

## MEMORANDUM

FROM: John Sybalsky  
DATE: September 15, 1991  
RE: Release 2.0 of Medley for the Sun Workstation

Enclosed is the software and documentation for Release 2.0 of Medley for the Sun Workstation. The package consists of the following:

- Tape containing the revised software.
- Release Notes, providing warnings and information important to the successful running of the software, followed by fixed bugs.
- *Medley for the Sun Workstation User's Guide*, encompassing release contents, instructions for installing Release 2.0, and information on using it. This *Guide* has been completely reorganized, and information about using the new installation script has been added.
- *Lisp Library Modules* revised pages, reflecting additions to the prior issue (replace the old sections with the corresponding new pages).



*Venue*



*Medley for the Sun Workstation<sup>®</sup> User's Guide*

*Release 2.0  
March, 1991*



*Venue*

*Medley for the Sun Workstation® User's Guide*

Release 2.0  
June, 1991

---

Address comments to:  
Venue  
User Documentation  
1549 Industrial Road  
San Carlos, CA 94070  
415-508-9672

---

## MEDLEY FOR THE SUN WORKSTATION® USER'S GUIDE

Release 2.0

June, 1991

Copyright © 1990, 1991 by Venue.

All rights reserved.

Envos is a trademark of Envos Corporation.

Medley is a trademark of Venue.

Xerox® is a registered trademark and InterPress is a trademark of Xerox Corporation.

UNIX® is a registered trademark of UNIX System Laboratories.

PostScript is a registered trademark of Adobe Systems Inc.

The following are trademarks of Sun Microsystems, Inc.:

Sun, Sun-2, Sun-3, Sun-4, SunOS, and SPARCstation are trademarks of Sun Microsystems, Inc.

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

The X Window System is a trademark of the Massachusetts Institute of Technology.

IBM is a trademark of International Business Machines, Inc.

MIPS is a registered trademark of MIPS Computer Systems, Inc.

Copyright protection includes material generated from the software programs displayed on the screen, such as icons, screen display looks, and the like.

---

The information in this document is subject to change without notice and should not be construed as a commitment by Venue. While every effort has been made to ensure the accuracy of this document, Venue assumes no responsibility for any errors that may appear.

Text was written and produced with Venue text formatting tools; Xerox printers were used to produce text masters. The typeface is Classic.

# TABLE of CONTENTS

---

<b>Preface</b> .....	ix
<b>1. Introduction</b> .....	1
What Medley Requires.....	1
Hardware .....	1
Input/Output Devices.....	1
Bitmap Display .....	1
Printers.....	1
Software Requirements.....	2
Medley and Other Applications.....	2
Display Usage .....	2
Processor Usage.....	2
System Components.....	2
Medley Device-Naming Conventions .....	3
Notation Conventions .....	3
Fonts.....	3
Packages.....	3
Prompts .....	3
Compatibility.....	4
Sysout Compatibility .....	4
Compiled-File Compatibility.....	4
End-of-Line Convention .....	4
Release Contents.....	4
Documentation.....	4
Software .....	5
LispUsers Modules .....	5
<b>2. Software Installation</b> .....	7
Getting Ready to Install Medley .....	7
Ensuring Adequate Swap Space .....	8
Installing Medley for Shared Use.....	8
Installing Medley Software.....	9
Using the Installation Script's Menu .....	9
Getting a Copy Protection Key .....	11
Changing Configurations or Adding Options .....	11

<b>3. Getting Started</b> .....	13
Getting Ready .....	13
Running Medley .....	13
Running Medley Directly .....	13
Using the Medley Shell Script .....	13
Where Medley Looks for Your Sysout .....	15
Where Medley Looks for Your Site Initialization File .....	15
Medley and X Windows.....	16
Starting X Windows.....	16
Running Medley Remotely.....	16
The Medley Window .....	17
Environment Variables .....	17
Keyboard Interpretation.....	17
Sun Type 3 Keyboard .....	18
Sun Type 4 Keyboard .....	19
<b>4. Using Medley on the Sun Workstation</b> .....	21
Setting Up a Site Init File .....	21
Setting Up a Personal Init File .....	22
Saving Your State .....	22
Sun-Specific Environment Functions.....	24
System Environment Functions and Variables .....	24
VM Functions.....	25
Stopping Lisp Temporarily .....	25
Login Functions .....	26
Environment Inquiry.....	26
Display and Keyboard Functions and Variables .....	27
Timers and Clocks .....	27
Miscellaneous Operational Differences .....	28
Console Messages .....	28

---

<b>5. Medley File Systems</b> .....	31
File Naming Conventions .....	31
Hosts that Medley Supports .....	31
Using SunOS Files from Medley .....	31
Common {DSK} and {UNIX} Naming Conventions.....	32
{DSK} Naming Conventions.....	33
Version Numbering.....	33
Pathnames.....	35
{UNIX} Naming Conventions.....	35
Directories.....	36
Directory Enumeration.....	36
Directory Creation .....	36
Directory Deletion.....	37
Open File Limit.....	37
Default Pathname.....	37
File Attributes .....	38
File Variables .....	39
File System Errors .....	39
<b>6. Error Recovery</b> .....	41
URAITD .....	41
Entering URAID .....	41
Conventions.....	41
URAITD Commands .....	42
Displaying a Stack .....	42
Viewing Frames from a Stack .....	43
Miscellaneous .....	44
Other Fatal Error Conditions .....	44
Lisp Errors.....	44
Errors While Running Medley .....	44
Xerox Workstation-Specific Errors .....	45
Virtual Memory Errors.....	45

<b>Appendix A. Installation Hints</b> .....	A-1
Medley Shell Variables .....	A-1
Running on Multiple Workstations .....	A-1
Installation for Sites with Sun-3 and Sun-4 Workstations .....	A-1
Using a "runlde" on Multiple Workstations .....	A-2
Configuring the Software .....	A-2
Relinking .....	A-3
Enabling PUP/XNS Ethernet .....	A-3
Using NIS to Manage the Keys for Multiple Workstations .....	A-3
<b>Appendix B. Verifying the Installation Tape's Validity</b> .....	B-1
Output .....	B-1
Examples .....	B-1
<b>Appendix C. Layout of Installation Tape Files</b> .....	C-1
Layout of Installation Tape .....	C-1
Font Directories .....	C-3
Manually Extracting Files from the Installation Tape .....	C-4
<b>Appendix D. Differences between Xerox Workstations and                   the UNIX Version of Medley</b> .....	D-1
Local Disk and Floppy Functions .....	D-1
Library Modules Not Supported on the Sun .....	D-1
TCP, TCPCHAT, etc. ....	D-1
DLRS232C, DLTTY .....	D-2
KEYBOARD, VIRTUALKEYBOARD .....	D-2
<b>Glossary</b> .....	GLOSSARY-1
<b>Index</b> .....	INDEX-1



[This page intentionally left blank]

# LIST of FIGURES

## LIST of TABLES

---

Figure	Page
2-1 Sample Installation Options Menu .....	10
3-1 Sun Type 3 Key Numbering.....	18
3-2 Sun Type 3 Left Key Pad .....	18
3-3 Sun Type 3 Right Key Pad .....	18
3-4 Sun Type 3 Center Key Pad Interpretation .....	19
3-5 Sun Type 4 Key Numbering.....	19
3-6 Sun Type 4 Left Key Pad .....	20
3-7 Sun Type 4 Right Key Pad .....	20
3-8 Sun Type 4 Center Key Pad Interpretation.....	20

Table	Page
C-1 Font Directories.....	C-3

[This page intentionally left blank]

# PREFACE

---

This *Guide* describes Medley release 2.0 for the Sun-3 and Sun-4 workstations and the SPARCstation: the release contents, instructions for installing the release, and information on using it.

## Audience

---

The *Medley For the Sun Workstation® User's Guide* is intended for users familiar with the Medley environment who want to use it on the Sun-3 or Sun-4 workstations or the SPARCstation. The *Guide* assumes that the user is already familiar with UNIX and SunOS concepts. The system administrator of a Sun system or network should read this *Guide* to ensure the correct installation of the Medley software.

Chapter 1 of this manual gives an overview of the product and its internal architecture, and is of interest to all users of the system.

System administrators should read Chapter 2, Software Installation; and Chapter 3, Getting Started. These chapters guide the administrator through the process of installing Medley 2.0 and configuring it on the Sun Workstation. Experienced Lisp users may want to configure the software; this procedure is described in Chapter 4.

Users already familiar with the Lisp environment on Xerox workstations should find Chapter 1 and Chapters 3 through 6 useful. These chapters describe the operation of the system after it has been installed as well as those functions and operations which are specific to the Sun Workstation.

## Using This Manual

---

**Chapter 1, Introduction**, describes the hardware, input/output devices, and software needed to run Medley on a Sun Workstation; describes Medley and how it works with other applications; lists the system components; introduces pertinent SunOS and UNIX conventions used throughout the *Guide*; explains Medley's compatibility; and lists the contents of the release.

**Chapter 2, Software Installation**, contains the installation and software configuration procedures.

**Chapter 3, Getting Started**, explains how to set up a site initialization file and install the X Windows System. It also shows the keyboard configuration and has instructions for getting started in Lisp on the Sun Workstation.

**Chapter 4, Using Medley on the Sun Workstation**, describes how specific Lisp functionality works on the Sun.

**Chapter 5, Medley File Systems**, discusses the file conventions that need to be followed when running in Medley on a Sun Workstation. Differences in Lisp file attributes and variables are also discussed.

**Chapter 6, Error Recovery**, describes the diagnostic error recovery program URAID. This chapter explains how to recover from fatal error conditions and lists specific Lisp errors that may be encountered when running Medley on the Sun.

**Appendix A, Installation Hints**, contains additional notes to help configure Medley, and includes a complete description of the installation script.

**Appendix B, Verifying the Installation Tape's Validity**, tells how to validate the contents of the tar tape.

**Appendix C, Layout of Installation Tape Files**, includes a listing of the tar tape directories and the font directories.

**Appendix D, Differences between Xerox Workstations and the UNIX Version of Medley**, includes functions for controlling device-specific behavior of the Xerox 1100 series workstation disk drives. It also describes the library modules not supported on the Sun.

The **Glossary** provides definitions of SunOS, UNIX, and Lisp terms used in this *Guide*.

Medley is a Venue product which was built on the Xerox Lisp environment. It provides an integrated programming environment consisting of Interlisp-D and Common Lisp, a windowing system, and a set of programs and utilities. Users not already familiar with the Xerox Lisp environment should try to become somewhat familiar with it before attempting serious development work.

## Supporting Documentation

---

The following reference documents are useful to have on hand during the installation process and when working in Medley on the Sun Workstation.

### Sun References

This literature from the Sun documentation set is useful during the installation and when running Medley on a Sun Workstation.

- *Installing UNIX on the Sun Workstation*
- *UNIX Interface Reference Manual*
- *SunOS Reference Manual*
- *Sun Software Technical Bulletin, March 1988*

### Venue Documentation

In addition to this *Guide*, the following documents describe the Medley system:

- Guy Steele, *Common Lisp, the Language, First Edition*, Bedford, MA: Digital Press, 1987
- *Interlisp-D Reference Manual, Volumes 1-3*, Koto Release
- *Xerox Common Lisp Implementation Notes*, Lyric Release
- *Lisp Documentation Tools*, Lyric Release
- *Lisp Library Modules*, Medley Release
- *Lisp Release Notes*, Medley Release

Templates for the Type 3 and Type 4 Sun keyboards are also part of the Medley documentation set.

New users of Medley receive, in the software kit, all the manuals listed above.

Users who are moving the Medley environment from a Xerox workstation to a Sun Workstation receive the following documentation in the software kit:

- *Lisp Release Notes*, Medley Release
- *Lisp Library Modules*, Medley Release
- *Medley for the Sun Workstation® User's Guide*
- Sun Type 3 and Type 4 keyboard templates

The manual *LispUsers' Modules*, Medley Release, which may be purchased separately, supplements the Medley release.

[This page intentionally left blank]

# 1. INTRODUCTION

---

Medley is an integrated programming environment, with support for the Interlisp and Common Lisp languages, an integrated windowing system, and a large collection of utilities and programs. It offers a mature and rich programming and development environment, as well as access to a large number of applications written for Interlisp, Interlisp-D, Common Lisp, and LOOPS.

Medley for the Sun Workstation has two versions, a Sun-3 version and a Sun-4 version, available on separate tapes. Medley 2.0 runs on the Sun-3 and Sun-4 workstations and the SPARCstation.

## What Medley Requires

---

### Hardware

Medley runs on Sun-3 and Sun-4 Workstations and the SPARCstation. It runs on both standalone workstations and diskless workstations linked to servers.

Medley on the Sun-3 Workstation requires the MC68881 floating-point coprocessor chip. On the Sun-4 Workstation, the Weitek 1164/1165 coprocessor is optional, but recommended.

For adequate performance, we recommend at least a 20 MHz 68020 (Sun 3/60 or 3/260), a 14 MHz SPARC (Sun 4/110 or 4/260), or a SPARCstation.

Except under X Windows, reasonable interactive performance can be expected with 8 megabytes (MB) or more of RAM. Smaller configurations of diskless workstations have been tested, but performance suffers. When using X Windows software, allow an additional 4 MB.

Naturally, larger applications will benefit from more memory. Medley's maximum working set is approximately 40 MB.

### Input/Output Devices

Medley provides access to the Sun's input/output devices, such as display, keyboard, mouse, and file systems. It also provides access to PUP and XNS Ethernet services directly.

### Bitmap Display

Medley supports all standard Sun displays and frame buffers.

### Printers

You can print on Xerox Interpress printers using the XNS networking protocols. The FX80 printer also works via the RS232 port.

If you have a PostScript printer, you can use the LispUser modules PostScriptStream and UNIXPrint to direct output to your printer.



## Software Requirements

Medley on the Sun-3 Workstation requires SunOS versions 3.2, 3.4, 3.5, 4.0, or 4.0.3. On the Sun-4 Workstation, Medley requires SunOS version 4.0, 4.0.3, or 4.1.

If you plan to run Medley under X Windows, you will need X11, version 4, or Motif.

NOTE: Medley's XNS Ethernet code will not work if you are running SunOS 3.5 configured for Kernel XNS Ethernet Support or Alpine.

## Medley and Other Applications

---

### Display Usage

When Medley is running alone, it takes over the entire display screen. When running under X, Medley uses one window as its screen; Medley maintains its own windows within that single window. Medley cannot run at the same time as Suntools or Open Windows.

### Processor Usage

Medley runs its own process scheduler; as far as the UNIX scheduler is concerned, Medley is always running. For this reason, other heavy computational jobs on the same Sun Workstation will not get as good performance as they would competing with conventional UNIX interactive applications.

Similarly, Medley may not have adequate interactive performance if it is competing with other compute-bound tasks on the same machine.

For these reasons, we recommend that Medley be used on machines that are set up primarily for a single user.

## System Components

---

Functionally, Medley consists of the following components:

emulator	A SunOS-executable program, which performs several functions. It executes the Interlisp-D virtual machine instruction set compatibly with the microcode of the Xerox 1100 series workstations. (This instruction set allows memory-efficient representation of Interlisp and Common Lisp programs.) It also provides access to the host machine's I/O (display, keyboard, file system), and executes some system functions directly.
sysout	A virtual memory image (the <i>sysout</i> ) containing both byte-code-compiled Lisp functions and data structures. The <i>sysout</i> provided can be used both on the Sun Workstation and on the Xerox 1100 series machines.
library	Files of compiled Lisp code and data structures.
fonts	Data describing the "looks" of printed characters used by Medley's graphics, windowing, and hardcopying subsystems. Font directories are in three groups: display fonts, InterPress printer fonts, and Press printer fonts.

**checksum** A script that reports inconsistent files, the correct checksum values for the files, and an error message. The checksum of individual files can be generated with the UNIX command `sum filename`. Use this when Medley installs correctly but does not run.

## Medley Device-Naming Conventions

---

Medley for the Sun Workstation lets you interact with SunOS file systems (including file systems mounted from other machines) by using host device names. The two device names are as follows:

- `{DSK}` A host name which gives you access to the SunOS file system using Xerox workstation local disk conventions.
- `{UNIX}` A host name which gives you access to the file system using normal SunOS conventions.

The `{DSK}` device name provides an interface to the Sun Workstation for users who want to maintain compatibility with existing development tools and applications originally developed on a Xerox workstation. The `{UNIX}` device name provides a way for new applications to interact naturally with UNIX. Chapter 5 explains, in greater detail, some important exceptions and restrictions to the `{DSK}` and `{UNIX}` device name.

## Notation Conventions

---

Text marked by a revision bar in the right margin contains information that was added or modified since the last release. Fonts, packages, and prompts have the following types of notation.

### Fonts

**Bold text in TITAN font** indicates text you should type in exactly as printed.

Regular `TITAN font` text indicates what the system prints on your workstation screen. Lisp functions and variables and UNIX files and programs are also shown in `TITAN FONT`.

*Text in Classic italics* indicates variables or parameters that you should replace with the appropriate word or string.

### Packages

Most Lisp symbols have a Lisp package qualifier; the `INTERLISP` package (`IL:`) is the default when no package qualifier is shown.

### Prompts

All examples which include SunOS dialogues use the following conventions for the SunOS prompt.

A number sign (`#`), part of the system prompt, indicates that you are logged on as `root` or is running `su`; for example,

prompt#

A percent sign (%), part of the system prompt, indicates that a user other than `root` is logged on; for example,

prompt%

## Compatibility

---

The Medley release on the Sun Workstation is designed for maximum compatibility with the Xerox workstation implementations. However, when moving applications to the Sun Workstation note the differences in end-of-line conventions and techniques for moving files.

### Sysout Compatibility

Sysouts of the same version are compatible with all machine types. But a sysout generated on a Sun Workstation cannot be used on a Xerox workstation.

NOTE: You cannot mix different versions of sysouts and emulators.

### Compiled-File Compatibility

Code compiled in a Medley 1.0, 1.1, 1.15 or 1.2 sysout cannot be loaded into Medley 2.0 sysouts, nor can code compiled in Medley 2.0 be loaded onto earlier sysouts. Code compiled for Medley 2.0 on a Xerox workstation cannot be loaded into Medley running on a Sun. The opposite is not possible either.

### End-of-Line Convention

Some care must be taken in moving files to and from Xerox workstations, since the default end-of-line convention in UNIX is to terminate lines with the line feed (LF) character, while, traditionally, Medley has terminated lines with the carriage return (CR) character. In particular, if you use some other file transfer mechanism, such as FTP or Kermit, be careful to transfer `.TEDIT`, `.DFASL`, and `.LCOM` files in binary mode.

In Medley on the Sun Workstation, the default end-of-line convention for all text files is line feed (LF). The default end-of-line convention for all binary files is carriage return (CR); this is because CR (ASCII 13) is used internally in the system.

## Release Contents

---

The release distribution contains the following documentation and software.

### Documentation

The Medley documentation kit for users moving from a Xerox workstation to a Sun Workstation contains:

- *Lisp Library Modules*, Medley Release
- *Lisp Release Notes*, Medley Release
- *Medley For the Sun Workstation® User's Guide*

- Sun Type 3 and Type 4 keyboard templates.

New customers also receive the following:

- *Interlisp-D Reference Manual, Volumes 1-3*, Koto Release
- *Xerox Common Lisp Implementation Notes*, Lyric Release
- *Lisp Documentation Tools*, Lyric Release
- Guy Steele, *Common Lisp, the Language, First Edition*

All users can also purchase this document:

- *LispUsers' Modules*, Medley Release

## Software

The software release is available on either a ¼-inch tape cartridge or a ½-inch 9-track tape. The software release is specific to the Sun architecture (Sun 3 or 4) for which you purchased Medley, but contains multiple SunOS versions. This tar tape contains the directories listed below. (See Appendix C for details of the directory contents.)

```
./install-medley
./medley
./install.sunos3/
./install.sunos4/
./install.sunos4.1/
./lisplibrary
./checksumdir
./lispsysouts
./fonts/display
./fonts/interpress
```

## LispUsers Modules

The Medley version of LispUsers Modules is a software supplement to Medley for the Sun Workstation. This is software written by our users which you may purchase separately. The support for these modules comes from each module's author; Venue has no commitment to support LispUsers' modules.

Two LispUsers Modules are particularly useful when you are running Medley on a Sun Workstation. For those users with Postscriptstream printers for output, the PostScript module is particularly useful. The LispUsers module RPC implements Sun remote procedure calls.

[This page intentionally left blank]

## 2. SOFTWARE INSTALLATION

---

This chapter describes how to install Medley on Sun Workstations.

To install the Medley Release on a Sun Workstation, you need the following:

- Release tape
- Medley documentation kit for the Sun Workstation.

### Getting Ready to Install Medley

---

Medley includes a shell script for automatic installation. The script infers as much as it can about your host and network, but will prompt you for answers when needed. Once it has collected the necessary information, it installs only those parts of Medley that you really need.

Do not worry if you forget something. You can run the installation again, and pick up any items you missed.

Before installing Medley, you should gather some facts about the hardware and network environment on which you will be using Medley. The following checklist will help you.

- Do you have the correct release tape correct for the kind of machine on which you plan to run?

The tape is labeled either Sun-3 or Sun-4/SPARCstation.

- Where is the tape drive you will be using?

Does your Sun have a 1/4-inch tape drive? If not, you need a Sun with a 1/4-inch tape drive on your network. You will need to know the host name for that machine.

- Does your system have sufficient swap space (45 MB) for Medley?

If you are not sure, see the section below for instructions.

- Are you installing Medley for a single user, or will several users be sharing this copy?

If it is for shared use, you will want to install Medley on a public directory on a shared server. For that, we recommend using `/usr/share/lde` as the directory name. You will need to be running on the server when you do the installation, and you will probably need to log in as `root`. Check with your System Administrator for details.

- Do you have enough disk space free ?

You need to select a file system with enough disk space to install the software. A minimal installation requires approximately 12 MB, and a full installation will require up to 23 MB. Use the UNIX command `df` to find one.

- Have you selected an installation directory?

The directory must be on a file system with enough space. For individual use, we recommend `/usr/local/lde`.

---

### CAUTION

If the installation directory contains a previously installed version of Medley, some of the older files will be replaced with new ones.

---

- Do you have write permission to create the installation directory and to write files into it?
- Will you be running on X Windows?  
If you are, you will need X11 R4 or Motif. If you have X Windows installed, the utility will install the software necessary to run Medley under X11.
- Will you be running XNS (Xerox Network Services) or PUP protocols?
  - If so, you must be logged in as `root` during the installation.
  - If so, be sure you are not running SunOS 3.5 Kernel XNS Ethernet Code ("Alpine Kernel").
- If your host is networked, do you have XNS (Xerox Network Services) servers on it?  
If so, you will want to install XNS-relevant software. Also, if you have XNS Print Services and InterPress printers on your network, you might want to install InterPress fonts, allowing you to use an InterPress printer from within Medley. The default is not to install any XNS-relevant software.
- Is this a new installation, or are you upgrading from an earlier version of Medley?  
If you are upgrading, you only need to install the `sysout`, the appropriate emulator, and library files. If you are making a new installation, you will need at least the display fonts as well.

### Ensuring Adequate Swap Space

Medley requires 45 MB of swap space on top of the normal swap space requirements. Check swap space using the `pstat` command:

```
prompt% /etc/pstat -s
37176k used (3176k text), 12920k free, 1344k wasted, 0k missing
max process allocable = 10224k
avail: 5*2048k 1*512k 4*256k 3*128k 6*64k 7*32k 7*16k 40*1k
4800k allocated + 2520k reserved = 7320k used, 64672k available
```

If you need more swap space, consult the *Sun Software Technical Bulletin, March 1988*, pages 335-36, for information on increasing the amount of available swap space.

### Installing Medley for Shared Use

If several people will be using Medley on different machines, it probably makes sense to install one copy and have people share it.

You will need to find one machine—probably your main file server—with enough disk space. You will also want to have the directory look the same to every user. We recommend calling the installation directory `/usr/share/lde`.

## Installing Medley Software

1. Log in under your username.

```
login yourname
```

2. Put the tape in the tape drive. The script will allow you to install from a tape drive on a remote host. If you are performing a remote installation, put the tape in the tape drive of the host.

3. Retrieve the installation utility from the tape, as follows.

- If the tape drive is on a different host, enter the following:

```
rsh remote-host dd if=/dev/rst0 | tar xf -
```

Replace *remote-host* with the name of the host on your network that has the tape drive you are using. This copies the file `install-medley` to your working directory.

- If the tape drive is local to your machine, type the following:

```
tar xf /dev/rst0
```

This copies the file `install-medley` to your working directory.

If you have any problems during this step of the installation, consult your local UNIX system administrator.

4. Run the installation utility:

```
install-medley
```

Each time you are prompted for information, the script will show you the default in [ ] brackets. Pressing the return key selects the default. To select a different option, type it.

After installing Medley, the script will offer to update two files which must refer to the installation directory. We recommend this. The two files are:

```
medley      A script for running Medley easily
```

```
site-init  A sample site-init file
```

You can stop the installation process at any time by typing ↑C (Control-C).

### Using the Installation Script's Menu

To choose an option from the Installation Options Menu, type at least the first three characters of the selection. Most of the menu lists items you might want to install. Choosing one of these options works as a toggle switch, either selecting or deselecting, depending on its previous setting. The other menu items act as commands when you select them. "OS version" lets you pick the SunOS version(s) for which you will need matching emulators. "Directory" lets you specify where to install Medley. "!" lets you use UNIX commands if you need to.

In Figure 2-1, the user has selected for installation the Sysout, Monochrome and X Windows emulators for SunOS 4.1, Display fonts, and Library modules. The menu shows that you need 15052.8 KB of disk space to finish the installation, but only 13002 KB are available. At this point you can either deselect an option to decrease the disk space requirements, or change the installation directory to one that has sufficient disk space.



```

<-----> Installation Options Menu <----->
----- Emulators -----
For one or several OS versions (At least one of monochrome,
color or X11-version is required for new installations)
  x Monochrome - 0.5 MByte
  - Color
  x X11-version - 0.6 MByte
  - XNS - allows use of XNS protocols
  - Object files - allows linking of Medley to other software
    OS version - Change versions. Selected: 4.1
----- Fonts -----
  x Display - 5.5 MByte (recommended)
  - Interpress
----- Sysout, Library & Checksum files -----
  x Sysout - 5.1 MByte (required for new installations)
  x Library modules - 3 MByte (recommended)
  - Checksum files
----- Commands -----
Directory - Change location of installation directory.
  -- Current: /usr/share/lde 13002 KB
  -- Disk-space(KByte) Available:13002 Needed:15052.8
!<Unix command> - Execute a Unix command
? or Help - Show menu instructions
Redraw - Redisplay this menu
None - Unmark all options
All - Mark all options
Continue installation
Quit installation
Select [Directory]:

```

**Figure 2-1. Sample Installation Menu**

Because of the disk space shortage, the script has offered [Directory] as the default next command. If that is what you want, just press Return. Otherwise, type some other command.

For example, to deselect Library modules, type:

```
Select [Directory]: lib
```

Alternately, to find a filesystem with enough disk space, issue the following command:

```
Select [Directory]: !df
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/sd0a	7548	4935	1858	73%	/
/dev/sd0g	75106	51704	15891	76%	/usr
/dev/sd1g	47999	21898	21301	51%	/home
king:/shared	416928	349202	26042	93%	/share

```
Select [Directory]:
```

To change the installation directory, type:

```
Select [Directory]:dir
```

At the new prompt, give the directory name:

```
Where do you want to install Medley? [/usr/share/lde]: /share/lde
```

If the directory does not exist, the script will attempt to create it.

---

## Getting a Copy Protection Key

---

Before starting Medley, you must first obtain a host access key from Venue. This key enables the use of Medley on one workstation. The software cannot run without the key. You must have one key for each host on which you wish to run. Note that your current host access key *will* work if you move from one version of SunOS to another, or if you upgrade from an earlier version of Medley to Medley 2.0.

To obtain a key:

1. Get the host ID of the machine on which you intend to run Medley by typing on that machine

```
prompt% hostid
```

```
310002f6
```

2. Call Venue at 800-228-5325 between 9:00 a.m. and 4:30 p.m. PST. Outside the United States, call your local distributor.
3. Ask for a host access key, giving the Venue representative your host ID. Venue provides you with a host access key, which you need during software configuration, below.
4. If you plan to use the Medley startup script, you will be prompted for the key the first time you invoke it. The script will automatically save the key into a file for future reference.

---

### CAUTION

Depending on your license agreement with Venue, your host access key may have an expiration date. After that date, your key is no longer valid.

---

---

## Changing Configurations or Adding Options

---

If you forgot to install something or need to add a new emulator, you can rerun the installation, and select only the new things you need. The installation script does not remove things.

This need arises most often when you start running X Windows and need the new emulator.

First, decide what you need to install. Then type `install-medley`. When you get to the Installation Options Menu, select all the new things you need and *deselect* everything else. Continuing the installation from there just adds the new items.

If you have Sun-3s and you just got a SPARCstation, all you need from the new installation are the proper emulators. Everything else is the same.

[This page intentionally left blank]

## 3. GETTING STARTED

---

### Getting Ready

---

To prepare your system to run Medley, use the following checklist:

1. Exit `suntools` or any other windowing system, unless you plan to run under X (in which case you can leave X running). Medley provides its own window system and must not run simultaneously with others.
2. Kill all your user processes (these have console as the control tty). Check to make sure you have killed any "selection\_svc" process. If you do not perform this step, certain error messages from UNIX (e.g., file system full) cause those processes to print to the console, resulting in scrolling of the display.
3. Check for the directory for the software, and add it to your path, if necessary:

```
prompt# set path = ($path /usr/share/lde/install.sunosx)
```

You can also add this to your `.login` file.

### Running Medley

---

#### Running Medley Directly

Invoke Medley by typing the name of the program, e.g.,

```
prompt% lde optional-sysout -k 'thishost-key' [-m memory-size]
```

If you are using either Xerox XNS or Xerox PUP Ethernet protocols, type instead

```
prompt% ldeether optional-sysout -k 'thishost-key' [-m memory-size]
```

This, in turn, runs `lde` and lets it use the Ethernet directly.

*optional-sysout* is the name of a Lisp virtual memory image file (see the section Where Medley Looks for Your Sysout below). *thishost-key* is the key you obtained from Venue for the machine on which you are running.

If the sysout was created on a machine with a different size display, the image will appear garbled for several seconds. After Lisp starts running, it readjusts the display to the current size.

The `-m` flag lets you control the maximum amount of memory Medley will use. *memory-size* is a number in the range 8 through 32, in megabytes. (See the detailed explanation on page 25.)

#### Using the Medley Shell Script

The script will try to find a key, an appropriate emulator, and a sysout. The script relies on information about where the Medley software was initially installed on your system. (The installation script `install-medley` automatically updates this information for you.) The `medley` script assumes that you have not changed the installation subdirectory structure from when it was originally installed.

The script will first try to find a key in the file *Installation directory/.medleyKey.hostname* or in *~/medleyKey.hostname*. *Installation directory* is where Medley was originally installed on your system. *hostname* is the name of the host for which the license key was issued. If neither file is found, you will be prompted for a key.

`medley [[emulator] sysout]` [Command]

---

*emulator* Given a pathname or a simple name, the command will search for *emulator* as follows:

- If *emulator* is a relative or absolute pathname, e.g., `/share/medley/emulators/lde`, it will only try that pathname.
- If *emulator* is a plain file name, e.g., `lde`, the script uses the regular UNIX search path to find it. If it cannot find it, the script looks in the installation directory for Medley at your site, e.g., `/usr/share/lde/install.sunos4.1/lde`.
- If you omit *emulator*, the script uses `lde` as the default value, searching for it in the same fashion as above.

*sysout* The command will search for *sysout* as follows:

- If *sysout* is a relative or absolute pathname, e.g., `../applications/my.sysout`, it will only try that pathname.
- If *emulator* is a plain file name, e.g., `my.sysout`, it will look for it in the following order:
  1. Current working directory `my.sysout`
  2. Installation directory for Medley at your site, e.g., `/usr/share/lde/lispsysouts/my.sysout`.
  3. Your home directory, `~/my.sysout`
  4. The `medley` subdirectory in your home directory, `~/medley/my.sysout`.
- If you omit *sysout*, the script looks for it as explained in the Where Medley Looks for Your Sysout section below.

## Examples

- `prompt% medley`

```
To start Medley 2.0, a host access key is required.
Call Venue at (1-800-228-5325) for one,
and be prepared to give them your workstations host ID#
Your workstations host ID# is: 51006da3
Type in key or [^C] to abort:8bf7723e 459aab34 73491feb
Saving key '8bf7723e 459aab34 73491feb' into file
'.medleyKey.hostname' ...
Trying /usr/share/lde/.medleyKey.hostname ... Write
protected !
```

```
Trying home-directory/.medleyKey.hostname ... Done
Starting up Medley 2.0 ...
..... Medley 2.0 starts .....
```

If you had Medley installed in `/share/medley` on your system, it would try to run the emulator `/share/medley/install.sunos4.1/lde`, using the `sysout` `/share/medley/lispsysouts/LISP.SYSOUT`.

In this example you are prompted for a key, which is saved into the file: `home-directory/.medleyKey.hostname`

The script tried to save the key into the installation directory but did not have write access there. Instead it was put into your home directory (`~`) `.hostname` is the name of the host running medley.

The next time you use the script `medley`, you will not be prompted for the key.

- `prompt% medley application.sysout`

If you had `application.sysout` in your home directory, it would try running the emulator `/share/medley/install.sunos4.1/lde` using `~/application.sysout`.

## Where Medley Looks for Your Sysout

If you run Medley directly, the system searches the following places, in order, for the `sysout` to be used:

- *command line*

The name of the `sysout` file can be given on the command line when starting Medley; for example,

```
prompt% lde sysout -k 'thishost-key'
```

- `LDESRCESYSOUT`

If no `sysout` file name is given on the command line, the value of the environment variable `LDESRCESYSOUT` is used as the name of the `sysout` file. For example:

```
prompt% setenv LDESRCESYSOUT my.sysout
prompt% lde -k 'thishost-key'
```

would run the host key `my.sysout`.

- `~/lisp.virtualmem`

Finally, Medley looks for the file `lisp.virtualmem` on your home directory.

## Where Medley Looks for Your Site Initialization File

When Medley starts, it reads in a Lisp site initialization file. This site initialization file sets things like pathnames for fonts, site parameters, and the like.

Greeting and initialization are described in the *Interlisp-D Reference Manual*, Section 12.1.

Medley looks for a site initialization file in a number of locations:

- LDEINIT

If the environment variable `LDEINIT` is set to a complete Lisp file name, Lisp looks there first for the site initialization file:

```
prompt% setenv LDEINIT /usr/lisp/my-site-init.lisp
```

- `/usr/share/lde/site-init.lisp`

If `LDEINIT` is not set or there is no file with the name given, Lisp looks for a site initialization file called `/usr/share/lde/site-init.lisp`. The distribution tape contains a sample site initialization file in the Lisp library directory `/usr/share/lde/lisplibrary/site-init`. The system administrator should copy `site-init` into `/usr/share/lde/site-init.lisp` then customize it for the site. The comments in the sample `site-init.lisp` describe the parameters it sets and give guidelines for customizing it to your local conditions.

- `{DSK}INIT.DFASL`, `{DSK}INIT.LCOM`, `{DSK}INIT.LISP`

Finally, Lisp looks for a site initialization file on your Medley home directory (`{DSK}`). Chapter 5, *Medley File Systems*, describes the `{DSK}` device.

## Medley and X Windows

---

Medley 2.0-S supports the X Window System, Version 11 Release 4 (X11R4). Medley runs in a single X window; Medley's "screen" is displayed in that window, and you use Medley as usual.

### Starting X Windows

Start the X server on your console. Use the `xinit` command.

If necessary, start a window manager as a client of X (`xinit` often starts a window manager). The window manager provides many window management functions, such as moving, resizing and iconifying the window. Medley has no window management function of its own.

### Running Medley Remotely

You can run Medley on one machine, with the window on some other machine. To do so, perform these steps on the machine whose keyboard and display you will be using:

1. Add the host name to execute the Medley access control list:

```
xhost + hostname
```

2. Open a new `xterm` and `rlogin` to the Sun Workstation on which Medley is to run. Set the environment variable `DISPLAY` to the host name of the server machine:

```
setenv DISPLAY servername:0
```

3. Set the `LDEKBDTYPE` environment variable to tell Medley what kind of keyboard you will be using. Possible values are:

```
type3    Sun Type 3 keyboard  
type4    Sun Type 4 keyboard  
rs6000   IBM RS/6000 or PS/2  
dec3100  DECstation 3100 or 5000
```



```
hp9000  HP9000 Series 700 or 800
x       Generic X terminal
```

If you don't set `LDEKBDTYPE`, it will default to `x`. The advantage of specifying a specific keyboard lies in how Medley treats the special function keys. The specific keyboard maps maximize the usefulness of keys marked, e.g., "Find". The generic keyboard code cannot do that reliably.

#### 4. Start up Medley.

A new window for Medley will appear on the X server's screen.

### The Medley Window

Normally, Medley uses the whole screen. Under X, Medley's "screen" appears in a single X window. Medley's screen is slightly smaller than the screen you are using to display it; if you make the X window full-screen-size, you see Medley's entire screen. If it is smaller, you will need to scroll to see parts of the screen.

The scroll bars (at the right and bottom of the X window) control what parts of Medley's screen appears in the window. Use the vertical scrollbar to scroll up and down, and the horizontal scrollbar to scroll left and right. The gravity buttons (at the lower right corner) set the bitgravity of the display window. Click the mouse button on one of these areas. The shade pattern is moved to the clicked area, and the bitgravity is set in the corresponding corner on the display window. The bitgravity determines how reshaping the X window affects what part of the Medley screen is visible.

## Environment Variables

Medley on the Sun uses several environment variables. They can be set from the shell with the `setenv` UNIX command. By convention, environment variable names use uppercase rather than lowercase letters, e.g., `LDEDESTSYSOUT`. The Medley environment variables are listed below, with a reference to sections in this *Guide* where further information can be found.

<code>LDEKBDTYPE</code>	See the Medley and X Windows or Sun Type 4 Keyboard sections in this chapter.
<code>LDEINIT</code>	See the Site Initialization File section in Chapter 4.
<code>LDESRCESYSOUT</code>	See the Where Medley Looks for Your Sysout section in this chapter.
<code>LDEDESTSYSOUT</code>	See the Saving Your State section in Chapter 4.
<code>LDESHELL</code>	See the UNIXCHAT section of the <i>Lisp Library Modules</i> .
<code>LDEFILETIMEOUT</code>	See the File System Errors subsection in Chapter 5.

## Keyboard Interpretation

This section describes how Medley interprets the Sun Type 3 and Type 4 keyboards. Except when running under X, Medley performs its own keyboard interpretation, taking raw up/down transitions directly from the keyboard. Medley uses its own key

numbering scheme; key numbers are used by Lisp functions such as `IL:KEYDOWNP` and `IL:KEYACTION`.

These key assignments were chosen to maximize compatibility with both the Xerox workstation keyboard and the normal Sun keyboards. You can attach a Sun Type 3 or Type 4 keyboard template, which also shows the Medley keyboard assignments, to your Sun Type 3 or Type 4 keyboard. Both templates are included with your Medley documentation set.

### Sun Type 3 Keyboard

Figure 3-1 shows the key number assignments for the Sun Type 3 keyboard. Figures 3 - 2 through 3 - 4 show Medley's key assignments for the Sun Type 3 keypads.

61	91	97	99	100	67	68	101	66	104	80	13	73	74	75					
92	63	33	32	17	16	1	0	2	4	53	22	8	10	59	105	45	81	82	83
14	62	34	19	18	3	48	49	51	6	23	25	11	58	29	15	84	85	87	
111	89	36	21	20	5	35	50	52	38	9	26	43	28	44	94	69	70		
90	46	41	40	24	37	7	39	54	55	27	42	12	60	71	98	76	72		
		56	31	57									93	47					

*Figure 3-1. Sun Type 3 Key Numbering*

Stop	Again
Help	Undo
Same	Move
Open	Copy
Find	Delete

*Figure 3-2. Sun Type 3 Left Key Pad*

NumLock	ScrolLock	Break
7 Home	8 ↑	9 PgUP
4 ←	5	6 →
1 End	2 ↓	3 PgDN
Ins	DOIT	CapsLock

*Figure 3-3. Sun Type 3 Right Key Pad*

Center	Bold	Italic			Case		Strikeout		Underline		Super Sub	Large Smaller	Margin	Back Word
Esc	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	( 9	) 0	- =	+ =	 \ _	~ ;
Tab	Q q	W w	E e	R r	T t	Y y	U u	I i	O o	P p	{ [	} ]	Backspace	
Ctrl	A a	S s	D d	F f	G g	H h	J j	K k	L l	: ;	" ,	Return		
Shift	Z z	X x	C c	V v	B b	N n	M m	< ,	> .	? /	Shift	LF		
Caps	Meta	Space										Expand	Next	

*Figure 3-4. Sun Type 3 Center Key Pad Interpretation*

### Sun Type 4 Keyboard

Figure 3-5 illustrates the keyboard interpretation for the Sun Type 4 keyboard. Figures 3-6 through 3-8 show the keyboard and the left and right key pads for the Sun Type 4 keyboard.

NOTES: In SunOS 4.0, the NEXT (ALT/GRAPH) key on the Type 4 keyboard is inaccessible. Later versions of SunOS fix this.

Medley cannot detect whether it is running on a workstation with a Type 4 keyboard when running SunOS 4.0, 4.0.1, or 4.1. To make it work correctly on your workstation, enter the following before you start running Medley:

```
setenv LDEKBDTYPE type4
```

61	91	97	99	100	67	68	101	66	104	80	106	107	108	105	13	75	110	74	73		
109	63	33	32	17	16	1	0	2	4	53	22	8	10	59	15	64	65	95	96		
14	89	34	19	18	3	48	49	51	6	23	25	11	58	29	0	81	82	83	102		
111	62	36	21	20	5	35	50	52	38	9	26	43	28	45	44	84	85	87			
90	46	41	40	24	37	7	39	54	55	27	42	12	60	71		94	69	70	76		
92		56	31	86	57							88	93	47	98	13					

*Figure 3-5. Sun Type 4 Key Numbering*

Stop	Again
Props	Undo
Same	Copy
Open	Move
Find	Delete
Help	

**Figure 3-6. Sun Type 4 Left Key Pad**

Break	PrSc	scroll lock	num lock
=	/	*	-
7 Home	8 ↑	9 PgUP	+
4 ←	5	6 →	
1 End	2 ↓	3 PgDN	DOIT
Ins		Del	

**Figure 3-7. Sun Type 4 Right Key Pad**

F1 Center	F2 Bold	F3 Italic	F4 Case	F5 Strike	F6 Under	F7 Super	F8 Large	F9 Margin	F10	F11	F12	\	Delete Word	
Esc	! 1	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* 8	( 9	) 0	-	+ =	Back Space	
Tab	Q q	W w	E e	R r	T t	Y y	U u	I i	O o	P p	{ [	} ]	Return	
Ctrl	A a	S s	D d	F f	G g	H h	J j	K k	L l	: ;	" ' ~			
Shift		Z z	X x	C c	V v	B b	N n	M m	< ,	> .	? /	Shift	LF	
Caps	Meta	Left Spc	Space									Right Spc	Expand	Next

**Figure 3-8. Sun Type 4 Center Key Pad Interpretation**

[This page intentionally left blank]

## 4. USING MEDLEY ON THE SUN WORKSTATION

---

Once the system administrator has installed Medley software on the Sun, Lisp users can customize their Medley Lisp environments. This chapter provides basic information to get you started in the Medley environment on a Sun Workstation.

### Setting Up a Site Init File

---

The users at a given site generally print to the same printers, load library files from the same directory, and so on. Medley uses variables to supply defaults for such things. The obvious place to set these variables is in one common initialization file. That is the Site Init File's role.

The Site Init File is a file of Lisp expressions that is loaded when you start Medley with a fresh `LISP.SYSOUT`.

The following Lisp symbols should be set in your site init file:

**`IL:USERGREETFILES`** **[Variable]**

---

A list of templates to search for the place where individuals should find their personal init files. If this is not set in the site init file, no personal init file is used. The list should be similar to the following:

```
(( {file-server} < USER >LISP>INIT.LCOM)
 {file-server} < USER >LISP>INIT)
 {file-server} < USER >INIT.LISP))
```

**`IL:DISPLAYFONTDIRECTORIES`** **[Variable]**

---

A list of directories to search when the system is looking for display fonts. The site initialization file should set it to a list of strings, each containing a complete pathname for font files, e.g., ("`{UNIX}/usr/local/lde/fonts/display/presentation/`").

**`IL:INTERPRESSFONTDIRECTORIES`** **[Variable]**

---

A list of directories to search when the system is looking for Interpress font widths.

**`IL:DIRECTORIES`** **[Variable]**

---

The list of paths to search for files that are not found in the current (Lisp) connected directory.

**`IL:LISPUSERSDIRECTORIES`** **[Variable]**

---

The list of paths to search for library and LispUsers' files. Remember that every path in this list should also be in `DIRECTORIES`.

**`IL:DEFAULTPRINTINGHOST`** **[Variable]**

---

A list of names of default printers.

**`IL:DEFAULTPRINTERTYPE`** **[Variable]**

---

The default printer type, e.g., `POSTSCRIPT`.

**XCL:\*LONG-SITE-NAME\*** **[Variable]**

The value of the function XCL:LONG-SITE-NAME, e.g., "Frobnitz, Baz and Lispers, Incorporated."

**XCL:\*SHORT-SITE-NAME\*** **[Variable]**

The value of the Common Lisp function XCL:SHORT-SITE-NAME, e.g., "Frobco".

**IL:\BeginDST** **[Variable]**

The day of the year on or before which Daylight Savings Time takes effect (i.e., the Sunday on or immediately preceding this day). Must be set to 98 in the USA if Lisp is to perform time computations correctly (subject, of course, to future legislation). If you are in a region where Daylight Savings Time is not observed, set the value to 367.

**IL:\EndDST** **[Variable]**

The day of the year on or before which Daylight Savings Time ends. Must be set to 305 in the USA.

---

## Setting Up a Personal Init File

---

Your personal init file keeps track of the location of your home directory and windows layout; it also remembers which library files you always load.

Your personal init file is a file of Lisp expressions that is loaded and run after the site init file. You can create it either as a text file, or have Medley's File Manager help you.

Your initialization file is normally ~/INIT.LCOM

---

## Saving Your State

---

On the Sun, lde is an ordinary UNIX program that allocates a 45 MB data area, reads into that area several megabytes of data (the sysout), and modifies it there. Under UNIX, that program's data requirements (which include the sysout) are handled by UNIX; all Medley does is modify in "memory" a copy of your original sysout file. UNIX, transparently to Medley, handles all real memory swapping. This has several consequences related to starting, saving, and restarting sysouts.

On Xerox workstations, the virtual memory partition is updated periodically and used to store new pages as they are allocated or flushed from the real memory of the machine. For example, LOGOUT and SAVEVM write out only those pages of data which are different from what might already be in the virtual memory file.

On the Sun Workstation, however, the contents of virtual memory are only written to a file by an explicit call to SAVEVM, LOGOUT, SYSOUT, or MAKESYS. This file is an ordinary SunOS file (normally ~/lisp.virtualmem). The entire virtual memory, which may be many megabytes of data, is written out there.

On the Sun Workstation, starting anew from a saved virtual memory file requires reading it into memory. On the Xerox workstation, it is necessary to first copy the saved sysout to the virtual memory file and then read it in. Thus, restarting a saved sysout or virtual memory file is significantly faster on a Sun Workstation.

The file that LOGOUT and SAVEVM writes is normally `~/lisp.virtualmem` (i.e., the file `lisp.virtualmem` on the user's home directory). However, the environment variable `LDEDESTSYSOUT` can be used to override this default. For example, you might want to keep virtual memory images on `/user/local`. During a demonstration where you do not want the memory image saved, you can reset `LDEDESTSYSOUT` to `/dev/null`. You can use the C-Shell command `setenv` to do this, e.g.:

```
prompt% setenv LDEDESTSYSOUT "/dev/null"
```

Cursor tracking interferes with writing out the screen bitmap as part of the Medley memory image. For this reason, Medley takes the cursor down before saving a virtual memory image as part of LOGOUT, SAVEVM, SYSOUT, or MAKESYS. When this happens, the message

```
Saving VMem, taking mouse down
```

appears in the prompt window, and cursor tracking is disabled.

Because the virtual memory file need not already exist to run Medley, the functions LOGOUT and SAVEVM can signal the following file errors:

```
File-System-Resources-Exceeded
Protection-Violation
File-Wont-Open
```

Even if some errors occur while saving a virtual memory, the old destination file is safe. Saving does not overwrite the old virtual memory file. The saving virtual memory file is named with `-temp`, such as `lisp.virtualmem-temp`. The file is renamed to a specified name, such as `lisp.virtualmem`, at the last sequence of the save.

When the user does not have enough space to save the virtual memory, the old virtual memory file can be overwritten by setting `IL:\LDEDESTOVERWRITE` to `T`. The initial value of `IL:\LDEDESTOVERWRITE` is `NIL`. In some cases, even if the user tries to overwrite, there may still not be enough space.

In Medley, a "page" is 512 bytes. Under SunOS, the page size is variable; some Sun Workstations use 8 Kbyte pages. In general, Medley functions deal only in units of Medley pages, e.g., the `SIZE` attribute of files is in terms of 512-byte pages, (`VMEMSIZE`) returns the number of 512-byte pages in use.

#### **(IL:LOGOUT *FAST*)**

**[Function]**

Lets you exit Medley cleanly. The parameter *FAST* indicates whether resumption of the same environment is desirable and in what fashion. Before exiting, disk buffers are written, and network connections subject to timeout are closed.

If *FAST* is `NIL`, LOGOUT first saves your virtual memory in a file. Change the file name by setting the UNIX environment variable `LDEDESTSYSOUT`. If this variable is not set, the file saved is `~/lisp.virtualmem` (i.e., `lisp.virtualmem` on the user's home directory).

If *FAST* is `T`, Medley stops without writing the virtual memory file. It is not possible to resume execution in the same image.

#### **(IL:SAVEVM )**

**[Function]**

Saves your state, but does not exit. It causes the current virtual memory image to be written to the location specified by the environment variable `LDEDESTSYSOUT`, if this variable is set; otherwise it is written to `~/lisp.virtualmem`. This allows Lisp to



continue. Execution in Medley continues after memory is saved; thus, `SAVEVM` operates as a sort of checkpoint of the current working state. `SAVEVM` can cause the following error:

`File-System-Resources-Exceeded.`

---

**(IL:SYSOUT FILE) [Function]**

Performs the equivalent of `SAVEVM` and then copies the saved image to `FILE` for devices other than `{DSK}` and `{UNIX}` (e.g., XNS file servers). (See Chapter 5, Medley File Systems, for further information on `{DSK}` and `{UNIX}`.) `SYSOUT` can cause the following error:

`File-System-Resources-Exceeded.`

---

## Sun-Specific Environment Functions

---

### System Environment Functions and Variables

These functions, which interrogate the system environment, operate as described below when they are invoked on the Sun Workstation:

---

**(IL:REALMEMORYSIZE) [Function]**

On some machines, returns the total amount of real memory available; does not work on a Sun Workstation (i.e., returns a meaningless value).

---

**(CL:MACHINE-TYPE) [Function]**

Returns a string identifying the type of computer hardware the system is running under. On the Sun-3 workstation `MACHINE-TYPE` returns "mc68020". On a Sun-4 workstation, `MACHINE-TYPE` returns the string "sparc".

---

**(IL:MACHINETYPE) [Function]**

Identifies the generic type of Lisp machine in use. On the Sun Workstation, it returns the symbol `IL:MAIKO`.

---

**(CL:MACHINE-VERSION) [Function]**

Returns a string identifying the version of the emulator running; e.g., "Microcode version: 279, memory size: 16384".

---

**(CL:MACHINE-INSTANCE) [Function]**

Returns a string containing the workstation host ID (in hexadecimal) and the host name.

---

**IL:LISP-RELEASE-VERSION [Variable]**

Identifies the release number within a single major release name. In Medley 2.0, `IL:LISP-RELEASE-VERSION` is 2.0 While `IL:MAKESYSNAME` does not change, `IL:LISP-RELEASE-VERSION` always changes with each new sysout release. This variable did not exist in the Medley 1.0-S sysout.

---

**IL:\MY.NSADDRESS****[Variable]**

Fills in the fields of the network address with the host ID if Medley is run without the Ethernet enabled. Programs that use the network address as a unique identifier should be aware that the value could vary from session to session depending on whether or not the Ethernet is enabled. (Refer to Chapter 14 of the *Interlisp-D Reference Manual* for further information.)

## VM Functions

The biggest difference is a change in terminology. On Xerox 1100 series workstations, Lisp itself handles all virtual memory operations directly, so the terms "sysout" and "virtual memory image" can be used interchangeably. The running sysout resides in a reserved area on the workstation local disk (the virtual memory partition) that Lisp reads from and writes to as it needs to move pages into and out of physical memory.

**(IL:VMEMSIZE)****[Function]**

Returns the number of 512-byte pages of the Medley virtual memory that are in use. This number is a good estimate of the size of a `SYSOUT`, `MAKESYS`, or `SAVEVM` virtual memory file.

**(IL:VMEM.PURE.STATE ON/OFF)****[Variable]**

Has no effect on the Sun Workstation. The virtual memory file is not modified except by an explicit `(LOGOUT)` or `(SAVEVM)`.

**IL:BACKGROUNDPAGEFREQ****[Variable]**

Has no effect on the Sun Workstation. The virtual memory file is not modified except by an explicit `(LOGOUT)` or `(SAVEVM)`.

You can control how much virtual memory Medley uses by using the `-m` switch, as described below.

```
ldeether [<SYSOUT-name>] [-m<memory-size>] [other options]
```

**[Command]**

Allows you to specify an arbitrary virtual memory size for Medley.

`-m`                    Specifies the memory size

*memory-size*        8 through 32 Mbytes

When you use `-m`, the value of `IL:\STORAGEFULLSTATE` in the sysout you start should not be 3 or 4. Those values mean it already used more than the 8-Mbyte space in the sysout. Because of the Medley storage management architecture, the virtual memory size cannot be changed after `IL:\STORAGEFULLSTATE` has been set to 3 or 4. This value can be examined just before `(IL:LOGOUT)` if you want to specify the virtual memory size during the next start-up.

Example: `ldeether /usr/LISP.SYSOUT -m 16`

This example means 16 Mbytes of virtual space will be assigned for Lisp.

## Stopping Lisp Temporarily

**(IL:SUSPEND-LISP)**

**[Function]**

Suspends, temporarily, the UNIX process running Medley. Using the `fg` C-Shell command, the Medley process can be continued from the C-Shell where it was started. `SUSPEND-LISP` has no effect on Xerox Lisp workstations. This function should not be used during I/O operations (file or network).

## Login Functions

This section describes the interaction between the usernames and passwords in Medley and the SunOS usernames and passwords. The functions `IL:USERNAME`, `IL:SETUSERNAME`, `IL:SETPASSWORD`, and `IL:LOGIN` access the username/password database used by Medley in network operations. (For further information, see Chapter 24 of the *Interlisp-D Reference Manual*.) When Medley is started, this database contains only the SunOS username, with no password. Except for this, there is no interrelation between these Medley functions and SunOS usernames and passwords.

`IL:USERNAME` returns the SunOS login name under which the emulator was started. A subsequent `IL:SETUSERNAME` or `IL:LOGIN` changes `IL:USERNAME`, and the default login name for network access to XNS and PUP hosts. However, it does not change the SunOS login name or access capabilities for files on `{DSK}` or `{UNIX}`. (See Chapter 5, Medley File Systems, for detailed information on `{DSK}` and `{UNIX}`.) Because it doesn't change the SunOS login name, it won't change the author name on SunOS files created from Lisp.

The following functions apply to login activities.

**`(IL:UNIX-USERNAME)`** **[Function]**

Returns a string consisting of the username of the SunOS process running Medley. Returns `NIL` if one of the following conditions apply:

- You are not running under UNIX
- You do not have a full name entered in `/etc/passwd` or the NIS password map
- An error occurs.

**`(IL:UNIX-FULLNAME)`** **[Function]**

Returns a string containing the full name of the owner of the SunOS process running Medley. Returns `NIL` if the user is not running under UNIX or an error occurs.

**`(IL:LOGIN HOST FLG DIRECTORY MSG)`** **[Function]**

Attempts to maintain user IDs and passwords for network as well as local access. If `HOST` is `NIL`, this function attempts to perform the SunOS `setuid` operation.

Unless you are running as root, this will not change your SunOS login.

## Environment Inquiry

The following functions return the values of UNIX environment variables or machine parameters. They return `NIL` if run in Medley on Xerox 1100 series workstations.

**`(IL:UNIX-GETENV STRING)`** **[Function]**

Returns the value of the environment variable with the given name. The argument `STRING` should be the name of a UNIX environment variable. For example, `(UNIX-GETENV "HOME")` might return the user's home directory.

**`(IL:UNIX-GETPARG STRING)`** **[Function]**

Returns the value of one of a few built-in parameters. The argument `STRING` should be the name of one of the following UNIX environment variables:

<b>Variable</b>	<b>If running on this hardware</b>	<b>Returns</b>
"MACH"	Sun-4	"sparc"
	Sun-3	"mc68000"
	RS/6000	"rs/6000"
	HP9000	"hp9000"
	DEC3100	"mips"
	PS/2	"i386"
"ARCH"	Sun-4	"sun4"
	Sun-3	"sun3"
	RS/6000	"rs/6000"
	HP9000	"hp9000"
	DEC3100	"dec3100"
	PS/2	"ps/2"
"HOSTNAME"	All	Returns the local host name
"HOSTID"	All	Returns the local host identification number as a hexadecimal string

## Display and Keyboard Functions and Variables

Some Medley display and keyboard functions and variables operate differently on the Sun Workstation.

The following functions have no effect on a Sun Workstation, and always return NIL:

IL:CHANGEBACKGROUNDBORDER

IL:VIDEORATE

IL:SETMAINTPANEL

IL:VIDEOCOLOR

The functions IL:BEEPON, IL:BEEPOFF, IL:PLAYTUNE, IL:RINGBELLS generate monotones.

**(IL:BEEPON *FREQ*)** **[Function]**

---

Turns on the keyboard tone generator on the Sun Workstation. The *FREQ* argument is ignored.

**(IL:BEEPOFF)** **[Function]**

---

Turns off the keyboard tone generator.

**(IL:PLAYTUNE *TUNEPAIRS*)** **[Function]**

---

Sounds tones, but ignores the frequencies of the values in *TUNEPAIRS*.

**(IL:RINGBELLS)** **[Function]**

---

Causes the machine to beep several times.

## Timers and Clocks

UNIX is a timesharing operating system. When Medley is running, other programs can be running at the same time on the same workstation.

On a Xerox workstation running Lisp, CPU time could be computed exactly from elapsed time after subtracting known system overhead. To allow older Interlisp-D programs to work unchanged, the timer functions were modified to allow programs that accounted for time on Xerox workstations to continue to run. Time is categorized as follows:

CPU time:	The total amount of time spent executing Medley's process in user mode.
SWAP time:	The total time spent running other processes (Elapsed time – (CPU time + Disk time)).
Disk I/O time:	The total amount of time spent in the system executing on the behalf of Medley's process.

The Medley functions `CLOCK`, `TIME`, and the like get the time of day directly from SunOS. The function `SETTIME` has no effect on the Sun Workstation.

`IL:\RCLKMILLISECOND`

**[Variable]**

The number of clock "ticks" in a millisecond. On the Sun Workstation, this value is always 1000. All of the timer functions that deal in clock ticks will do their computation in microseconds. Note, however, that the Sun Workstation does not have that accurate a clock resolution. While clock resolution varies from one operating system version to another, it often has a resolution no better than 1/60th of a second.

## Miscellaneous Operational Differences

The stack and virtual memory handling functions on the Sun Workstation are implemented differently from the way they are on the Xerox workstations. For this reason, the "cursor bars" used on the Xerox workstations are not used on the Sun Workstation.

When working in Medley on a Sun workstation, you should periodically load a fresh sysout. Older Medley sysouts don't run as well as "fresh" sysouts due to a number of factors such as fragmentation of memory, increased working set, more objects taking up various spaces (e.g., gc tables), reduced symbol space.

On Xerox workstations, users are reminded to reload fresh sysouts, because they eventually fill up their sysout partition. With Sun workstations, there is no such limit reminder, so users' sysouts tend to grow to the maximum size (32 MB), and thus run slower and slower.

## Console Messages

Under SunOS, various system processes and operations attempt to log information on the console. Since Medley takes over the screen, console messages are redirected (except when running under X); a background process in Medley causes them to appear in the prompt window.

However, when Medley is run remotely (i.e., not from the console), most console, or operating system, messages are printed in the prompt window. However, some messages may also appear in the middle of the Medley display screen or on the remote tty. This occurs because UNIX is often confused about where to send messages. Note that Medley is normally run remotely only for debugging purposes.

---

**CAUTION**

Critical UNIX system processes can hang if the buffer holding console messages fills. Medley uses a temporary file, `/tmp/XXXX-lisp.log`, where `XXXX` is the user's login name, to buffer console messages before printing them. Do not delete this log file while Medley is running. If the log file is deleted, console messages can no longer be printed in the Medley prompt window.

---

[This page intentionally left blank]



## 5. MEDLEY FILE SYSTEMS

---

This chapter discusses the conventions for using files from Medley.

### File Naming Conventions

---

In Lisp, a file name (pathname) consists of a collection of fields: the *host*, *directory*, *name*, *extension* and *version*. These fields are optional. The standard Lisp syntax for these fields is:

*{host}<directory>name.extension;version*

The *directory* field can be a directory path consisting of a sequence of directory and subdirectory components. Slashes (/) and right angle brackets (>) can be used to delimit a directory name; there is no distinction made between them. Square brackets ([]) are not acceptable as directory delimiters.

Duplicated directory delimiters are treated as a single delimiter. Thus, the following two file names specify the same file:

```
{DSK}<LISP>USERS>FOO.;1
{DSK}</LISP/USERS/>FOO.;1
```

### Hosts that Medley Supports

---

{CORE}	Creates "files" in memory; useful for quick temporary files
{LPT}	Creates files that are automatically sent to your printer
{NULL}	Creates a file that does nothing
{DSK} and {UNIX}	Give you access to the Sun's file systems; the rest of the chapter concentrates on them.

The above hosts are described in more detail in the IRM.

### Using SunOS Files from Medley

---

You can access any mounted SunOS file system directly from Lisp. The mounted file system is available as an I/O device of the Lisp environment. This file system appears as the local disk of Lisp, even though it may be a remotely mounted file system of networked Sun file servers.

Many of the file devices to which the Medley environment can talk, including PUP, XNS file servers, the {CORE} device, and others, have facilities that are not directly supported by SunOS. For example, many file systems have file version numbers and case insensitive file search conventions.

Medley on the Sun Workstation has two distinct "host" names that can be used to access the SunOS file system. These host names are provided for compatibility with existing applications and tools. They also simultaneously allow natural interaction with the SunOS file system. The names are:

- `{DSK}` On the Xerox workstation, `{DSK}` gave you access to your local hard disk; to use `{DSK}`, you had to create a directory on each disk partition you wanted to use. On the Sun Workstation, in contrast, the `{DSK}` device lets you access the file system using similar conventions to those used for `{DSK}` on the Xerox workstation local disk devices. In particular, `{DSK}` files have version numbers; `{DSK}` file name recognition also ignores the case of letters.
- `{UNIX}` The `{UNIX}` device lets you use the mounted file systems with the normal naming conventions of the SunOS file system. `{UNIX}` files do not have version numbers, and the file name recognition treats lowercase letters as distinct from their uppercase equivalents.

File streams can be opened or closed on both devices. The reason for having both devices is to more easily support the running of applications that were originally developed on a Xerox workstation, while still allowing new applications to interact more naturally with UNIX.

NOTE: Both `{DSK}` and `{UNIX}` work as filters. They act as pointers to a device. On 11xs, `{DSK}foo` is the same as `{DSK}<lispfiles>foo`. On the Sun, `{DSK}foo` is the same as `$HOME/foo ~user/foo`.

### Common `{DSK}` and `{UNIX}` Naming Conventions

- To include a special character (e.g., `>` or `;`) in a file name, precede it with a single quote (`'`). To include a single quote in a file name, precede it with another single quote. You can quote any of these characters: `<`, `>`, `,`, `~`, and a period (`.`). The following examples show how the single quote notation on `{DSK}` and `{UNIX}` is used.

#### **`{DSK}` Name From Lisp**

```
foo'>bar.baz;l  
foo';bar.baz;l  
foo''bar.baz;l
```

#### **File Name From SunOS**

```
foo>bar.baz  
foo;bar.baz  
foo'bar.baz
```

- `{DSK}` and `{UNIX}` do not allow you to use either the slash (`/`) or the NUL character in file names. Thus, you cannot name files containing these characters.
- Both `{DSK}` and `{UNIX}` can handle the following characters, which were defined as special characters in Medley Release 1.1: backslash (`\`) and tilde (`~`).
- `{DSK}` and `{UNIX}` can distinguish between a file name with a period at the end (e.g., `foo.`) and a simple file name (e.g., `foo`). The final period is preceded with a single quote, as shown in the following example:

#### **`{DSK}` Name From Lisp**

```
foo.;l  
foo'..;l
```

#### **File Name From SunOS**

```
foo  
foo
```

- On `{DSK}` and `{UNIX}`, the C-Shell and SunOS directory notations (`~`, `..`, and `...`) are supported in the Lisp directory specification. The tilde character (`~`) is allowed at the very beginning of the directory specification of a pathname. A combination of relative path specifiers (`~`, `..`, `...`) is supported. The tilde character corresponds to the

user's home directory at login. The period (.) corresponds to the current working directory. Two periods (..) indicates the parent of the current working directory.

- File names are returned by the system (e.g., `INFILEP`) in more canonical form. The function which returns the full file name returns it in the canonical form, as in `{DSK}<usr>etc` rather than `{DSK}/usr/etc/`. This change will make some tools which depend on the conventional file name representation described in the *Interlisp-D Reference Manual* work correctly on the Medley file system (e.g., `COPYFILES`).

## {DSK} Naming Conventions

The {DSK} device performs the following file name transformation when actually accessing the SunOS file system:

- Mixed case letters are read as such.
- File name searches are done case-sensitive first; if a match is not found, the system does a case-insensitive search.
- The left angle bracket character (<) is translated to a slash (/), the delimiter for the root directory.
- {DSK} supports relative pathnames. You can specify relative pathnames by omitting a slash (/) or left angle bracket (<) as the first character in the directory field. For example:

`{DSK}foo.fee` and `{DSK}~/foo.fee` are relative to the user's UNIX home directory (`~/foo.fee`).

`{DSK}./foo.fee` is relative to the user's current working directory (`SunOS./foo.fee`).

`{DSK}../foo.fee` is relative to the parent directory of the user's current UNIX working directory (`../foo.fee`).

The Medley 2.0-S {DSK} device supports the notation in which the three meta characters (' , .. , and ~) are used together, as shown in the following example:

```
{DSK}~/../tom/foo.c
```

In this example, the {DSK} device interprets `tom` as one of the subdirectories of the parent directory of the user's home directory.

{DSK} also supports the tilde-name (`~name`) convention. {DSK} interprets `{DSK}~tom/foo.c` as a file named `foo` on `tom`'s home directory. In this notation, the user name is case-sensitive (e.g., `~tom` and `~Tom` are treated as different users).

## Version Numbering

The UNIX file system does not support version numbers in file names; {DSK} emulates versions with a naming convention. (GNU Emacs also uses this convention.) This section explains how {DSK} version numbers are represented in the SunOS file system.

- When you create a completely new file, it appears in the SunOS file system without a version number.

**{DSK} Name From Lisp**

**File Name From SunOS**

`bar.baz;1``bar.baz`

- When you create (from Medley) a file with a version other than 1, Medley adds version numbers to that file name, as a trailing number between tildes, e.g., “myfile.~12~” for the twelfth version of myfile.

The following shows some examples of equivalent file names in Lisp and SunOS.

**{DSK} Name From Lisp**

```
bar.baz;1
bar.baz;2
bar.;23
```

**File Name From SunOS**

```
bar.baz.~1~
bar.baz.~2~
bar.~23~
```

- Medley always maintains a versionless file which is hard-linked to the highest extant version of the file (i.e., they are two names for the very same file). This file name does not appear in the {DSK} directory listing.

**From {DSK}**

```
foo.c;15
foo.c;23
```

**From SunOS**

```
foo.c (hard linked with foo.c.~23~)
foo.c.~15~
foo.c.~23~
```

Similarly, a file created in UNIX with no version number is treated by {DSK} as the highest version.

- When you create a new version of a file, the versionless-file link is broken, and the versionless file is hard-linked to the new highest version.

**From {DSK}**

```
foo.c;15
foo.c;22
foo.c;24 (new file)
```

**From SunOS**

```
foo.c (hard linked with foo.c.~24~)
foo.c.~15~
foo.c.~22~ (no link with foo.c)
foo.c.~24~ (new file, link from foo.c)
```

- When you delete the highest version of a file, the versionless file is also deleted. If any older versions of the file remain, a new link is created from the versionless name to the highest version extant. For example, if you have the files

**From {DSK}**

```
foo.c;1
foo.c;2
```

**From SunOS**

```
foo.c (linked to foo.c.~2~)
foo.c.~1~
foo.c.~2~
```

and you delete `foo.c;2` from {DSK}, the resulting files are:

**From {DSK}**

```
foo.c;1
```

**From SunOS**

```
foo.c (linked to foo.c.~1~)
foo.c.~1~
```

- When you rename a file, it works the same as deleting the file under the old name then creating it under the new name. For example, if you have the following {DSK} files

**From {DSK}**

```
foo.c;1
foo.c;2
fee.c;1
fee.c;2
```

**From SunOS**

```
foo.c (linked to foo.c.~2~)
foo.c.~1~
foo.c.~2~
fee.c (linked to fee.c.~2~)
```

fee.c.~1~  
fee.c.~2~

and you rename "foo.c" to "fee.c", your renamed {DSK} files and the linked SunOS files would appear as:

**From {DSK}**

foo.c;1  
fee.c;1  
fee.c;2  
fee.c;3

**From SunOS**

foo.c (linked to foo.c.~1~)  
foo.c.~1~  
fee.c (linked to fee.c.~3~)  
fee.c.~1~  
fee.c.~2~  
fee.c.~3~ (renamed file)

- When a file has a name suffix that is not a valid version number (e.g., myfile.~12x~), the suffix is regarded as part of the file name.

**From {DSK}**

myfile.~12x~;1

**From SunOS**

myfile.~12x~

## Pathnames

A pathname on {DSK} is always case insensitive. When the user specifies a file, the {DSK} device handler first searches for the file with the specified name. If no such file is found, it then searches for a file with the same spelling but different case.

Most Lisp functions, such as `FINDFILE` and `INFILEP`, which return pathnames return them with the original case when they are applied on files on {DSK} when `IL:*DSK-UPPER-CASE-FILE-NAMES*` is `NIL`. If `IL:*UPPER-CASE-FILE-NAMES*` is not `NIL`, these functions return only uppercase pathnames. The only exception is the function `DIRECTORY`, which returns a list of pathnames. The case of the pathnames is controlled by the global variable `IL:*UPPER-CASE-FILE-NAMES*` in a similar manner to `IL:*DSK-UPPER-CASE-FILE-NAMES*`.

If a pathname on {DSK} has no directory specification, a tilde-slash combination (`~/`) is used, i.e., the Lisp directory specification `{DSK}foo` is the equivalent of `{UNIX}~/foo`.

## {UNIX} Naming Conventions

For the {UNIX} device, file name translation takes place only on the directory. An initial left angle bracket (`<`) is treated as if it were an initial slash (`/`); both signify a path relative to the SunOS file system root directory; if there is no initial left angle bracket or slash, the directory is relative to the current working directory. Initially this is the working directory where Lisp was started; you can change it using the `CHDIR` function, described below. Tilde (`~`) is translated to the user's home directory.

For example, `{UNIX}myfile/abc` means the file `abc` on the `./myfile` directory.

The {UNIX} device does not recognize version numbers, does not return them, and ignores them for recognition.

No case translation or recognition is done; upper- and lowercase letters are treated as distinct.

Examples:

<b>{UNIX} Name From Lisp</b>	<b>File Name From SunOS</b>
<foo>fee>bar.baz;1	/foo/fee/bar.baz;1
<foo>fee/bar.;1	/foo/fee/bar.;1
<foo/fee>	/foo/fee/
</foo/fee/>	/foo/fee/
/foo/fee/bar.~1~	/foo/fee/bar.~1~
/foo/fee/	/foo/fee/

In the first two examples the ;1 is treated as part of the file name, not the version number. In the last two examples that translation is not done.

## Directories

In places where Lisp expects a directory name, {UNIX} paths must end with a slash (/).

### Directory Enumeration

You cannot use the wildcard character, asterisk (\*), in subdirectories for either {DSK} or {UNIX} devices. For example

```
>(DIRECTORY ' {DSK}/users/x*/foo)
NIL
```

Enumeration of files in directories differs between {DSK} and {UNIX} devices. On the {DSK} device, a versionless file which has a link to the highest version file is not enumerated in a directory.

On the {UNIX} device, all files are enumerated in a directory. For instance, if the following SunOS files linked with foo.c.~2~ exist

```
foo.c
foo.c.~1~
foo.c.~2~
```

the {DSK} directory enumeration would look like this:

```
>(DIRECTORY ' {DSK}/users/venue/*)
( {DSK}/users/venue/foo.c;1
  {DSK}/users/venue/foo.c;2)
```

The {UNIX} directory enumeration, on the other hand, would look like this:

```
>(DIRECTORY ' {UNIX}/users/venue/*)
( {UNIX}/users/venue/foo.c
  {UNIX}/users/venue/foo.c.~1~
  {UNIX}/users/venue/foo.c.~2~)
```

### Directory Creation

{DSK} When you write a new file on {DSK}, if the directory named in a pathname does not exist, the {DSK} device handler creates the directory automatically. This feature is provided for compatibility with other Interlisp-D implementations.

If you try to "connect" to a nonexistent directory (using the `CONN Exec` command or the function `IL:CNDIR`), Medley returns the message

```
Nonexistent directory
```

{UNIX} The {UNIX} device does not support such directory creation. An attempt to create a file on a nonexistent directory results in an error.

```
{UNIX}/users/venue/foo.c.~2~)
```

## Directory Deletion

Neither {UNIX} nor {DSK} support automatic directory deletion. To delete a directory you must use the SunOS C-Shell command `rmdir`.

## Open File Limit

The number of simultaneously open {DSK} and {UNIX} files must fall within the SunOS limits for a process. For OS 3.4, this number of open files may be configured, with 30 as the maximum permissible number of open files per process. This means that it is not possible to have more than 30 files open for a process, minus whatever files Medley has open for its own use, at any one time in the Medley system. If you try to open too many files, the system call error number 24, Too many open files, appears in the prompt window.

For OS 4.0, the maximum number of files/processes that can be open at one time is 64, unless your kernel is configured otherwise.

## Default Pathname

If no path is given, the {DSK} device defaults to the user's home directory, tilde-slash (~/). The {UNIX} device defaults to the current working directory. This current working directory can be changed with the `CHDIR` function. The current working directory is also used to resolve the interpretation of the period (.) and double period (..) specifications at the beginning of a {DSK} pathname.

**(IL:CHDIR *PATHNAME*)** **[Function]**

---

Changes the current working directory for the current invocation of Lisp. For example,

```
(CHDIR "{DSK}~/subdir/")
(OPENSTREAM "{DSK}./foo" ...)
```

opens the SunOS file `~/subdir/foo`.

When *PATHNAME* does not end with a slash (/) or right angle bracket (>), the whole *PATHNAME* is treated as a directory name:

```
(CHDIR '{DSK}<users>local>')
> "{DSK}<users>local>"
(CHDIR '{UNIX} /usr/local)
> "{UNIX}<usr>local>"
```

If *PATHNAME* is `NIL`, `CHDIR` tries to change the current working directory to the current connected directory. If the directory is connected to devices other than {DSK} or {UNIX}, the error message

```
Bad Host Name
```

appears, followed by the host name of the current connected directory.



If the *PATHNAME* does not exist, the error message  
     No-Such-Directory  
 appears followed by the system echo of the pathname.

## File Attributes

---

This section describes how the various file attributes are treated by Lisp on the Sun Workstation and what they translate to in SunOS.

GETFILEINFO obtains file attributes and SETFILEINFO sets the attributes.

### **WRITEDATE and CREATIONDATE** **[File Attributes]**

---

Resets the date to the current time whenever the contents of a file are modified. This only works for the owner of the file. Since UNIX does not naturally support more than one date for file modification, the WRITEDATE and CREATIONDATE are treated identically by Lisp functions OPENSTREAM, OPENFILE, GETFILEINFO, and by the {DSK} and {UNIX} devices.

### **TYPE** **[File Attribute]**

---

Sets the TYPE property of files; normally either TEXT or BINARY. However, UNIX does not distinguish between TEXT and BINARY files. Normally, programs will infer the type by the file extension, using the Lisp variables DEFAULTFILETYPE and DEFAULTFILETYPELIST. This is the convention used by Medley. If no file extension is given, the value in DEFAULTFILETYPE is used. SETFILEINFO cannot change the TYPE attribute.

### **EOL** **[File Attribute]**

---

Returns the end-of-line convention. Both the {DSK} device and {UNIX} use line feed (LF) as the default EOL convention for text. The EOL for binary files is carriage return (CR). EOL uses the TYPE property of files. (The TYPE property of a file depends on the file extension and the DEFAULTFILETYPE and DEFAULTFILETYPELIST variables). If the TYPE property of a file is TEXT, LF (=10) is used as EOL. If the TYPE property of a file is BINARY, CR (=13) is used as EOL.

NOTE: EOL conventions on {DSK} are not compatible with those on Xerox workstations.

### **AUTHOR** **[File Attribute]**

---

Returns the author of the file, i.e., the login name of the user who created it. This attribute cannot be changed.

### **PROTECTION** **[File Attribute]**

---

Returns file protection attributes. The file protection attributes of files under the SunOS cannot be directly manipulated from inside Lisp. It is necessary to use the UNIX chmod command to change file protection bits.

**SIZE** **[File Attribute]**

---

Returns the file size. For compatibility with other Lisp environments running on Xerox workstations, the `SIZE` attribute is computed as the length of the file (in bytes) divided by 512 (rounded up).

NOTE: `SETFILEINFO` lets you change the `SIZE` attribute of I/O streams and output streams. However, a file cannot be expanded this way.

## File Variables

---

This section discusses how certain file variables are used by Medley in SunOS.

**IL:FileTypeConfirmFlg** **[Variable]**

---

The file-type attribute of a file on `{DSK}` or `{UNIX}` is decided from its extension, `DEFAULTFILETYPEPELST` and `DEFAULTFILETYPE`. Extensions of binary files should be registered in `DEFAULTFILETYPEPELST`. When this rule is broken, a hardcopy of files on `{DSK}` and `{UNIX}` may confuse the printers. So when you try to hardcopy a file whose extension is not registered in `DEFAULTFILETYPEPELST`, a menu is invoked to confirm the file type. Text or binary can be selected. The invocation of this menu can be stopped by setting `IL:FileTypeConfirmFlg` to `NIL`. The default value of `IL:FileTypeConfirmFlg` is `T`.

When extensions of binary files are not registered in `DEFAULTFILETYPEPELST`, copy or rename from a `DSK/UNIX` device to a non-`DSK/UNIX` device also may cause file type confusion. This type of copy or rename results in one of the following warning messages in the prompt window, as appropriate:

**Extension of `{DSK}foo.fee;1` isn't in `DEFAULTFILETYPEPELST`. `{CORE}foo.fee;1` was copied as `TEXT`.**

**This message can be stopped by set `FileTypeConfirmFlg` to `NIL`.**

**Extension of `{DSK}foo.fee;1` isn't in `DEFAULTFILETYPEPELST`. `{CORE}foo.fee;1` was renamed as `TEXT`.**

**This message can be stopped by set `FileTypeConfirmFlg` to `NIL`.**

Either of these messages can be stopped by setting `IL:FileTypeConfirmFlg` to `NIL`.

**IL:DEFAULTFILETYPE** **[Variable]**

---

Initially set to `TEXT`. Used with the file attribute `TYPE`.

**DEFAULTFILETYPEPELST** **[Variable]**

---

A list of accepted file types. Initially set to `((NIL . TEXT)(C . TEXT)(H . TEXT)(LISP . TEXT)(LSP . TEXT)(O . BINARY)(OUT . BINARY)(LCOM . BINARY)(DFASL . BINARY)(DCOM . BINARY)(SKETCH . BINARY)(TEDIT . BINARY)(DISPLAYFONT . BINARY)(WD . BINARY)(IP . BINARY)(RST . BINARY)(BIN . BINARY)(MAIL . BINARY)(SYSOUT . BINARY))`

Used with the file attribute `TYPE`. Binary files, such as Sketch files, InterPress files, or Press files, should have their extensions registered in `DEFAULTFILETYPEPELST`. This is especially important because UNIX does not support file types.

## File System Errors

Several types of errors may occur in the Medley file system.

When a remotely mounted file system or NFS service is down, or when network traffic is heavy, any attempt to access a file on that file system results in an error. The following error message is printed in the prompt window:

```
File access timed out
```

Medley will wait until the file system responds or until a timeout occurs. If the file system is mounted with the "hard" option, the timeout is controlled by the value of environment variable `LDFILETIMEOUT`. If the file system is mounted with the "soft" option, the timeout depends on the NFS file system timeout time, and the value of `LDFILETIMEOUT`. Medley will wait until the shorter of these two times is exceeded. The NFS file system timeout time, retry times, etc., are controlled by the UNIX command **mount**.

If `LDFILETIMEOUT` is not set, the default value of 10 seconds is used. The variable is inspected at boot time, and a setting between 1 and 100 seconds is appropriate in most cases.

The following error messages may appear when there are Medley file errors:

```
Not owner
Device error:
Protection-violation
File-won't-open
Too-Many-Files-Open
File too large
File-System-Resources-Exceeded
Connection timed out
No-Such-Directory
Bad Host Name
FS-RENAMEFILE-SOURCE-COULDNT-DELETE
```

Another type of error occurs when the user has insufficient access to files. When this happens, Medley will print the following message:

```
File not found
```

The following message then appears in the SunOS prompt window:

```
System call error: open errno=13 Permission denied
```

See the *UNIX Interface Reference Manual*, Intro (2), for descriptions of all OS system call messages.

[This page intentionally left blank]

## 6. ERROR RECOVERY

---

Medley on the Sun Workstation has an error handling system which includes the following:

- The Xerox Lisp error system, described in the IRM
- A diagnostic program, URAID, which handles emulator errors

Occasionally, you may encounter SunOS error messages. Refer to your Sun documentation set for recovery procedures when these errors occur. When running Medley on a Sun Workstation, previous Lisp error handling such as Teleraid and MP errors are no longer available. However, you can still use Teleraid from a Sun Workstation to debug a Xerox 1100 series workstation.

### URAIID

---

The Medley system normally operates as a self-contained environment. In some unusual circumstances Medley may encounter a situation from which it cannot recover. In this case, when an unrecoverable emulator error is encountered, the emulator halts and enters into a small debugger called URAID. URAID allows you to inspect memory, or to look inside the sysout file, and attempt to recover from the error.

If you produce the same type of error condition in Medley on a Sun Workstation as you did on a Xerox workstation, you get a URAID error instead of an MP error.

#### Entering URAID

Normally, the emulator automatically enters URAID when an unrecoverable emulator error occurs. However, there are two additional methods available when you want to enter URAID directly.

- Use the SHIFT-CTRL-DELETE key combinations to enter URAID between opcodes. Note that the DELETE key referred to here is in the L10 position on the left keypad of the Sun keyboards. This sequence allows you to return to Lisp later.
- Use SHIFT-CTRL-NEXT for emergency interrupts only. Note that the NEXT key is in the ALTERNATE key position on the regular Sun keyboard. These combinations are useful for exiting from an opcode infinite loop. SHIFT-CTRL-NEXT does not necessarily enter URAID between opcodes; once you are in URAID mode, another URAID command such as "f" could cause the emulator to crash. At this point it is unlikely that you could return to Lisp. USE WITH CAUTION!

#### Conventions

URAIID uses these display conventions:

- Numbers are displayed in hexadecimal unless otherwise noted.
- The *litatom* should be an uppercase string when used with a package prefix (e.g., XCL: EVAL).
- Symbols are displayed with a package prefix, but with no escape character.

- Symbols in the Interlisp (IL:) package are case-sensitive (e.g., IL:\InterfacePage); symbols in other packages are case-insensitive (e.g., XCL:EVAL).

In addition, these input conventions apply:

- Symbols may only be qualified by their home package.
- A full package name may prefix an input symbol. URAID also supports approved abbreviations of package names (e.g., XCL:, SI:, CL:, XCLC:).

A symbol without a prefix is treated as a symbol in the Interlisp package. For instance, `\InterfacePage` is the equivalent of `IL:\InterfacePage`.

- Type-in is uppercase for symbols in any package except the Interlisp package; type-in is in mixed case for `IL:` package symbols or symbols with no prefix.

## URAIID Commands

URAIID has a few simple commands which you can use to attempt diagnosis and error recovery. All URAID commands are case-sensitive.

- h** Hard Reset. Attempts to recover by resetting the Lisp stack. Quits URAID and causes Lisp to resume execution. This command should not be used unless you are sure that execution can be resumed.
- e** Exits to SunOS. Medley will end.
- q** Quits URAID and returns to Lisp.

NOTE: An error may occur while the Medley system is running uninterruptably. The following message signals this error:

```
Error in uninterruptable system code -- ^N to continue into
error handler
```

Disregard the `^N` command; it is not supported by URAID. Use the **q** command to continue.

## Displaying a Stack

For casual users, the **l** command followed by several **f** commands generally provides the most useful information. Many of the other commands require some knowledge of the internal representation of Lisp objects and stack frames.

- c** Checks all user stack contents; stack inconsistency is displayed.
- k type** Changes the stack link that precedes the **l** command to be *type*, which is either **a** (to follow ALinks) or **c** (to follow CLinks). The default is to trace ALinks. ALinks follow the chain of free variable access.
- l type** Shows the stack as a back trace consisting of a numbered sequence of frame names. The default is the user stack. The argument *type* is a single letter denoting the stack to view. The system has a number of special contexts, which are areas of stack space used by certain system routines. Legal values of *type* are as follows:
  - g** (garbage collect)
  - k** (keyboard handler)
  - m** (miscellaneous)
  - p** (page fault)
  - r** (reset)
  - u** (user stack) - Default

*type* := g|k|m|p|r|u or nil

- C** Checks the contents by scanning all stack space in the sysout. For example:

```
0x11880 BF,[ivar:0x1800]
0x11802: FX for CL:T[ ]
0x11816 BF,[ivar:0x1816]
0x11818: FX for IL:\TURN.ON.PROCESSES[ ]
```

## Viewing Frames From a Stack

After displaying a particular stack with the **I** command, the following commands view individual frames from that stack:

- f** *number* Displays the contents of frame *number* (decimal) with its basic frame, IVars and PVars. The frame is printed in two parts, a basic frame containing the function's arguments and a frame extension containing control information, the function's local (PROG) variables, and dynamic values. On the left side of the printout are the hexadecimal contents of each cell of the frame, with an interpretation, usually as a Lisp value, on the right. The following message appears as you display a frame with the **f** command:
- ```
Press Return (To quit ESC and RET)
```
- To abort the printing of a frame, first press the ESC key then the RETURN key. The URAID prompt "<" reappears.
- <CR>** Displays the next frame (closer to the root, or bottom, of the stack). This is the same as **f** *n*+1, where *n* is the number of the frame most recently viewed. Immediately after an **I** command, *n* is zero, so **<CR>** views the first frame.
- a** *litatom* Displays the top-level value of the *litatom*
- d** *litatom* Displays the contents of definition cell for the *litatom*. If it is compiled code, this command prints a CCODEP hexadecimal address pointer; for example,
- ```
{CCODEP}0x14ccc4
```
- Otherwise, it prints a Lisp definition; for instance, interpreted code returns
- ```
( LAMBDA ( ) ... )
```
- M** Displays TOS, CSP, PVar, IVar, PC.
- m** *func1 func2* Moves the definition of *func1* to *func2*.
- t** *Xaddress* Displays the type of this object.
- p** *litatom* Displays the contents of the *litatom*'s property list.
- w** Displays the current function name and PC.
- x** *Xaddress[Xnum]* Prints *Xnum* word (16-bits) of the raw contents of the virtual memory starting at virtual address *Xaddress*. This is most useful for examining the contents of a datatype which other commands simply print as its virtual address.



- @litatom[*snumber*/NIL/*T*]* Sets the TOPVAL of *litatom* to the specified value. *snumber* is a signed smallp number.
- <Xaddress val* Sets the the contents of the word (16-bits) at the *Xaddress* to *val*.

### Miscellaneous

- v filename* Saves the current virtual memory on the *filename*. This file can be examined using the functions READSYS and VRAID in the TeleRaid Lisp Library module, but cannot be used as a sysout file.
- NOTE: This sysout cannot be restarted.
- s** Invokes a subshell.
- (num* Sets the print level (default is 2).
- ?** Displays this summary.
- !** Prints the error message passed from the emulator.

## Other Fatal Error Conditions

Occasionally, other emulator, operating system, or system administration errors may occur from which the URAID program cannot recover. Such error conditions include the process dying, the emulator going into an infinite loop, the keyboard being lost, or the system freezing up.

If any of these emulator errors occur, use the UNIX `kill` command to kill the `lde` process.

## Lisp Errors

### Errors While Running Medley

The following Lisp errors may occur when running Medley on the Sun Workstation.

| <b><u>ERROR MESSAGE</u></b> | <b><u>CAUSE</u></b>                                                                                                                                                                                                                                |
|-----------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| File access timed out       | Occurs when you try to access a file when the remotely mounted file system or NFS service is down, or when network traffic is heavy. See the File System Errors subsection of Chapter 5.                                                           |
| File too large              | Self-explanatory.                                                                                                                                                                                                                                  |
| Too-Many-Files-Open         | Occurs when you exceed one of the following: <ul style="list-style-type: none"> <li>• SunOS open file limit (see Chapter 5, Medley File Systems)</li> <li>• System file resources while writing a sysout (using <code>IL:SYSOUT</code>)</li> </ul> |

|                                     |                                                                                                              |
|-------------------------------------|--------------------------------------------------------------------------------------------------------------|
| Nonexistent directory               | Occurs when you try to connect to a nonexistent directory using the IL:CNDIR function or the CONN command.   |
| No-Such-Directory                   | CHDIR                                                                                                        |
| Connection timed out                | Self-explanatory.                                                                                            |
| Bad Host Name                       | Self-explanatory.                                                                                            |
| FS-RENAMEFILE-SOURCE-COULDNT-DELETE | Occurs when you try to rename a file which exists on a directory or which you do not have delete permission. |

### Xerox Workstation-Specific Errors

These Xerox workstation-specific errors may occur if certain functions are inadvertently used on the Sun Workstation.

| <u>ERROR MESSAGE</u>                     | <u>CAUSE</u>                                                                                                    |
|------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| Floppy: No floppy drive on this machine. | Self-explanatory.                                                                                               |
| Device error: {FLOPPY}                   | Occurs when the user tries to enter a Lisp floppy function while running on the Sun Workstation.                |
| Wrong machinetype                        | Occurs when functions controlling Xerox disk drive device-specific behavior are entered while running in SunOS. |

### Virtual Memory Errors

| <u>ERROR MESSAGE</u>           | <u>LISP FUNCTION RESPONSIBLE</u> |
|--------------------------------|----------------------------------|
| File-System-Resources-Exceeded | IL:SYSOUT, IL:LOGOUT, IL:SAVEVM  |
| Protection-Violation           | IL:SYSOUT, IL:LOGOUT, IL:SAVEVM  |
| File-Wont-Open                 | IL:SYSOUT, IL:LOGOUT, IL:SAVEVM  |

[This page intentionally left blank]

# APPENDIX A. INSTALLATION HINTS

---

## Medley Shell Variables

---

The following is a fragment of a `.cshrc` file which you may want to adapt to your own needs. In this example Smythe works in Building 12b (bldg12b), and always wants a fresh sysout, containing Rooms, loaded.

```
# =====  
# Set up various Medley variables.  
setenv LDEDESTSYSOUT /user/smythe/sysouts/saved.virtualmem  
setenv LDESRCESYSOUT /usr/share/lde/lispsysouts/ROOMS.SYSOUT  
setenv LDEINIT      /usr/share/lde/site-files/bldg12b-init.lcom  
# Assuming you are using UNIXChat and VTChat,  
#   configure the Chat window  
if ($?LDESHELL == 1) then  
    setenv TERM vt100  
    stty erase ^H  
endif  
# =====
```

## Running on Multiple Workstations

---

### Installation for Sites with Sun-3 and Sun-4 Workstations

In Medley 2.0, the only differences between the Sun-3 and Sun-4 distributions are in the `install.sunosX` directories. Thus, during installation the common subdirectories (`lispsysouts`, `lisplibrary`, `fonts`, etc.) might be installed instead to a shared file system, saving 15 MB of unnecessary duplicated space. In the example below, `/sharedserver` is a remote file system mounted on the local machine.

```
prompt% mkdir /sharedserver/lde  
prompt% cd /sharedserver/lde  
prompt% tar xvfb /dev/rxx0 126 ./lispsysouts ./lisplibrary  
      ./fonts
```

If soft links are then left on `/usr/share/lde`, the installation can proceed as before.

```
prompt% ln -s /sharedserver/lde/lispsysouts  
      /usr/share/lde/lispsysouts  
prompt% ln -s /sharedserver/lde/lisplibrary  
      /usr/share/lde/lisplibrary  
prompt% ln -s /sharedserver/lde/fonts /usr/share/lde/fonts
```

Otherwise, the site initialization file needs to be changed appropriately.

The install directories are left on `/usr/share/lde`, since those directories are typically local to a particular processor architecture.

```
prompt% cd /usr/share/lde
prompt% tar xvfb /dev/rxx0 126 ./install.sunos4
```

## Using a "runlde" on Multiple Workstations

The following is an example of a `runlde` script that might be used for running Medley on different machines.

```
# (invokes CSH)
# =====
# Usage: runlde optional-sysout
#
# The script below is for the following machines:
#
#   Host   HostID
#   ----   -
#   timber      1700319b
#   gopher      17003016
#   tree        13003565
# =====

switch ("`hostid`")
  case '1700319b':
    ldeether $1 -k '99e8bfc6 92299f45 9199a409'
    breaksw
  case '17003016':
    ldeether $1 -k '70c5a8d8 7b0498cc 45e35500'
    breaksw
  case '13003565':
    ldeether $1 -k 'ce7627bf b5b61ac8 2f990cc0'
    breaksw
  default:
    echo "Sorry, host '`hostname`' is not in this shell
    script"
endsw
```

---

## Configuring the Software

The software comes in these two forms:

- An executable binary image for users who have not modified the Sun kernel too extensively
- An object file that can be relinked for your particular system.

If you want to use the executable that Venue supplies, skip to the `Enabling PUP/XNS Ethernet` subsection below.

## Relinking

If you have tried the prelinked software and it doesn't work, link the object code with the Sun libraries. To do this, you need the `suntool`, `sunwindow`, and `pixrect` libraries, and `make`, `cc`, etc., available on your search path. To configure the system, connect (`cd`) to the directory `usr/share/lde/install.sunosx` (where `x` is the version of SunOS that you are running, e.g., SunOS 4.0 in the following), and type `make`.

```
prompt% cd /usr/share/lde/install.sunos4
prompt% rm lde ldeether; make
```

This procedure replaces the two executable programs, `lde` and `ldeether`. The program `ldeether` enables access to Xerox network protocols from Lisp.

## Enabling PUP/XNS Ethernet

If you intend to use the PUP or XNS Ethernet directly from Medley, you need to change file ownership and permissions of `ldeether`. Note that you do this on the server where `ldeether` is actually residing (root permission must be on the server). Log in to the machine where `ldeether` resides. To find out where `ldeether` resides, type:

```
prompt% df filename
```

where `filename` is the pathname of `ldeether`. The system responds with the name of a file system (e.g., `/dev/sd0g`) for a local file, or with a machine name and directory (e.g., `python:/user1`) for an NFS file.

Now you can change the `ldeether` file ownership and permissions.

```
prompt% rlogin server
server% su
server# cd /usr/share/lde/install.sunos4
server# /etc/chown root ldeether
server# chmod 4755 ldeether
server# exit
```

If you are using the Ethernet, substitute `ldeether` whenever `lde` appears in the instructions below.

---

## Using NIS to Manage the Keys for Multiple Workstations

---

Here is an example how to handle several Medley licenses on a network, by using the Sun Network Information Service (NIS).

Create a file containing an association list of hostnames vs. license keys, for each host that has a Medley license. For example:

```
# medley-keys.by-hostname
# =====
king          6a1c33bf 11dc1a48 a4c34080
sidewinder    7b636e98 55a26cd4 26b80560
hognose       190750c0 17c658e0 08060ac0
boa           8334d182 00793e07 4903890b
asp           c90faa4f d3477c53 d304b85b
rattler       70b8fd18 2d79f344 c30051c0
```

NOTE that the following commands should all be run as `root`.

On your NIS master server, create an NIS database of hostname vs. Medley keys:

```
prompt% /usr/etc/yp/makedbm ./medley-keys.by-hostname \
      /var/yp/your-domain/medley-keys
```

Replace *your-domain* with the name of your NIS domain. The output is put in the directory containing your master NIS maps.

If you have NIS slave servers serving your domain, you will need to update each one manually the first time the map is created. Thereafter, they will be updated automatically. On each NIS slave server do the following:

```
% /usr/etc/yp/ypxfr -f -h your-NIS-master medley-keys
```

Replace *your-NIS-master* with the name your NIS master server.

After updating all NIS slave servers, you now need to propagate the NIS map to your NIS clients. On your NIS master, type:

```
% /usr/etc/yp/yppush medley-keys
```

From now on, any changes made to the `medley-keys.by-hostname` file will only require the propagation of the map to your NIS clients. The following steps are required:

1. Create a new NIS map using the `makedbm` command as described above.
2. Propagate the changes to your NIS clients using the `yppush` command as described above.

You can now use the newly created map. Below is an example of a `runlde` script that uses the newly created NIS map.

```
#!/bin/csh -f
# =====
# Usage : runlde [sysout]
#
# Script for running Venue Medley software.
#
# =====
if ($#argv > 1) then
    echo "Usage : runlde [sysout]"
endif

set SYSOUT = "$1"
set HOSTNAME = `bin/hostname`
set KEY = `bin/ypmatch $HOSTNAME medley-keys`
```

```
if ! $status then
  ldeether $SYSOUT -k "$KEY"
endif
```

Consult the *Sun Network and Communications Administration* manual for more details about NIS and how to add the new map to the `/var/yp/Makefile`.



[This page intentionally left blank]

## APPENDIX B. VERIFYING THE INSTALLATION TAPE'S VALIDITY

---

If you encounter inexplicable problems shortly after you install Medley, they may be due to files being corrupted — the release tape may have been damaged, errors may have occurred while the tape was being read, etc. If you have unexplained problems, we recommend that you verify the checksums of your installed files.

The script generates checksum files named `*.check` and compares them to the released `*.sum` residing in the `/checksumdir` subdirectory.

The checksum script reports inconsistent files, the correct checksum values for the files, and an error message. The checksum of individual files can be generated with the UNIX command `sum filename`.

`ldechecksum [-cg] medleydir [ dir / dirgroup ]` **[Command]**

|                        |                                                                                                                                                                                              |
|------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>-c</code>        | Generates checksums for your installed files and compares them with correct values. This is the default action.                                                                              |
| <code>-g</code>        | Generates checksums for the files specified.                                                                                                                                                 |
| <code>medleydir</code> | Name of the Medley installation directory. Default is <code>/usr/share/lde</code> .                                                                                                          |
| <code>dir</code>       | Any specific directory residing under <code>medleydir</code> . Only relative pathnames with respect to <code>medleydir</code> are accepted.                                                  |
| <code>dirgroup</code>  | The directory group, either <b>all</b> (the default) or <b>lisp</b> , which includes the <code>X/install.xxxx</code> , <code>X/lisplibrary</code> and <code>X/lispysouts</code> directories. |

### Output

As it begins checking each directory, the script prints a message in the form:

```
Checking directory: /usr/share/lde/subdir
```

Error and warning messages may be in one of two forms:

```
< E > 32711 49 4045XLPSTREAM.DFASL
```

indicates that file `4045XLPSTREAM.DFASL` is erroneous or does not exist in the directory. The correct checksum of `32711`, together with the size (49 Kbytes) of the file, are shown.

```
< W > /usr/share/lde/fonts/display/chinese : Directory not installed
```

indicates that Chinese fonts were not installed or were removed after Medley was installed.

### Examples

```
prompt% ldechecksum /usr/share/lde
```

All files in the installed Medley directories in `/usr/share/lde` are checked.

```
prompt% ldechecksum /usr/share/somedir/lde lisp
```

This example checks all files in:

```
/usr/share/somedir/lde/install.xxxx  
/usr/share/somedir/lde/lisplibrary  
/usr/share/somedir/lde/lispsysouts
```

```
prompt% cd/usr/share/lde
```

```
prompt% ldechecksum -c . fonts/display
```

This example checks only the display font directories. The period (.) is used because you are positioned under the current Medley installation directory.

[This page intentionally left blank]

# APPENDIX C. LAYOUT OF INSTALLATION TAPE FILES

## Layout of Installation Tape

Below follows the layout of the Medley Installation Tape with a description of the individual files.



| <b>File</b> | <b>Contents</b>                                                                                                                           | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
|-------------|-------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1           | <code>./install-medley</code>                                                                                                             | The Medley installation utility                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| 2           | <code>./medley</code>                                                                                                                     | The Medley startup script                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| 3           | <code>./install.sunos3/</code> (only on the Sun3 installation tape)<br><code>./install.sunos4/</code><br><code>./install.sunos4.1/</code> | Each subdirectory contains:<br><br><code>lde</code> Used as a bootstrapper to load the right emulator, depending on the frame-buffer of your host and whether X Windows is running.<br><br><code>ldeether</code> Used when you want to use the XNS protocol from within Medley on a Sun. It will set up your system to intercept XNS and PUP packets and then immediately runs <code>lde</code> .<br><br><code>ldesingle</code> The emulator used to run Medley on a workstation with a monochrome display or one with a color frame-buffer of type <code>cg2</code> , <code>cg4</code> , or <code>cg9</code> .<br><br><code>ldemulti</code> The emulator used to run Medley on a workstation with a color frame-buffer of type <code>cg3</code> or <code>cg6</code> .<br><br><code>ldex</code> The emulator used to run Medley on a workstation where an X Windows server is running.<br><br><code>ldesingle.o</code><br><code>ldemulti.o</code><br><code>ldex.o</code> These object files are used when recompiling the emulators to either include your own C subroutines or when problems arise.<br><br><code>makefile</code><br><code>usersubrs.c</code> Used when you wish to link your own C subroutines into the emulator (a non-documented feature).<br><br><code>ldeether.c</code> The source code for the <code>ldeether</code> . Its only purpose is to allow you to recompile the ethernet set-up code should you run into any problems. |

|          |                                 |                                                                                                                                                  |
|----------|---------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>4</b> | <code>./lisplibrary</code>      | <b>Contains all the Medley 2.0 Lisp Library files</b>                                                                                            |
| <hr/>    |                                 |                                                                                                                                                  |
| <b>5</b> | <code>./checksumdir</code>      | <b>Contains <code>ldchecksum</code>, <code>checksum</code> and <code>X.sum</code> checksum files (See Appendix B for a detailed explanation)</b> |
|          | <code>./lispsysouts</code>      | <b>Contains the <code>sysout</code>, <code>lisp.sysout</code></b>                                                                                |
| <hr/>    |                                 |                                                                                                                                                  |
| <b>6</b> | <code>./fonts/display</code>    | <b>Contains the display fonts (See Table C-1 for a detailed description of the individual font files)</b>                                        |
|          | <code>./fonts/interpress</code> | <b>Contains the Interpress printer fonts (See Table C-1 for a detailed description of the individual font files)</b>                             |

## Font Directories

Table C-1 shows the organization of the font directories, as well as the descriptions and contents of the directories.

**Table C-1. Font Directories**

| <b>Directory Name</b>                                             | <b>Description</b>                                                                   | <b>Font Families</b>                                                  | <b>Font Types</b>                                                                                                                                                                    |
|-------------------------------------------------------------------|--------------------------------------------------------------------------------------|-----------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| ./fonts/display/presentation<br>./fonts/interpress/presentation   | All presentation fonts for display and user interface applications                   | Helvetica<br>Gacha<br>Times Roman                                     | Sans serif<br>Monospace screen font in 8, 10, 12 MRR<br>Serif                                                                                                                        |
| ./fonts/display/publishing<br>./fonts/interpress/publishing       | All publishing fonts for character sets, foreign characters, and technical alphabets | Classic<br>Modern<br>Terminal                                         | Serif; in all character sets, sizes, faces<br>sans serif; in all character sets, faces, but with selected sizes<br>Monospaced, in all character sets, faces, but with selected sizes |
| ./fonts/display/printwheel<br>./fonts/interpress/printwheel       | All printwheel fonts for word processing applications                                | BoldPS<br>LetterGothic<br>Titan                                       | Proportional serif<br>Monospaced sans serif<br>Monospaced serif                                                                                                                      |
| ./fonts/display/JIS1<br>./fonts/interpress/JIS1                   | Japanese Kanji fonts, character set 1                                                | Classic                                                               | Point sizes 8 through 24                                                                                                                                                             |
| ./fonts/display/JIS2<br>./fonts/interpress/JIS2                   | Japanese Kanji fonts, character set 2                                                | Classic                                                               | Point sizes 8 through 24                                                                                                                                                             |
| ./fonts/display/chinese<br>./fonts/interpress/chinese             | Chinese character fonts                                                              | Classic<br>Modern                                                     | Point sizes 12 and 24<br>12 point                                                                                                                                                    |
| ./fonts/display/miscellaneous<br>./fonts/interpress/miscellaneous | Miscellaneous fonts for nonstandard and rare applications                            | ClassicThin<br>Hippo<br>Logo<br>Math<br>OldEnglish<br>Symbol<br>Tonto | Brackets and parentheses in point sizes 16, 20, 26, and 30<br>Greek or Latin<br>Xerox logo<br>Math symbols<br>Point sizes 10 and 18<br>Math symbols<br>Thick monospaced 14 point MRR |
| ./fonts/press                                                     | All metric information for Press printers.                                           |                                                                       |                                                                                                                                                                                      |

## Manually Extracting Files from the Installation Tape

---

You can manually extract individual files or directories from the Medley installation tape. For example, if you want to extract the X-window emulator `ldex` for SunOS release 4.1 from the tape do the following:

```
prompt% mt -f /dev/nrst0 rewind
```

Ensures that the tape is positioned at the beginning of the tape.

```
prompt% mt -f /dev/nrst0 fsf 2
```

Positions the tape at the beginning of the third file on the tape. The `n` in the `/dev/nrst0` makes sure the tape is not rewound after the command has been completed.

```
prompt% tar xvf /dev/nrst0 ./install.sunos4.1/ldex
```

Extracts `ldex` from the Medley installation tape and puts it in your current working directory.



# APPENDIX D. DIFFERENCES BETWEEN XEROX WORKSTATIONS AND THE UNIX VERSION OF MEDLEY

---

## Local Disk and Floppy Functions

---

The functions for controlling device-specific behavior of the Xerox 1100 series workstation disk drives are not supported. These functions signal the error

Wrong machinetype

if called when running under UNIX. These functions include

IL:PURGEDSKDIRECTORY  
IL:CREATEDSKDIRECTORY  
IL:VOLUMESIZE  
IL:DISKFREEPAGES  
IL:DISKPARTITION  
IL:SCAVENGEDSKDIRECTORY  
IL:FILENAMEFROMID

The following functions for controlling the Xerox 1100 series workstation floppy disk drive also signal an error under UNIX:

IL:FLOPPY.FORMAT, IL:FLOPPY.NAME, IL:FLOPPY.TO.FILE,  
IL:FLOPPY.FROM.FILE, IL:FLOPPY.ARCHIVE, IL:FLOPPY.UNARCHIVE,  
IL:FLOPPY.MODE, IL:FLOPPY.FREE.PAGES, IL:FLOPPY.CAN.READP,  
IL:FLOPPY.CAN.WRITEP, IL:FLOPPY.WAIT.FOR.FLOPPY,  
IL:FLOPPY.SCAVENGE

These functions signal the error

Floppy: No floppy drive on this machine. Device error:  
{FLOPPY}

The following functions have no effect and always return NIL on UNIX:

IL:VOLUMES  
IL:LISPDIRECTORYP  
IL:DSKDISPLAY

## Library Modules Not Supported on the Sun

---

The following modules listed in the manual *Lisp Library Modules*, Medley Release, are not supported on the Sun Workstation running Medley.

### TCP, TCPCHAT, etc.

Because SunOS supports TCP/IP directly, TCP packets cannot be routed to Medley. For this reason, the TCP library modules are not supported on the Sun Workstation.

## **DLRS232C, DLTTY**

The DLRS232C and DLTTY library modules are specific to the hardware devices available on the Xerox 1100 series workstations. Serial lines and other devices can be accessed from Medley either through sub-shells, or by using the {UNIX} file device, e.g., writing to {UNIX}/dev/ttya or {UNIX}/dev/ttyb.

The following library modules are normally used with equipment attached to the Xerox 1186 RS232 serial lines:

- FX-80DRIVER
- 4045XLPSTREAM
- KERMIT
- RS232CHAT.

## **KEYBOARDEDITOR, VIRTUALKEYBOARD**

Medley does not include versions of KEYBOARDEDITOR or VIRTUALKEYBOARD library modules that know about the Sun keyboards.

VIRTUALKEYBOARD lets you bring up keyboard images that give you access to special characters via the mouse. The keyboard itself is unaffected.

[This page intentionally left blank]

# GLOSSARY

---

- access permissions\*** Determines what operations can be performed on a file.
- alias\*** A user-created C-Shell command defined in terms of other commands or programs. For example, if you type (or put in your `.cshrc` file)
- ```
alias runlde "lde ~/sysout -k xx"
```
- then when you type `runlde` to the C-Shell, it acts as if you had typed
- ```
lde ~/sysout -k xx
```
- backing store** A Xerox 1100 series workstation file, the virtual memory partition. This file stores pages as they are allocated or flushed from real memory.
- byte code emulator** A byte-code instruction interpreter. Executes the Interlisp-D virtual machine instruction set compatibly with microcode for the Xerox workstations.
- chmod\*** A program used to change access permissions of a file.
- chown\*** A program used to change ownership of a file.
- {DSK}** A host device name allowing users to access the SunOS file system. Uses conventions (e.g., version numbers and file name recognition which ignores the case of letters) similar to those used by the Xerox 1100 series workstation local disk device (`{DSK}`).
- environment variable\*** A name/value pair that is passed to subprocesses. Can be set from the shell with the `setenv` command. By convention, environment variable names use uppercase rather than lowercase letters, e.g., `LDEDESTSYSOUT`. The Medley environment variables are `LDESRCESYSOUT`, `LDEDESTSYSOUT`, `LDEINIT`, `LDESHELL`.
- home directory\*** The working directory when a user logs in.
- host access key** A special code which must be entered to Medley to run Medley software on the Sun Workstation.
- lde** Lisp development environment.
- ldeether** A program produced during the software startup procedure; runs `lde` after enabling access to Xerox network protocol.
- .login\*** The name of a file in the home directory that is read by the shell when a user first logs in. Contains C-Shell commands.
- Medley** The Venue programming environment; also, the name of the release. Supports Common Lisp and Interlisp; a library of utilities, graphics packages, applications; a complete windowing system; network protocols. Runs on both Xerox and Sun workstations.

**NFS\*** Network File System; the way SunOS handles remote file systems.

- pathnames\*** In UNIX, a position identifier of a file or directory within the file system tree structure.
- An *absolute* pathname gives the position, beginning with the root directory, of the file or directory in the file system hierarchy. Each directory in the pathname is delimited by a slash (/).
- A *relative* pathname locates the position of the desired file or directory from the working directory. Again, all directories in this pathname are delimited by the slash (/).
- root directory\*** The root of the directory tree. Designated by a slash (/) at the *beginning* of an absolute pathname. Slashes elsewhere in a pathname are simply delimiters.
- shell\*** Command interpreter (akin to the Medley Exec).
- shell script\*** A file that contains shell commands. Can be run by typing the file name provided the user has execute permission on the file.
- site initialization file** A Lisp file, used when Medley is started up. Contains standardized information about the site environment such as pointers to fonts and site parameters.
- SunOS** Sun's version of UNIX.
- suntools** A Sun system window-based program tool. A program that allows all of the Sun window-based tools to run on the screen.
- tar** A program for copying data to and from magnetic tape.
- {UNIX}** A host device name allowing users to access the SunOS file system using UNIX naming conventions. Files on the {UNIX} device have no version numbers and file name recognition distinguishes between upper- and lowercase letters.

\* Indicates a UNIX term. See UNIX documentation for full definition.

[This page intentionally left blank]

# INDEX

## A

Access key 11,13  
Asterisk 36  
AUTHOR (*File Attribute*) 38

## B

Back trace 40,42  
BACKGROUNDPAGEFREQ (*Variable*) 25  
BEEPOFF (*Function*) 27  
BEEPON (*Function*) 27  
\BeginDST (*Variable*) 12,22  
Binary files 4,38,40  
Binary image, executable A-2; 13  
Brackets  
    left angle 33  
    right angle 31  
    square 31

## C

C-Shell 32  
Carriage return 4,38,40  
Case sensitivity 33,35,39,40,41,42  
CHANGEBACKGROUND BORDER (*Function*) 27  
Characters, special 32  
CHDIR (*Function*) 35,37,44,45  
checksum 1,3, B-1  
chmod (*UNIX Command*) 38  
CLOCK (*Function*) 28  
Clocks 27  
CNDIR (*Function*) 37  
Compatibility  
    compiled-file 4  
    end-of-line convention 4  
    sysout 4  
Configuration  
    changing 11  
    software A-2  
CONN (*Command*) 37  
Console messages 28  
Conventions  
    common {DSK} and {UNIX} 32  
    {DSK} naming 33  
    fonts 3  
    Medley devices 3,17,35  
    notation 3  
    URAIID 39,41  
Copy protection 11  
CREATIONDATE (*File Attribute*) 38  
.cshrc file 13, A-1

## D

Daylight Savings Time  
    setting values for 12,22  
DEFAULTFILETYPE (*Variable*) 38,39  
DEFAULTFILETYPEPELIST (*Variable*) 38,39  
DEFAULTPRINTERTYPE (*Variable*) 12,21  
DEFAULTPRINTINGHOST (*Variable*) 12,21  
DIRECTORIES (*Variable*) 12,21  
Directory  
    changing 32,37

creation 36,37  
deletion 37  
enumeration 36  
home 32,37  
name delimiting 31  
parent 32

DISKFREEPAGES (*Function*) D-1  
DISKPARTITION (*Function*) D-1  
Display functions 27  
Display fonts, how to find 12,21  
DISPLAYFONTDIRECTORIES (*Variable*) 12,21  
DLRS232C D-2  
DLTTY D-2  
{DSK} 26,32,36, 38  
    special characters 32  
    naming conventions 33  
{DSK}INIT. 12, 16  
DSKDISPLAY (*Function*) D-1

## E

Emulator 2  
\EndDST (*Variable*) 13,22  
End-of-line convention 4,38,40  
Environment variable A-1  
    LDEDESTSYSOUT 23  
    LDEINIT 11,16  
    LDESOURCESYSOUT 14,15  
    obtaining value of 26  
EOL (*File Attribute*) 38  
Errors  
    fatal 44  
    file system 39,41  
    Lisp 44

## F

fg (*UNIX Command*) 25  
File attributes 38  
File name  
    conventions 31  
    recognition 32  
File protection bits, changing 38  
file resources, exceeding 44  
File streams 32  
File system errors 39  
File types 39  
File variables 39  
FILENAMEFROMID (*Function*) D-1  
Files  
    binary 4,38,40  
    finding 12,21  
    open 37, 44  
    text 38  
    transfer 38  
    versionless 34  
FINDFILE (*Function*) 35  
FLOPPY.ARCHIVE (*Function*) D-1  
FLOPPY.CAN.READP (*Function*) D-1  
FLOPPY.CAN.WRITEP (*Function*) D-1  
FLOPPY.FORMAT (*Function*) D-1  
FLOPPY.FREE.PAGES (*Function*) D-1  
FLOPPY.FROM.FILE (*Function*) D-1



**FLOPPY.MODE** (Function) D-1  
**FLOPPY.NAME** (Function) D-1  
**FLOPPY.SCAVENGE** (Function) D-1  
**FLOPPY.TO.FILE** (Function) D-1  
**FLOPPY.UNARCHIVE** (Function) D-1  
**FLOPPY.WAIT.FOR.FLOPPY** (Function) D-1  
 Font directories C-3  
 Fonts 2  
     font conventions 3  
     Interpress 12,21  
 Frames, viewing 41,43  
 Functions  
     display and keyboard 27  
     environment inquiry 26  
     Lisp-stopping 25  
     login 26  
     system environment 24  
     timer and clock 27  
     VM 25

**G**  
**GETFILEINFO** (Function) 38

**H**  
 Hardware, requirements 1,24  
 Host access key 13  
 Host ID 11  
     identifying 24  
 Host name, identifying 24  
 Hosts supported by Medley  
     {CORE} 31  
     {DSK} 31  
     {LPT} 31  
     {NULL} 31  
     {UNIX} 31

**I**  
 Input/output devices, requirements 1  
 install.sunosX 1  
 Installation  
     preparation 7  
     script 9  
     software 9  
     tape layout C-1  
         extracting files from C-4  
 Installation Options Menu 9  
 Interlisp package 39,41  
 InterPress files 39  
 InterPress fonts, finding 12,21  
**INTERPRESSFONTDIRECTORIES** (Variable) 12,21

**K**  
 Kermit 38  
**KEYACTION** (Function) 15,17  
 Keyboard functions 27  
 Keyboard template  
     Sun 3 15,18  
     Sun 4 15,18  
 Keyboard tone generator 27  
**KEYBOARDEDITOR** D-2  
**KEYDOWNP** (Function) 15,17  
**kill** (UNIX Command) 42,44

**L**  
 lde 13,22, A-3  
     killing 42,44  
 ldechecksum (Command) 1, B-1  
**LDEDESTSYSOUT** (Variable) 23  
 ldeether 13, A-3  
**LDEFILETIMEOUT** (Variable) 40  
**LDEINIT** (Variable) 11,16  
**LDEKBDTYPE** (Variable) 16,19  
**LDEKBDTYPE** (Variable) 17  
**LDESRCESYSOUT** (Variable) 14,15  
 Left angle bracket 33,35  
 Library files, finding 12,21  
 Line feed 4,38,40  
 Lisp symbols  
     set in site initialization file 12,21  
**LISP-RELEASE-VERSION** (Variable) 24  
 LispUsers' Modules 5  
     finding 12,21  
**LISPUSESDIRECTORIES** (Variable) 12,21  
 litatom 39,41  
**LOGIN** (Function) 26  
 Login functions 26  
**LOGOUT** (Function) 22,23  
**\*LONG-SITE-NAME\*** (Variable) 12, 22  
**LONG-SITE-NAME** (Function) 12,22

**M**  
**MACHINE-INSTANCE** (Function) 24  
**MACHINE-TYPE** (Function) 24  
**MACHINE-VERSION** (Function) 24  
**MACHINETYPE** (Function) 24  
 Medley, exiting 23  
 Memory, requirements 1  
 MP errors 39,41  
**\MY.NSADDRESS** (Variable) 24

**N**  
 Naming onventions 32, 33  
 Network address, identifying 24  
 NFS service 40

**O**  
 Object file A-2  
**OPENFILE** (Function) 38  
**OPENSTREAM** (Function) 38  
 Operating system requirements 2  
 Options, adding 11

**P**  
 Packages 3  
 Passwords, maintaining for access 26  
 Pathname, Lisp 31,35  
 Period  
     single 32,37  
     double 32,37  
 Personal init file, set up 22  
**PLAYTUNE** (Function) 27  
 Postscript 1  
 Postscriptstream Module 5

Press files 39  
 Printers 1  
   default 12,21  
 PROTECTION (*File Attribute*) 38  
 Protocol  
   Ethernet 2  
   PUP 13  
 pstat (*UNIX Command*) 8

PUP protocol 13, 38  
   installation 3,8  
 PURGEDSKDIRECTORY (*Function*) D-1

**R**

\RCLKMILLISECOND (*Variable*) 28  
 READSYS 42,44  
 REALMEMORYSIZE (*Function*) 24  
 Relative pathnames 33  
 Release contents 4, 5  
 Relinking A-3  
 Right angle bracket 31  
 RINGBELLS (*Function*) 27  
 rmdir (*UNIX Command*) 37  
 root (*UNIX Command*) 13  
 RPC 5

**S**

SAVEVM (*Function*) 22,23  
 setenv (*UNIX Command*) 23  
 SETMAINTPANEL (*Function*) 27  
 SETPASSWORD (*Function*) 26  
 SETTIME (*Function*) 28  
**setuid** (*UNIX Command*) 26  
 SETUSERNAME (*Function*) 26  
 \*SHORT-SITE-NAME\* (*Variable*) 12, 22  
 SHORT-SITE-NAME (*Function*) 12,22  
 Site initialization file 11,15  
   how to find 12,21  
 site-init.lisp 11,16  
 SIZE (*File Attribute*) 38  
 Sketch files 39  
 Slash 31  
 Software requirements 2, A-2  
 Special characters 32  
 Square brackets 31  
 Stack 40,42  
 Sun type 3 keyboard 17,18  
 Sun type 4 keyboard 17, 19  
 Sun Workstations, sharing 2  
 SunOS  
   versions supported 2,37,38,39  
   Type 4 keyboard 16,19  
   console messages 28  
   directory notations 32  
   file system 31  
   username 26  
 SunOS process  
   identifying username of 26  
 SUSPEND-LISP (*Function*) 25  
 Swap space, allocating additional 8  
 Symbols  
   set in site initialization file 12,21  
 Sysout 2,13,39,41  
   files 37, 39  
   locations of 13,15  
 SYSOUT (*Function*) 24,45  
 System administrator 9,13

**T**

TCP D-1  
 TCPCHAT D-1  
 TCP/IP 38

- Teleraid 39,41
- Template  
  Sun 3 keyboard 15,18  
  Sun 4 keyboard 15,18
- Text files 4,38,40
- tilde 32
- tilde-slash 35,37
- TIME (Function) 28
- Timers 27
- /tmp/XXXX-lisp.log 29
- TYPE (File Attribute) 38,39
- U**
- {UNIX} 26,36,37,38  
  naming conventions 35
- {UNIX} (Function) 32
- UNIX process, suspending 25
- UNIX-FULLNAME (Function) 26
- UNIX-GETENV (Function) 26
- UNIX-GETPARG (Function) 26
- UNIX-USERNAME (Function) 26
- URAIID 39,41  
  commands 40,42  
  quit 40,42
- /usr/share/lde 1, 2, B-1
- User IDs, maintaining for access 26
- USERGREETFILES (Variable) 12,21
- USERNAME (Function) 26
- V**
- Version  
  identifying machine 24  
  numbering 33  
  numbers 32,35
- VIDEOCOLOR (Function) 27
- VIDEORATE (Function) 27
- VIRTUALKEYBOARD D-2
- Virtual memory  
  saving 23,24,25  
  saving with URAID 42,44
- VM functions 25
- VMEM.PURE.STATE (Variable) 25
- VMEMSIZE (Function) 23,25
- VOLUMES (Function) D-1
- VOLUMESIZE (Function) D-1
- VRAID 42,44
- W**
- WRITEDATE (File Attribute) 38
- X**
- X Windows 16  
  Medley window 17  
  preparing to run Medley 13,16  
  running Medley remotely 16  
  starting 17
- XNS Ethernet protocol 2, 8,13  
  enabling A-3
- [**
- [] 31
- \
- \BeginDST (Variable) 12,22
- \EndDST (Variable) 13,22
- \MY.NSADDRESS (Variable) 24
- \RCLKMILLISECOND (Variable) 28
- {
- {CORE} 31
- {DSK} 26,32,36,38  
  special characters 32  
  naming conventions 33
- {DSK}INIT. 12,16
- {LPT} 31
- {NULL} 31
- {UNIX} 26,36,37,38  
  naming conventions 35
- {UNIX} (Function) 32
- ~
- ~ 32
- ~/ 35,37
- ~/lisp.virtualmem 14,15,22,23
- \*
- \* 36
- \*LONG-SITE-NAME\* (Variable) 12,22
- \*SHORT-SITE-NAME\* (Variable) 12,22
- .
- . 32,37
- .. 32,37
- .cshrc file A-1;13
- .login file 13
- /
- / 31
- /install.sunosx A-3
- /usr/share/lde 1, 2, B-1
- <
- < 33,35
- >
- > 31