstab, 2-12, 2-29  
  /etc/group, 2-14  
  /etc/passwd, 2-14  
  /etc/rc.local, 2-25  
  /etc/rc.valid, 2-25, 3-8  
  /etc/services, 2-24  
  gettytab, 7-30  
  hosts.equiv, 7-4  
  license.HOSTID, 2-23  
  lpd-errs, 7-26  
  options.template, 3-5  
  printcap, 7-3  
  .profile, 2-15  
  .rhosts, 4-2

files, con't.

*servers.template*, 2-27  
  *services*, 7-27  
  */tmp/vinstall.msg*, 2-20, C-3  
  */usr/tmp/license.log*, 3-13

find command, 6-9

forgotten passwords, 4-10

**G**

group ID numbers, 4-4

**H**

hardcopy, 6-25

  error log file, 7-25

hardcopy command (GED), 7-19

*/home/valid*, linked to */usr/valid*, 2-14

host ID number, identifying the server,  
  2-16

hosts, 7-4

**I**

ID numbers, group and user, 4-4

identifying the server, 2-16

IKON interface board, base address,  
  A-5

installation, overview, 2-2

  error messages, 2-20

  first-time, 2-13

  requirements, 2-2

  using vscrip, 2-17

---

---

- installing new license file, 3-9
- installing Sun3 and Sun4, 2-21
- installing the Valid applications, using  
vscript, 2-16
- installing the Valid kernel, B-3
- installing Valid software, first-time,  
2-13
- interface board
  - installation, A-2
  - jumper settings, A-7
  - switch settings, A-3
- internal cabling, A-11

## L

- lib, adding a lib account, 2-14
- libraries
  - disk space, 2-7
  - file permissions, 6-18
  - location of, 2-5
  - lsttl, 6-18
  - master library, 6-19
  - master.lib*, 6-19
  - ownership, 6-18
  - SCALD, 6-18
- license availability, fault tolerance, 3-4
- license file, DAEMON entries, 3-7
- license manager, starting the daemons,  
2-26
- license manager daemon, 2-23
  - killing the process, 3-8

- license.HOSTID* file, 2-23
  - changing location, 3-13
  - new license file installed, 3-9
  - reinstallation, 3-13
  - sample diagram, 3-11
  - syntax, 2-23
- license.log* file, sample, 3-13
- linking the tools directories, 2-14
- lmgrd*, license manager, 2-23
- location
  - of Valid libraries, 6-2
  - of Valid software, 6-2
- location of applications, 2-5
- lpc utility, 7-21
- lpd daemon, 7-27
- lpd-errs*, file, 7-26
- ls -a command. *See* ls command
- ls command, 4-12, 5-7, 6-10
  - ls -l, 6-11, 6-14

## M

- MAIL variable, 4-13
- maintaining file system, 6-21
- master library, 6-19
  - master.lib*, example, 6-19
- maximum number of licenses, 2-33
- message file, errors during installation,  
2-20

mixed architecture  
  copying the servers file, 2-28  
  installation, 2-21  
  installing new license file, 3-9  
  reserving licenses, 3-6

mkdir command, 6-17

mkdir utility, 2-13

mknewpentry, 4-5

mknod, A-15

mkscaldusr, 4-5

mkusr  
  for design directories, 6-4  
  on Yellow Pages server, 4-2  
  problems with, 4-5  
  system requirements, 4-2  
  troubleshooting, 4-5

mkusr utility, 4-2

mkvaliddevices, A-15, B-5

more command, 6-10

mount command, 2-12, 6-17

mounting, remote directory, 6-17

mounting file servers, 2-31

mounting file systems, 2-30

multiple client configuration  
  making the directories, 2-29  
  mounting the server, 2-29

multiple server configuration  
  starting the daemons, 2-26  
  license availability, 3-4

## N

network plotting across, 7-4

new license file, installing, 3-9

newfs command, 5-10

number of licenses, 2-24, 2-33

## O

*options* file, sample, 3-15

*options.template* file, 3-5

ownership, 2-15

## P

package selection, 2-19

passwd command, 4-8, 4-10

password file, create new entry, 4-5

password protection, 2-16

passwords  
  changing, 4-10  
  changing root, 4-10  
  forgotten, 4-10  
  using passwd, 4-8

pen, plotters, 7-14

pen plotters, HP, 7-14

- 
- 
- permissions
    - default, 6-11
    - default file, 6-4
    - execute, 6-10
    - libraries, 6-18
    - program, 6-13
    - read, 6-10
    - script, 6-13
    - set-group-ID, 6-14
    - set-user-ID, 6-14
    - special
      - permission bits, 6-14
      - sticky bit, 6-14
    - strict, 6-13
    - using `ls -l`, 6-11
    - write, 6-4, 6-10
  - plot spooling, 7-3
  - plotters, 7-7
    - pen, 7-14
  - plotting
    - error logs, 7-25
    - ged, 7-19
    - monitoring the queues, 7-21
    - printcap file, 7-3
    - queues, 7-21
    - remote, 7-4
    - troubleshooting problems, 7-24
  - plotting across the network, 7-4
  - primary license server, 2-23
  - printcap file, 7-3, 7-25
    - plotting, 7-3
    - syntax, 7-3
    - troubleshooting, 7-24
  - printers
    - baud rate, 7-29
    - laser, 7-17
    - postscript, 7-17
  - .profile* file, 2-15
  - project security
    - simple, 6-4
    - strict, 6-4
  - `ps -ax` command, 6-25
  - PSI variable, %, 4-16
  - `pwd` command, 5-7
- Q**
- quit command, 5-9
  - quitting the installation, 2-18
- R**
- rc.valid*, 2-25, 3-8
  - `rdump` command, 5-10
  - remote mounting, 6-17
  - removing users, 4-11
  - report generator. *See* `valid_report` utility
  - required disk space, 2-6
    - libraries, 2-7
  - RESERVE entry, options file, 3-5
  - reserving communication port, 2-24
  - reserving licenses, 3-5
  - restarting the daemons, 2-28, 3-7

restore, i/f/v/b flags, 5-7  
restore command, 5-6  
restoring file systems, 5-10  
restoring files, 5-6  
rm -r command. *See* rm command  
rm command, 4-11  
root  
    *See also* super-user  
    file system maintenance. *See* super-user  
root password, changing, 4-10  
running vscrip, 2-18

## S

sample options file, 3-15  
saving designs, 6-27  
SCALD directories, 6-15  
SCALD directory, example, 6-15  
SCALD/UNIX directories, relationship, 6-15  
script permissions, 6-13  
security  
    file ownership, 6-5  
    simple, 6-4  
    strict, 6-4  
selecting packages, 2-19  
server configuration, 2-10  
servers file, 2-27  
    updating information, 2-28  
set command, 4-14  
set noascii command, 6-24  
setenv command, 4-13  
setting the PATH variable, 2-15  
shutdown command, 6-13  
simple mode, 3-3  
software installation, 2-2  
software organization, 2-5  
    standalone systems, 2-9  
    two-disk systems, 2-9  
special permissions, user-ID, group-ID  
    sticky bit, 6-14  
spooling plots, 7-3  
standalone systems, software organization, 2-9  
standby license servers, 2-30  
starting the daemons, 2-26, 3-2  
    Bourne shell command, 3-21  
    C shell command, 3-21  
startup files  
    .cshrc, 4-5  
    .login, 4-5, 4-5  
    .profile, 4-5  
    rc.valid, 2-25, 3-8  
    super-user, 4-7  
startup.ged file, 4-5  
status command, 7-23  
strict permissions, 6-13

---

---

Sun Microsystems  
  Sun documentation, 1-2, 2-2  
  Sun executables, 2-18  
    location of, 2-10  
  Sun installation, 2-11  
  Sun operating systems, 2-11  
  Sun programs, requirements, 2-11  
Sun VME-VME adapter, installation,  
  A-2  
Sun3, Sun4 licenses, 3-5  
super-user  
  creating accounts, 4-2  
  file system maintenance. *See* root  
  mounting remote directory. *See* root  
  startup files, 4-7  
switch/jumper locations, A-4  
symbolic link, 2-10  
symbolic links, creating, 2-14  
syntax, *printcap* file, 7-3  
system maintenance, user files, 6-23  
System Tools, installing, 2-21

## T

tar  
  archiving, 5-14  
  archiving designs, 6-28  
  backup utility, 5-3  
  c/v/f/b flags, 5-15  
tar command, 2-17, 3-9, 5-14  
  tar c command, 5-15  
  tar x command, 5-15

TCP/IP port, 2-24  
template files, editing, 2-25  
text editor, vi, 6-10  
*/tmp/vinstall.msg* file, error messages,  
  2-20  
troubleshooting access control, 3-19

## U

UNIX files, location of, 6-2  
UNIX utilities, backup, 5-3  
umask command, 6-10, 6-12  
  *.login* file, 6-7  
  *.profile* file, 6-7  
unavailable licenses, 2-33  
UNIX directories  
  creating, 6-4  
  location of, 6-2  
updating the servers file, 2-28  
user accounts  
  creating, 4-2  
  ID numbers, 4-4  
user file systems, file system maintenance, 6-23  
user ID numbers, to find, 4-4  
user passwords, password file, 4-8  
users, removing from system, 4-11  
utilities, 7-21  
  dd, 5-3  
  dump, 5-3  
  lpc, 6-25

utilities, con't.  
  mkdir, 2-13  
  mknewpentry, 4-5  
  mknod, A-15  
  mkscaldusr, 4-5  
  mkusr, 4-2, 6-4  
  mkvalddevices, A-15  
  mkvaliddevices, B-5  
  tar, 5-3, 5-14  
  valid\_report, 3-17  
  valid\_status, 3-16  
  valid\_status -a, 3-7  
  vinstall, C-2  
  vscript, 2-12, 2-16, 2-17  
  vscript -p, 3-9

## V

Valid applications  
  disk space, 2-6  
  location of, 2-5  
  physical location of, 2-13

Valid executables, location of, 2-5

Valid products, feature names, 3-16,  
  D-2

Valid Real Products, B-2

Valid software installation, 2-2  
  using vscript, 2-12

Valid software organization, 2-5

valid\_report utility, 3-17  
  sample report, 3-18

valid\_status -a, 3-7

valid\_status utility, 3-16  
  sample display, 3-16

valid1 daemon, 2-24

valid2 daemon, 2-24

Versatec plotter configuration, B-2

vi, text editor, 6-10

vinstall  
  for Sun3 workstations, 2-16  
  for Sun4 workstations, 2-16

vinstall errors, C-3

vinstall utility, C-2

VME-VME adapter, installation, A-13

vplot termcap entry, 7-7

vpq command, 7-21

vprint termcap entry, 7-7

vprm command, 7-21

vscript, 2-12

## W

write permission, using vi, 6-10

## Y

Yellow Pages, creating accounts, 4-2

yppasswd command, 4-8, 4-10

## Symbols

% primary prompt, 4-16

\ backslash, 3-21, 6-29

> restore prompt, 5-7