

TABLE of CONTENTS

Preface

1. Introduction

<u>The Medley Release</u>	1
<u>Differences Between Medley 1.0-S and Medley 1.1-S</u>	1
<u>System Overview</u>	2
<u>Useful SunOS and UNIX Conventions</u>	2
<u>Medley Device Conventions</u>	3
<u>Notation Conventions</u>	3

2. System Requirements

<u>Prerequisites</u>	5
<u>Processor Hardware</u>	5
<u>Memory</u>	5
<u>Swap Space</u>	5
<u>Disk Space</u>	5
<u>Input/Output Devices</u>	6
<u>Bitmap Display</u>	6
<u>Printers</u>	6
<u>Tape Access</u>	6
<u>Operating System Requirements</u>	6
<u>Constraints</u>	6
<u>Resource Constraints</u>	6
<u>Shared Sun Workstations</u>	6
<u>Release Contents</u>	7
<u>Documentation</u>	7
<u>Software</u>	7
<u>LispUsers' Modules</u>	9

3. Software Installation

<u>Insuring Adequate Swap Space</u>	11
<u>Installing Software</u>	11
<u>Copy Protection</u>	13
<u>Configuring the Software</u>	14
<u>Relinking</u>	14
<u>Enabling PUP/XNS Ethernet</u>	14
<u>Using/Installing the Host Access Key</u>	15

4. Getting Started	17
Site Initialization File	17
Starting Medley	19
Keyboard Interpretation	20
Sun Type 2 Keyboard	20
Sun Type 3 Keyboard	20
Sun Type 4 Keyboard	21
5. Using Medley on the Sun Workstation	23
Lisp Functions and Variables	23
System Environment Functions and Variables	23
Login Functions	24
UNIX Environment Inquiry	25
Virtual Memory Functions and Variables	25
Display and Keyboard Functions and Variables	27
Local Disk and Floppy Functions	28
Timers and Clocks	29
Miscellaneous Operational Differences	29
Console Messages	30
Library Modules Not Supported on the Sun	30
TCP, TCPCHAT, etc.	30
DLRS232C, DLTTY	30
KEYBOARDEDITOR, VIRTUALKEYBOARD	31
6. Medley For the Sun File Systems	33
Using SunOS Files from Medley	33
File Naming Conventions	33
Common {DSK} and {UNIX} Naming Conventions	34
{DSK} Naming Conventions	34
Version Numbering	35
Pathnames	36
{UNIX} Naming Conventions	37
Directories	37
Directory Enumeration	37
Directory Creation	38
Open File Limit	38
Default Pathname	38
File Attributes	39

File Variables	40
File System Errors	41
7. Compatibility Issues	43
Sysout Compatibility	43
SunOS Compatibility	43
File Compatibility	44
End-of-Line Convention	44
File Transfer Mechanisms	44
Kermit	44
TCP/IP	44
PUP	44
8. Error Recovery	45
URAID	45
Entering URAID	45
Conventions	46
URAID Commands	46
Displaying a Stack	47
Viewing Frames from a Stack	47
Miscellaneous	48
Other Fatal Error Conditions	48
System Error Conditions	49
Lisp Errors	50
Errors While Running Medley	50
I/O Errors	51
Virtual Memory Errors	51
Appendix A. UNIXCHAT	A-1
Appendix B. UNIXCOMM	B-1
Appendix C. Installation Hints	C-1
Appendix D. Checksum Control	D-1
Glossary	GLOSSARY-1
Index	INDEX-1

[This page intentionally left blank]

LIST of FIGURES

LIST of TABLES

Figure	Page
4-1 Sun Type 3 Key Numbering	21
4-2 Sun Type 3 Left Key Pad	21
4-3 Sun Type 3 Right Key Pad	21
4-4 Sun Type 3 Center Key Pad Interpretation	21
4-5 Sun Type 4 Key Numbering	22
4-6 Sun Type 4 Left Key Pad	22
4-7 Sun Type 4 Right Key Pad	22
4-8 Sun Type 4 Center Key Pad Interpretation	22

Table	Page
2-1 Font Directories	8
7-1 Medley Sysout Compatibility	43
7-2 SunOS Compatibility with Medley	43

[This page intentionally left blank]

This *Guide* describes Medley release 1.1-S for the Sun 3 and 4 workstations: 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 Xerox Lisp environment who want to use it on the Sun 3 or 4 workstations. 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* in order to correctly install the Medley 1.1-S 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, System Requirements; Chapter 3, Software Installation; and Chapter 4, Getting Started. These chapters guide the administrator through the process of installing Medley 1.1-S and configuring it on the Sun Workstation. Experienced Lisp users may want to configure the software; they should reference Chapter 4.

Users already familiar with the Lisp environment on Xerox workstations should find Chapter 1, Introduction, and Chapters 4 through 8, 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, gives an overview of Medley on the Sun Workstation, and introduces pertinent SunOS and UNIX conventions that are used throughout the *Guide*.

Chapter 2, System Requirements, describes the hardware, software, and documentation needed to run Medley on a Sun Workstation.

Chapter 3, Software Installation, contains the installation and software configuration procedures.

Chapter 4, Getting Started, explains how to set up a site initialization file. It also shows you the keyboard configuration and has instructions for getting started in Lisp on the Sun Workstation.

Chapter 5, Using Medley on the Sun Workstation, describes how specific Lisp functionality works on the Sun.

Chapter 6, *Medley For the Sun 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 7, *Compatibility Issues*, presents information on file transfer mechanisms and file compatibility.

Chapter 8, *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 you may encounter when running Medley on the Sun.

Appendix A, *UNIXChat*, has a description of the new *UNIXChat* library module for Medley on the Sun Workstation.

Appendix B, *UNIXComm*, describes the *UNIXComm* library module for Medley on the Sun Workstation.

Appendix C, *Installation Hints*, contains additional notes to help you configure Medley.

Appendix D, *Checksum Examples*, tells you how to validate the contents of the *tar* tape.

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

Medley is an Envos product which was built on the Xerox Lisp environment. It provides you with an integrated programming environment, supporting 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 your 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*

Envos Documentation

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

- Guy Steele, *Common Lisp, the Language*, 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 Envos 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 *Lisp Users' Modules*, Medley Release, which may be purchased separately, supplements the Medley release.

Other References

Refer to the following works when using the Medley Kermit library module with Medley on the Sun Workstation.

Da Cruz, Frank, *Kermit, A File Transfer Protocol*, Bedford, MA: Digital Press, 1987.

Da Cruz, Frank, ed., *Kermit User Guide*, New York: Columbia Center for Computing Activities, 1986.

Acknowledgements

Medley for the Sun Workstation is the result of a joint project, known as Maiko, undertaken by Fuji Xerox, Envos Corporation, and the Xerox Palo Alto Research Center. We hope that Interlisp-D and Xerox Common Lisp users find it a useful mechanism to allow them to make the transition between the Medley environment on Xerox 1100 series workstations and future products.

[This page intentionally left blank]

The Medley Release

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 you 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 is primarily intended for users with applications and development on the Xerox 1100 series workstation who wish to move their applications and development work to Sun Workstations.

On the Sun Workstation, the user has access to the SunOS operating system and a large number of utilities, programs, and programming languages. Medley on a Sun Workstation provides the following advantages over Xerox workstations:

- Hardware: larger and faster disk, faster processing, streaming tape, larger real memory.
Software: availability of UNIX, C, and other programming tools and environments.

Compared to other Lisp systems on the same workstation, Medley offers

- a complete, integrated programming environment,
- both Interlisp and Common Lisp,
- an integrated windowing system,
- integrated text and document processing tools,
- a large library of utilities, graphics packages, and applications,
- full support of Xerox PUP and XNS Ethernet protocols,
- efficient use of available memory.

Medley provides the Sun user with access to a large number of useful applications written in Common Lisp, Interlisp, and Interlisp-D.

Medley for the Sun Workstation has two versions, a Sun 3 version and a Sun 4 version, available on separate tapes. Medley 1.1-S runs on both Sun 3 and Sun 4 workstations.

Differences Between Medley 1.0-S and Medley 1.1-S

Medley 1.1-S is the successor to Medley 1.0-S, and completely supersedes that release. Medley 1.1-S offers the following major improvements over Medley 1.0-S.

- It runs on the Sun 4 workstation as well as on the Sun 3 workstation.

- Medley 1.1-S runs on SunOS 4.0 as well as SunOS 3.x. Installation is thus slightly different.
- On the Sun 3, Medley 1.1-S is up to 25% faster than Medley 1.0-S.
- On the Sun 3, under SunOS 4.0, Medley 1.1-S runs PUP and XNS protocols.
- It supports the Sun Type-4 keyboard in addition to the Type-3 keyboard.
- Medley 1.1-S automatically configures color displays in monochrome mode.

Medley 1.1-S fixes many Medley 1.0-S bugs in the sysout, emulator, library, and fonts. Note, however, that Medley 1.1-S sysouts **cannot run** on the Medley 1.0-S emulator.

System Overview

Functionally, the Medley system for the Sun Workstation consists of the following parts:

<i>emulator</i>	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.
<i>sysout</i>	A virtual memory image (the sysout) containing both byte-code compiled Lisp functions and data structures. The sysout provided can be used both on the Sun Workstation and on the Xerox 1100 series machines.
<i>library</i>	Files of compiled Lisp code and data structures.
<i>fonts</i>	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.

Useful SunOS and UNIX Conventions

SunOS is Sun Microsystems' version of the UNIX operating system. For the Xerox workstation user unfamiliar with a Sun Workstation, the following SunOS (and UNIX) conventions, which differ from corresponding Lisp conventions, are in use throughout the manual.

For complete information on UNIX and SunOS, refer to your Sun documentation set.

case, filenames	Type-in to UNIX is case sensitive. Typically, input is in lower case. When UNIX searches for a name, it is case sensitive; it distinguishes between lower and upper case characters. By convention, most names are lower case characters.
shell	Command interpreter (akin to the Medley Lisp Exec); the commands shown are in the C-Shell, unless otherwise noted.

Medley Device Conventions

Medley for the Sun Workstation allows users to interact with SunOS file systems (including file systems mounted from other machines) by using host device names. The device names are

- {**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 6, Medley For the Sun File Systems, 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	In this document, 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 .
	<i>Text in Optima italics</i> 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 the user is logged on as **root** or is running **su**; e.g.,
prompt #
 - a percentage sign (%), part of the system prompt, indicates that a user other than **root** is logged on; e.g.,
prompt %

[This page intentionally left blank]

This chapter outlines the hardware and system requirements for running Medley on a Sun Workstation. It also describes the contents of the software release and documentation provided.

Prerequisites

Processor Hardware

Medley runs on Sun 3 and Sun 4 workstations. It runs on both standalone workstations and diskless workstations linked to servers. NFS (Network File System) service is available via the host's file system transparently.

Medley on the Sun 3 workstation requires the MC68881 floating point coprocessor chip; all Sun 3 workstations are currently sold with this chip. On the Sun 4 workstation, the Weitek 1164/1165 coprocessor is optional.

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

Memory

You can expect reasonable interactive performance with 8 megabytes or more of RAM. Smaller configurations of diskless workstations have been tested, but performance suffers.

Swap Space

Of the required swap space, 45 megabytes must be available for Medley. Medley reserves this space at startup, but its requirement does not grow.

Disk Space

You need a minimum of 16 megabytes of disk file space for loading the software from tape, and an additional 1 megabyte of disk file space (on the file system where the installation is taking place) to install and configure the Lisp image. The 16 megabytes of disk space needed for the Lisp software can be broken down as follows:

Lisp sysout	5 Mb
Library modules	3 Mb
Fonts	7 Mb
Byte code emulator	2 Mb

Input/Output Devices

Medley provides you with 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 only runs on machines with either monochrome displays or color displays that can be operated in single-bit-per-pixel mode. Medley supports both the standard resolution display (1152 x 900) and the high-resolution display (1600 x 1280).

Printers

For hardcopy output, Medley, via PUP/XNS, currently requires that you print to an Interpress or Press printer. The FX80 printer also works via the RS232 port.

If you have a Postscript printer, you can use the LispUsers' module PostScript to direct output to your printer.

Tape Access

For installation you need either a $\frac{1}{4}$ -inch cassette tape cartridge or a $\frac{1}{2}$ -inch 9-track tape drive, located locally or on a remote machine.

Operating System Requirements

Medley on the Sun 3 workstation requires SunOS operating system versions 3.2, 3.4, 3.5, or 4.0. On the Sun 4 workstation, Medley requires SunOS version 4.0.

Note: Medley XNS Ethernet code cannot be run simultaneously with SunOS 3.5 Kernel XNS Ethernet code.

Constraints

Resource Constraints

When Medley is running, it takes over the entire display screen. Other window systems such as **sunttools** are unavailable.

Shared Sun Workstations

Medley runs its own process scheduler; the Lisp process is always running as far as the UNIX scheduler is concerned. 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 environments.

Similarly, Medley may not have adequate interactive performance if it is competing with other computationally-bound processes on the same Sun Workstation.

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

Release Contents

The release distribution contains the following documentation and software.

Documentation

The Envos 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 receive, in addition to the manuals listed above,

- *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*
- All users can also purchase the document
- *LispUsers' Modules, Medley Release*

Software

The software release is available on either a $\frac{1}{2}$ -inch tape cartridge or a $\frac{1}{2}$ -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 following directories:

./checksumdir

Contains **ldechecksum**, **checksum**, and **X.sum** checksum files

./install.sunos3

Contains **makefile**, **lde.o**, **ldeether.c**, **lde**, **ldeether**, **usersubrs.c** for SunOS 3.x.

./install.sunos4

Contains **makefile**, **lde.o**, **ldeether.c**, **lde**, **ldeether**, **usersubrs.c** for SunOS 4.x.

./lispsysouts

Contains the **sysout**, **lisp.sysout**

<code>./lisplibrary</code>	Contains all Medley 1.1-S Lisp library files
<code>./lisppatches</code>	Contains patch files, if any, for Lisp code.
<code>./fonts</code>	Contains the font directories.

Table 2-1 shows the organization of the font directories, as well as the descriptions and contents of the directories.

Table 2-1. Font Directories

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

If you do not use some of these files, you may choose to delete them after installation. Alternately, you might selectively **tar** off only those files/directories of interest (see your UNIX documentation). For example, most sites might not use Chinese fonts, or Interpress/Press printers, so the directories ./display/chinese, ./interpress, and ./press would be candidates for deletion.

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; Envos 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 PostScript printers for output, a particularly useful LispUsers' module is PostScript. The LispUsers' module RPC implements Sun remote procedure calls.

[This page intentionally left blank]

This chapter contains the Medley installation instructions to be used by the system administrator for Sun Workstations.

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

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

Insuring Adequate Swap Space

Prior to installing the software, you should insure that your system has adequate swap space. Medley requires 45 megabytes of swap space on top of the normal swap space requirements. You can 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.

Once you have checked your swap space you can

- load the contents of the release tape;
- configure Lisp for your system.

In the following procedures, the acronym **lde** is frequently a part of a name. It stands for Lisp development environment.

Installing Software

The installation procedure shows the steps required, with examples where appropriate, to install the Medley software. Only those users who are system administrators and have **root** privileges on the host/server where the software is to reside can use this procedure.

Before starting software installation, remember that

- the software requires 16 megabytes of disk space;
- the entire installation procedure takes about 25 minutes to complete on a standalone system with a $\frac{1}{4}$ -inch tape.

1. Log in under your username.

```
login yourname  
prompt#
```

2. Check for available space with the **df** (1) command:

```
prompt# df
```

Filesystem	kbytes	used	avail	capacity	Mounted on
/dev/xy0a	7437	5470	1223	82%	/
/dev/xy0h	148455	4900	128709	4%	/usr/misc

3. Determine if you need to run **su** to make a directory for the distribution. If so, type in **su**:

```
prompt# su
```

4. Make a directory for the distribution. If you previously installed Medley 1.0-S, you might save or move the old directory to another location. (Since Medley 1.1-S supercedes 1.0-S, you should eventually delete it.) Let's assume that there is not currently a /usr/local/lde directory. If you have enough space on the file system containing /usr/local, then create an lde directory:

```
prompt# mkdir /usr/local/lde
```

If you don't have enough space on /usr/local, go to step 6.

5. Make yourself the owner of this directory:

```
prompt# /etc/chown yourname /usr/local/lde
```

where **yourname** is your username. Go to step 7.

6. If you don't have space on the file system which contains /usr/local, but do have space somewhere else, for instance on /usr1, then make the directory there and link /usr/local/lde to it:

```
prompt# mkdir /usr1/lde  
prompt# /etc/chown yourname /usr/usrl/lde  
prompt# ln -s /usrl/lde /usr/local/lde
```

7. If you ran **su**, type in **exit**:

```
prompt# exit
```

8. Mount the 9-track tape on the drive or insert the tape cartridge in the cartridge drive.

9. Change the working directory to /usr/local/lde:

```
prompt# cd /usr/local/lde
```

10. Load the Medley software from tape. Indicate the appropriate device abbreviation for your tape drive by replacing **xx** in the example below with

ar for the Archive drive,

st for a SCSI tape drive,

mt for the Tapemaster $\frac{1}{2}$ -inch (1600 bpi).

This example shows the command entry sequence:

```
prompt# tar xvfb /dev/rxx0 126
```

As the software is copied from tape (a process that takes a long time) the system prints a series of lines in the following form:

```
x ./lisplibrary/GRAPHER.LCOM, 57923 bytes, 114 tape blocks
```

The **x** at the beginning of the line indicates that the file is being extracted from the tape. See Appendix D, Checksum Control, for information on validating the contents of the tape.

Copy Protection

To use Medley on your host machine, you must first obtain a host access key from Envos. This key enables the use of Medley on one host. The software cannot run without the host access key. You must have one host access key for each host on which you wish to run. Note that your current host access key will work if you move from SunOS 3.4 to 4.0 in Medley 1.0-S, or if you upgrade from Medley 1.0-S to Medley 1.1-S.

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 Envos at

800-228-5325 (United States)

800-824-6449 (California)

between 7:30 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 Envos representative your host ID. Envos provides you with a host access key, which you need during software configuration, below.

CAUTION

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

Configuring the Software

The software comes in two forms:

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

If you want to use the executable that Envos supplies, skip to the section, Enabling PUP/XNS Ethernet.

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/local/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/local/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/local/lde/install.sunos4  
server# /etc/chown root ldeether  
server# chmod 4755 ldeether  
server# exit
```

If you are using the Ethernet, use 'ldeether' whenever 'lde' appears in the instructions below.

Using/Installing the Host Access Key

When you want to run Medley, you can enter the host access key on the command line:

```
prompt% lde sysoutname -k 'host-key'
```

In this example, replace *host-key* with the actual key (still enclosed in single quotes). Replace *sysoutname* with the name of the sysout you want to use.

You can also avoid having to type the host access key each time. One way is to define an alias in the .cshrc file.

```
alias runlde "lde sysout-fullpathname -k 'host-key'"
```

Another method is via a shell script.

```
prompt% echo "lde sysout-fullpathname -k 'host-key'" >
.. /usr/local/lde/install.sunos4/runlde
prompt% chmod 755 /usr/local/lde/install.sunos4/runlde
prompt% rehash
```

Remember to replace *sysout-fullpathname* with the full name of the sysout you want to use. Then, when you wish to run, you can simply type the alias or the script name.

```
prompt% runlde
```

Finally, you should arrange for *lde* and *ldeether* to be conveniently located on users' search paths. For example:

```
prompt% ln -s /usr/local/lde/install.sunos3/lde /usr/bin/lde
prompt% ln -s /usr/local/lde/install.sunos3/ldeether
/usr/bin/ldeether
```

See Appendix C, Installation Hints, for further discussion on setting up installation.

[This page intentionally left blank]

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.

Site Initialization File

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

Greeting and initialization are described in the *Interlisp-D Reference Manual*, section 12.1. The following paragraphs briefly explain some differences in the behavior of the initialization file when Medley runs under SunOS.

When running under SunOS, Lisp first 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/local/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/local/lde/site-init.lisp**. The distribution tape contains a sample site initialization file in the Lisp library directory **/usr/local/lde/lisplib/INIT.MAIKO**. The system administrator should copy **INIT.MAIKO** into **/usr/local/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 6, Medley For the Sun File Systems, describes the **{DSK}** device.

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

IL:USERGREETFILES

[Variable]

Returns a list of templates to search for the place where individuals should find their personal initialization files. If not set in the site initialization file, no initialization 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]

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

IL:INTERPRESSFONTDIRECTORIES

[Variable]

Returns the directory to search for Interpress font widths.

XCL:*LONG-SITE-NAME*

[Variable]

Returns the value of the function **XCL:LONG-SITE-NAME**, for example, "Frobnitz, Baz and Lispers, Incorporated."

XCL:*SHORT-SITE-NAME*

[Variable]

Returns the value of the Common Lisp function **XCL:SHORT-SITE-NAME**, for example, "Frobco".

IL:DIRECTORIES

[Variable]

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

IL:LISPUSERSDIRECTORIES

[Variable]

Returns 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]

Returns a list of names of default printers.

IL:DEFAULTPRINTERTYPE

[Variable]

Returns the default printer type, for example, Interpress.

\BeginDST

[Variable]

Returns 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. (Note: This number is subject to future Congressional legislation.) If you are in a region where Daylight Savings Time is not observed, the value is 367.

\EndDST

[Variable]

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

Starting Medley

Running Medley requires both an executable (**lde**) and a virtual memory image (a sysout). Generally, Medley requires a host access key matched to the host ID of the processor on which it is running. In addition, if Medley is to perform direct network communication (for Xerox PUP or XNS protocols), it needs a separate program, **ldeether**, to initialize the Ethernet. To obtain Ethernet packets, **ldeether** must run as **root**. To do this, the owner of the file **ldeether** must be **root**. The system administrator must install it with **root** as owner and the "set user id" bit set.

To run Medley on the Sun Workstation, follow these steps.

1. Exit **suntools** or any other windowing system. 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.

Note: 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/local/lde/install.sunosx)
```

You can also add this to your .login file or .cshrc file.

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

```
prompt# lde optional-sysout -k 'this host-key'
```

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

```
prompt# ldeether optional-sysout -k 'this host-key'
```

This, in turn, runs **lde** and allows it to use the Ethernet 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; e.g.,

```
prompt# lde sysout -k 'this host-key'
```

- **LDESRCESYSOUT**

If no sysout file name is given on the command line, the value of the environment variable **LDESRCSYSOUT** is used as the name of the sysout file.

- `~/lisp.virtualmem`

Finally, Medley looks for a file `lisp.virtualmem` on the home directory of the current user.

Once Medley has started, it allocates virtual memory space to hold the working environment, maps the screen frame buffer into its address space, and reads the sysout. As pages from the sysout are read into the display area, they appear on the screen. If the sysout was written while running on a machine with a different size display, the image will appear garbled. This process takes several seconds. After Lisp starts running, it readjusts the display to the current size.

Keyboard Interpretation

This section describes how Medley interprets the Sun Type 2, Type 3, and Type 4 keyboards. 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 **IL:KEYDOWNP** and **IL:KEYACTION**, for example.

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 2 Keyboard

The Medley key assignments for the Sun Type 2 keyboard resemble those for the Sun Type 3 keyboard:

- The R15 key assignment (on the right key pad) is CAPS LOCK.
- There is no NEXT key; the BREAK key functions as a NEXT key.
- The BACK SPACE key assignment is DELETE WORD.
- The DELETE key assignment is BACK SPACE.
- The RIGHT key assignment is EXPAND.

Sun Type 3 Keyboard

Figure 4-1 shows the key number assignments for the Sun Type 3 keyboard.

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
14	62	34	19	18	3	48	49	51	6	23	25	11	58	29
111	89	36	21	20	5	35	50	52	38	9	26	43	28	44
90	46	41	40	24	37	7	39	54	55	27	42	12	60	71
		56	31				57					93	47	

Figure 4-1. Sun Type 3 Key Numbering

Figures 4-2 through 4-4 show Medley's key assignments for the Sun Type 3 keypads.

Stop	Again
Help	Undo
Same	Move
Open	Copy
Find	Delete

Figure 4-2. Sun Type 3 Left Key Pad

Num Lock	Scroll Lock	Break
7 Home	8 ↑	9 PgUP
4 ←	5	6 →
1 End	2 ↓	3 PgDN
Ins	DOIT	Caps Lock

Figure 4-3. Sun Type 3 Right Key Pad

Center	Bold	Italic		Case		Strikeout		Underline		Super Sub		Larger Smaller	Margin	Back Word
Esc	!	@	#	\$	%	6	&	*	()	-	=	{	}
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 4-4. Sun Type 3 Center Key Pad Interpretation

Sun Type 4 Keyboard

Figure 4-5 illustrates the keyboard interpretation for the Sun Type 4 keyboard.

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

Figure 4-5. Sun Type 4 Key Numbering

Figures 4-6 through 4-8 show the keyboard and the left and right key pads for the Sun Type 4 keyboard.

Stop	Again
Props	Undo
Same	Copy
Open	Move
Find	Delete
Help	

Figure 4-6. Sun Type 4
Left Key Pad

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

Figure 4-7. 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	!	@	#	\$	%	↑	&	*	()	-	+	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 4-8. Sun Type 4 Center Key Pad Interpretation

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

[This page intentionally left blank]

5. USING MEDLEY ON THE SUN WORKSTATION

The operation of the Medley environment on the Sun Workstation is quite similar to that on Xerox Lisp workstations. This chapter describes the operational differences. Readers of this section should be acquainted with the Xerox Lisp environment.

Lisp Functions and Variables

On the Sun Workstation, the following functions behave differently from the way they operate on a Xerox workstation.

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]

Ordinarily 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 virtual machine (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 1.1, **IL:LISP-RELEASE-VERSION** is 1.1. While **IL:MAKESYSNAME** doesn't change, **IL:LISP-RELEASE-VERSION** always changes with each new

sysout release. This variable does 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.)

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 6, Medley For the Sun 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

- the user is not running under UNIX
- the user does not have a full name entered in /etc/passwd, or
- 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.

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

UNIX 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.

<u>(IL:UNIX-GETENV STRING)</u>	[Function]
Returns the value of the environment variable with the given name. The argument <i>STRING</i> should be the name of a UNIX environment variable. For example, (UNIX-GETENV "HOME") might return the user's home directory.	
<u>(IL:UNIX-GETPARM STRING)</u>	[Function]
Returns the value of one of a few built-in parameters. The argument <i>STRING</i> should be the name of one of the following UNIX environment variables: "MACH" returns the machine type, "ARCH", the general machine architecture, "HOSTNAME", the local host name, and "HOSTID", the Sun host identification number as a hexadecimal string.	

Virtual Memory Functions and Variables

These features deal with saving a Medley sysout, leaving Medley, and restarting. Before using them on the Sun Workstation, you should be aware of differences in how virtual memory is handled by Medley on the Sun Workstation and Xerox 1100 series workstations.

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.

On the Sun, **1de** is an ordinary UNIX program that allocates a 45-megabyte 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

In Medley, a "page" is 512 bytes. Under SunOS, the page size is variable; some Sun Workstations use 8K byte 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: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:LOGOUT FAST)**[Function]**

Allows you to 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. You can 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.

On Xerox workstations, when calling **LOGOUT** with *FAST* the symbol **IL:?** would conditionally save virtual memory. On the Sun Workstation, (**LOGOUT IL:?**) operates like (**LOGOUT T**).

(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).

(IL:SAVEVM --)**[Function]**

Operates similarly to (**LOGOUT**) 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 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 6, Medley For the Sun File Systems, for further information on {**DSK**} and {**UNIX**}.) **SYSOUT** can cause the error

File-System-Resources-Exceeded.

(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**.

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:CHANGEBACKGROUND BORDER

IL:VIDEORATE

IL:SETMAINTPANEL

IL:VIDEOCOLOR

The Sun Workstation has only a monotone tone generator in the keyboard; thus, 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.

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:VOLUME SIZE

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

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:

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]

Returns 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 Mbytes), 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; 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. There are two points to watch for:

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.

The process **\10MBWATCHER**, used to watch for Ethernet packets, reads console messages. Thus, you should never kill the **\10MBWATCHER** process, even if you don't use the Ethernet capabilities of Medley. If you do kill the **\10MBWATCHER** process, console messages cannot be printed and the operating system can hang.

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]

6. MEDLEY FOR THE SUN FILE SYSTEMS

This chapter discusses the file conventions used when you run Medley on the Sun Workstation.

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 allows you to 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 lower case letters as distinct from their upper case 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.

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. Note that square brackets ([])) are not acceptable as directory delimiters.

Common {DSK} and {UNIX} Naming Conventions

- The single quote ('') is not supported.
- The following special characters cannot be handled on {DSK}: backslash (\), slash (/), right angle bracket (>), semicolon (;) and tilde (~). Thus, from {DSK} you cannot name files containing these characters. Special characters in a file name cannot be escaped by a single quote. The following UNIX file names are illegal in Medley:
`foo/fee`
`foo>fee`
`foo;3`
`foo;`
- {DSK} cannot distinguish between a file name with a period (e.g., foo.) and a simple file name (e.g., foo). For {DSK}, each is version 1 of the file (e.g., {DSK}foo.;1).
- On {DSK} the C-Shell and SunOS directory notations (~, ., and ..) are only allowed in the Lisp directory specifications at the very beginning of the directory specification of a pathname. The tilde character (~) corresponds to the user's home directory at login. The period (.) corresponds to the current working directory, while two periods (..) indicates the parent of the current working directory.

{DSK} Naming Conventions

File access to the {DSK} device goes through 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} does not accept file names ending in tilde
- {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).

However, a combination of relative path specifiers (', ..) is not supported. For instance, {DSK} cannot interpret {DSK}~'../anotherone/foo.c. In addition, {DSK} does not support the tilde-name (~name) convention (e.g., {DSK} cannot interpret {DSK}~tom/foo.c). {DSK} also does not accept file names ending in tilde; a {DSK} file name ending in tilde causes an error message to appear in the prompt window.

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 12th version of myfile.

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

{DSK} Name From Lisp	File Name From SunOS
bar.baz;1	bar.baz.~1~
bar.baz;2	bar.baz.~2~
bar.;23	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}	From SunOS
foo.c;15	foo.c (hard linked with foo.c.~23~)
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}	From SunOS
foo.c;15	foo.c (hard linked with foo.c.~24~)
foo.c;22	foo.c.~15~
foo.c;24 (new file)	foo.c.~23~ (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}	From SunOS
foo.c;1	foo.c (linked to foo.c.^2^)
foo.c;2	foo.c.^1^
	foo.c.^2^

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

From {DSK}	From SunOS
foo.c;1	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}	From SunOS
foo.c;1	foo.c (linked to foo.c.^2^)
foo.c;2	foo.c.^1^
fee.c;1	foo.c.^2^
fee.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}	From SunOS
foo.c;1	foo.c (linked to foo.c.^1^)
fee.c;1	foo.c.^1^
fee.c;2	fee.c (linked to fee.c.^3^)
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^**), that file is inaccessible using {DSK}.

From {DSK}	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. Lisp functions such as **FINDFILE** and **DIRECTORY** return pathnames with the original case. If **IL:*UPPER-CASE-FILE-NAMES*** is NIL, **DIRECTORY** returns pathnames with the original case. When file names are given for recognition, the system searches for the highest version number and either matches or not according to the file recognition mode (**OLD**, **NEW**, **OLDEST**, **OLD/NEW**).

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 **CWDIR** 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 lower case letters are treated as distinct.

Examples:

{UNIX} Name From Lisp	File Name From SunOS
<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. Note 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/envos/*)
>({DSK}/users/envos/foo.c;1
{DSK}/users/envos/foo.c;2)
```

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

```
(DIRECTORY '{UNIX}/users/envos/*)
>({UNIX}/users/envos/foo.c
{UNIX}/users/envos/foo.c.^1^
{UNIX}/users/envos/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.

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. Note that 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.

<u>(CHDIR PATHNAME)</u>	<u>[Function]</u>
	<p>Changes the current working directory for the current invocation of Lisp. For example,</p> <pre>(CHDIR "{DSK}~/subdir/") (OPENSTREAM "{DSK}./foo" ...)</pre> <p>opens the SunOS file ~/subdir/foo.</p> <p>Only the directory field of <i>PATHNAME</i> is used for the current directory. If a there is a name field, the name is ignored.</p> <pre>(CHDIR '{DSK}/users smith/test/foe) > "{DSK}/users smith/test/" (CHDIR '{UNIX}temp/foo) > "{UNIX}/users smith/temp/"</pre> <p>Thus, to specify a directory the pathname should end with a slash (/).</p> <p>The error message No-Such-Directory appears following the system echo of the pathname if the <i>PATHNAME</i> does not exist.</p>

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.

<u>WRITEDATE and CREATIONDATE</u>	<u>[File Attributes]</u>
	<p>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.</p>

TYPE	[File Attribute]
<p>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 DEFAULTFILETYP LIST. 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.</p>	
EOL	[File Attribute]
<p>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 DEFAULTFILETYP LIST variables). If the TYPE property of file is TEXT, LF(=10) is used as EOL. If the TYPE property of a file is BINARY, CR(=13) is used as EOL.</p> <p>Note: EOL conventions on {DSK} are incompatible with those on Xerox workstations.</p>	
AUTHOR	[File Attribute]
<p>Returns the author of the file, i.e., the login name of the user that created it. This attribute cannot be changed.</p>	
PROTECTION	[File Attribute]
<p>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.</p>	
SIZE	[File Attribute]
<p>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).</p> <p>Note: SETFILEINFO allows you to change the SIZE attribute of I/O streams and output streams but a file cannot be expanded this way.</p>	

File Variables

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

DEFAULTFILETYPE	[Variable]
<p>Initially set to TEXT. Used with the file attribute TYPE.</p>	

DEFAULTFILETYPELIST

[Variable]

A list of accepted file types. Should be set to ((C . TEXT) (H . TEXT) (O . BINARY) (OUT . BINARY) (LCOM . BINARY) (DFASL . BINARY) (SKETCH . BINARY) (IP . BINARY))

Used with the file attribute **TYPE**. Binary files, such as Sketch files, Interpress files, or Press files, should have their extensions registered in **DEFAULTFILETYPELIST**. 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 the NFS service is down, any attempt to access a file on the file system eventually results in a timeout error. The following error message is printed in the prompt window:

File access timed out

If the mounted device is mounted with the "hard" option, Medley continues to wait until the mounted device responds. During that time, user interrupts are not available. We recommend mounting remote file systems with the "soft" option. You can use the UNIX command **/etc/mount** to check the current mount options.

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**

Another type of error occurs occasionally when the file system prints an incorrect message such as

File not found

A more accurate console message appears, at the same time, in the prompt window. This message appears in the SunOS message form

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]

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

Table 7-1 shows sysout compatibility for the Medley releases. Note that the Medley 1.1-S sysout is not compatible with the Medley 1.0-S sysout.

Table 7-1. Medley Sysout Compatibility

Medley Release	Sysout Machine	Medley 1.0-S Sun-3	Medley 1.1-S	
		< = >	Sun-3	Sun-4
1.0-S	Sun-3	< = >	N	N
1.1-S	Sun-3	N	< = >	< = >
1.1-S	Sun-4	N	< = >	< = >
		< = >	Sysout can be moved in either direction.	
		= > or < =	Sysout can be moved in this direction only.	
		N	Sysout cannot be moved between these configurations.	

SunOS Compatibility

Table 7-2 indicates SunOS and Sun hardware compatibility for the Medley releases.

Table 7-2. SunOS Compatibility with Medley

Medley Version	SunOS Version			
	3.2	3.4	3.5	4.0
Medley 1.0-S	Sun-3	Sun-3	Sun-3	+
Medley 1.1-S	Sun-3	Sun-3	Sun-3	Sun-3,4

† Not available for the Sun 4. On the Sun 3, Lisp cannot directly access low level Ethernet capabilities. As a result, you cannot access NS and PUP file servers or print to NS or PUP-based printers. You can still access Sun servers and other hosts; e.g., you can still use FILEBROWSER to transfer files between Sun hosts and you can also continue to use UNIXChat to the UNIX shell as a means of accessing remote hosts that use NFS and/or TCP/IP.

File Compatibility

You can develop, run, and load systems on the Xerox 1100 series workstation in the Medley release and move the total application to the Sun Workstation. However, the sysout supplied on the release tape and loaded on the Sun Workstation must also be compatible with the Xerox 1100 series workstations.

Note that sysout files created on the Sun Workstation cannot be subsequently moved back to Xerox workstations.

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 most Xerox systems have 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.

File Transfer Mechanisms

You can move files from Xerox workstations to the Sun Workstation using Kermit, TCP/IP, or PUP.

Kermit

The Lisp library contains a Kermit file transfer mechanism. It is possible to connect the Xerox serial port directly to the serial port of the Sun Workstation and use Kermit if you have it on the Sun Workstation. Refer to *Kermit, A File Transfer Protocol* and *Kermit User Guide* by Frank Da Cruz when using this library module.

TCP/IP

The Koto, Lyric, and Medley releases of Lisp for the Xerox 1100 series workstations can directly talk TCP to SunOS.

PUP

If Medley is running on a Sun with the Ethernet enabled, FTPServer can also be used to transfer files.

[This page intentionally left blank]

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

- the Xerox Lisp error system,
- 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 Teieraid 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 later return to Lisp.
- 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

URAID uses these display conventions:

- Numbers are displayed in hexadecimal unless otherwise noted.
- The *litatom* should be an upper case 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.

URAID Commands

URAID 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 | Exit 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 **I** 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 I command to be <i>type</i> , which is either a or c . The default is to trace ALinks. ALinks follow the chain of free variable access. <i>type := a(Alink) c(Clink)</i>
I [type]	Shows the stack as a back trace consisting of a numbered sequence of frame names. The default is the user stack. The argument <i>type</i> 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 <i>type</i> are k (keyboard handler) r (reset) m (miscellaneous) g (garbage collect) p (page fault). <i>type := k m r g p nil</i>

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 <i>number</i> (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 message Press Return (To quit ESC and RET) appears as you display a frame with the f command. 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 <i>n</i> is the number of the frame most recently viewed. Immediately after an I command, <i>n</i> is zero, so CR views the first frame.
a /litatom	Displays the top-level value of the <i>litatom</i>
d /litatom	Displays the contents of definition cell for the <i>litatom</i> . 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 <i>func1 func2</i>	Moves the definition of <i>func1</i> to <i>func2</i> .
t <i>Xaddress</i>	Displays the type of this object.
p <i>litatom</i>	Displays the contents of the <i>litatom</i> 's property list.
w	Displays the current function name and PC.
x <i>Xaddress [Xnum]</i>	Prints <i>Xnum</i> word (16-bits) of the raw contents of the virtual memory starting at virtual address <i>Xaddress</i> . 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 <i>litatom</i> to the specified value. <i>snumber</i> is a signed smallp number.
< Xaddress val	Sets the the contents of the word(16-bits) at the <i>Xaddress</i> to <i>val</i> .

Miscellaneous

v <i>filename</i>	Saves the current virtual memory on the <i>filename</i> . This file can be examined using the functions READSYS and VRAID in the TeleRaid Lisp Library module.
	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.

System Error Conditions

The following are error messages generated by SunOS. For complete information on these error messages, see the *SunOS Reference Manual*, Intro(2).

ERROR MESSAGE	DESCRIPTION
0 Unused	
1 EPERM	Not owner
2 ENOENT	No such file or directory
3 ESRCH	No such process
4 EINTR	Interrupted system call
5 EIO	I/O error
6 ENXIO	No such device or address
7 E2BIG	Arg list too long
8 ENOEXEC	Exec format error
9 EBADF	Bad file number
10 ECHILD	No children
11 EAGAIN	No more processes
12 ENOMEM	Not enough core
13 EACCES	Permission denied
14EFAULT	Bad address
15 ENOTBLK	Block device required
16 EBUSY	Mount device busy
17 EEXIST	File exists
18 EXDEV	Cross-device link
19 ENODEV	No such device
20 ENODIR	Not a directory
21 EISDIR	Is a directory
22 EINVAL	Invalid argument
23 ENFILE	File table overflow
24 EMFILE	Too many open files
25 ENOTTY	Not a typewriter
26 Unused	
27 EFBIG	File too large
28 ENOSPC	No space left on device
29 ESPIPE	Illegal seek
30 EROFS	Read-only file system
31 EMLINK	Too many links
32 EPIPE	Broken pipe
33 EDOM	Math argument
34 ERANGE	Result too large
35 EWOULDBLOCK	Operation would block
36 EINPROGRESS	Operation now in progress
37 EALREADY	Operation already in progress
38 ENOTSOCK	Socket operation on non-socket
39 EDESTADDRREQ	Destination address required
40 EMSGSIZE	Message too long
41 EPROTOTYPE	Protocol wrong type for socket
42 ENOPROTOOPT	Bad protocol option
43 EPROTONOSUPPORT	Protocol not supported
44 ESOCKTNOSUPPORT	Socket not supported
45 EOPNOTSUPP	Operation not supported on socket
46 EPFNOSUPPORT	Protocol family not supported
47 EAFNOSUPPORT	Address family not supported by protocol family

ERROR MESSAGE		DESCRIPTION
:48	EADDRINUSE	Address already in use
:49	EADDRNOTAVAIL	Can't assign requested address
:50	ENETDOWN	Network is down
:51	ENETUNREACH	Network is unreadable
:52	ENETRESET	Network dropped connection on reset
:53	ECONNABORTED	Software caused connection abort
:54	ECONNRESET	Connection reset by peer
:55	ENOBUFS	No buffer space available
:56	EISCONN	Socket is already connected
:57	ENOTCONN	Socket is not connected
:58	ESHUTDOWN	Can't send after socket shutdown
:59	Unused	
:60	ETIMEDOUT	Connection timed out
:61	ECONNREFUSED	Connection refused
:62	ELOOP	Too many levels of symbolic link
:63	ENAMETOOLONG	File name is too long
:64	EHOSTDOWN	Host is down
:65	EHOSTUNREACH	No route to host
:66	ENOTEMPTY	Directory not empty
:67	Unused	
:68	Unused	
:69	EDQUOT	Disc quota exceeded
:70	ESTALE	Stale NFS file handle
:71	EREMOTE	Too many levels of remote in path
:72	ENOSTR	Not a stream device
:73	ETIME	Timer expired
:74	ENOSR	Out of stream resources
:75	ENOMSG	No message of desired type
:76	EBADMSG	Not a data message
:77	EIDRM	Identifier removed

LISP Errors

Lisp error handling operates much as it does on a Xerox workstation.

Normal Lisp errors are handled in the same way as they were in the Lyric Lisp environment.

Errors While Running Medley

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

ERROR MESSAGE	REASON/FUNCTION RESPONSIBLE
File access timed out	Occurs when user tries to access a file when the remotely mounted file system or NFS service is down.
File too large	Self explanatory.

Too-Many-Files-Open	Occurs when 1) user exceeds the SunOS open file limit (see Chapter 6, Medley For the Sun File Systems) 2) user exceeds system file resources while writing a sysout (using IL:SYSOUT)
Nonexistent directory	Occurs when user tries to connect to a nonexistent directory using IL:CNDIR or the CONN command.
No-Such-Directory	CHDIR
Connection timed out	Self explanatory.
Bad Host Name	Self explanatory.

I/O Errors

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

ERROR MESSAGE	REASON/FUNCTION RESPONSIBLE
Floppy: No floppy drive on this machine.	Self-explanatory.
Device error: {FLOPPY}	Occurs when 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 when running in SunOS.

Virtual Memory Errors

ERROR MESSAGE	REASON/FUNCTION RESPONSIBLE
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]

The UNIXChat library module is similar to Chat, but communicates with a C-Shell on your own host rather than with another machine.

Requirements

UNIXChat depends on Chat and UNIXComm.

Installation

Load UNIXCHAT.LCOM from the library. We recommend that you also load VTCHAT.LCOM from the library.

User Interface

Use the following procedure to open a Chat window and activate the terminal emulator.

1. Invoke Chat in one of the following ways:

a. Choose CHAT from the background menu and type **SHELL** at the *Host:* prompt in the Prompt Window.

b. Call the Chat function:

(CHAT 'SHELL LOGOPTION INITSTREAM WINDOW)

[Function]

See the manual *Lisp Library Modules*, Medley Release, for the definition of the Chat function. Use the variable CHAT.DISPLAYTYPES, if necessary.

You should now be talking to a UNIX C-Shell in the Chat window.

2. Set your terminal type:

prompt% setenv TERM vt100

NOTE: Alternately, shells started from UNIXChat have the shell variable LDESHELL set. Thus, in your .cshrc file you could have

```
if ($?LDESHELL ==1) then
    setenv TERM vt100
    setenv erase ^H
endif
```

This makes your Chat C-Shells easier to use.

3. When you've completed a Chat session, close the connection by using CLOSE in the right button window menu or by typing **exit** to the C-Shell.

Note: You can have several Chat windows open to **SHELL** at the same time.

[This page intentionally left blank]

UNIXComm starts up a UNIX process on a Sun Workstation. UNIXComm, with two functions, provides the user with a stream interface to the SunOS Bourne shell.

Installation

Load UNIXCOMM.LCOM from the library.

User Interface

Two functions allow you to open and close SunOS subprocess streams.

(IL:CREATE-PROCESS-STREAM STRING)

[Function]

Interfaces to the SunOS **system** function (see Chapter 3 of the *SunOS Reference Manual*). This causes a subprocess running the Bourne shell to be spawned; *STRING* is passed as a command to it. A bidirectional stream is returned with input reading data which the process writes to its standard output (**stdout**) and standard error (**stderr**). Output writes data which the process can read from its standard input (**stdin**). EOFP does not return a meaningful value for those streams; instead, you must change the default EOF mechanism for the end of stream.

(IL:UNIX-STREAM-CLOSE STREAM)

[Function]

Returns the numeric status value of the process created by **IL:CREATE-PROCESS-STREAM** if it has exited; otherwise, kills the process. **IL:CLOSE** can be used instead of **IL:UNIX-STREAM-CLOSE** if the process status value is not of interest.

[This page intentionally left blank]

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/local/lde/lispssouts/ROOMS.SYSOUT
setenv LDEINIT      /usr/local/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 1.1, the only differences between the Sun 3 and Sun 4 distributions are in the install.sunosX directories. Thus, during installation the common subdirectories (lispssouts, 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 ./lispssouts ./lisplibrary ./fonts
```

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

```
prompt% ln -s /sharedserver/lde/lispssouts /usr/local/lde/lispssouts
prompt% ln -s /sharedserver/lde/lisplibrary /usr/local/lde/lisplibrary
prompt% ln -s /sharedserver/lde/fonts      /usr/local/lde/fonts
```

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

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

```
prompt% cd /usr/local/lde
prompt% tar xvfb /dev/rx0 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
```

[This page intentionally left blank]

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.

Description

The script generates checksum files named *FOO.check* and compares them to the released *FOO.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.

Commands

ldechecksum [-cg] [medleydir [dir] dirgroup]]	[Command]
--	------------------

-c Generates checksums for your installed files and compares them with correct values. This is the default action.

-g Generates checksums for the files specified.

medleydir Name of the Medley installation directory. Default is **/usr/local/lde**.

dir Any specific directory residing under *medleydir*. Only relative pathnames with respect to *medleydir* are accepted.

dirgroup The directory group, either **all** (the default) or **lisp**, which includes the *X/install*, *X/lisplib* and *X/lispout* directories.

Output

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

Checking directory: /usr/local/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/local/lde/fonts/display/chinese :  
Directory not installed
```

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

Examples

```
prompt% ldechecksum
```

All files in the installed Medley directories in **/usr/local/lde** are checked.

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

This example checks all files in:

```
/usr/local/somedir/lde/install  
/usr/local/somedir/lde/lisplib  
/usr/local/somedir/lde/lispssysouts
```

```
prompt% cd/usr/local/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]

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) <code>alias runlde "lde ~/sysout -k xx"</code> then when you type <code>runlde</code> to the C-Shell, it acts as if you had typed <code>lde ~/sysout -k xx</code>
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 <code>setenv</code> 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 in order 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 Envos 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 <i>absolute pathname</i> 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 <i>relative pathname</i> 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 <i>beginning</i> 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 lower case letters.

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

[This page intentionally left blank]

A
 access key 13,19
 architecture, machine
 finding value of 25
 asterisk 37
AUTHOR (*File Attribute*) 40

B
 back trace 47
BACKGROUNDPAGEFREQ (*Variable*) 27
 backslash 34
BEEPOFF (*Function*) 28
BEEPON (*Function*) 28
\BeginDST (*Variable*) 18
 binary files 40,44
 binary image, executable 14,19
 brackets
 right angle 34
 square 34

C
 C-Shell 3,34,A-1
 carriage return 40,44
 case sensitivity 3,34,36,46
CHANGEBACKGROUND BORDER (*Function*) 28
 characters, special 34
 Chat A-1
CHAT (*Function*) A-1
CHDIR (*Function*) 37,39,51
 checksum 7,D-1
chmod (*UNIX Command*) 40
CLOCK (*Function*) 29
CLOSE (*Function*) B-1
CNDIR (*Function*) 38
 configuration, software 14
CONN (*Command*) 38
 console messages 30
 conventions
 URaid 46
 copy protection 13
CREATE-PROCESS-STREAM (*Function*) B-1
CREATEDSKIRECTORY (*Function*) 28
CREATIONDATE (*File Attribute*) 39
.cshrc file 1,15,19

D
 Daylight Savings Time
 setting values for 18
DEFAULTFILETYPE (*Variable*) 40
DEFAULTFILETYPELIST (*Variable*) 40,41
DEFAULTPRINTERTYPE (*Variable*) 18
DEFAULTPRINTINGHOST (*Variable*) 18
df (*UNIX Command*) 12
DIRECTORIES (*Variable*) 18
 directory
 creation 38
 deletion 38
 enumeration 37
 home 34,39
 parent 34
 changing 34,39
DIRECTORY (*Function*) 36
 directory name

 delimiting 34
 disk file space 5
DISKFREEPAGES (*Function*) 28
DISKPARTITION (*Function*) 28
 display functions 27
 display fonts
 how to find 18
DISPLAYFONTDIRECTORIES (*Variable*) 18
DLRS232C 31
DLTTY 31
{DSK} 24,33,37,38,39,40
 special characters 34
 naming conventions 34
{DSK}INIT. 17
DSKDISPLAY (*Function*) 29

E
 emulator 2
 end-of-line convention 44
\EndDST (*Variable*) 18
 environment variable 1
 LDEDESTSYSOUT 27
 LDEINIT 17
 LDESHELL A-1
 LDESOURCESYSOUT 20
 obtaining value of 25
EOL (*File Attribute*) 40
 end-of-line convention 40
 errors 41
 Lisp 50
/etc/mount (*UNIX Command*) 41

F
fg (*UNIX Command*) 27
 file attributes 39
 file name 3,33
 conventions 33,34
 Lisp 33
 recognition 33
 file protection bits, changing 40
 file recognition mode 36
 file resources
 exceeding 51
 file streams 33
 file system errors 41
 file types 41
 file variables 40
FILENAMEFROMID (*Function*) 28
 files
 finding 18
 binary 40,44
 open 38,51
 text 40,44
 transfer 44
 versionless 35
FINDFILE (*Function*) 36
FLOPPY.ARCHIVE (*Function*) 28
FLOPPY.CAN.READP (*Function*) 29
FLOPPY.CAN.WRITEP (*Function*) 29
FLOPPY FORMAT (*Function*) 28
FLOPPY.FREE.PAGES (*Function*) 28
FLOPPY.FROM.FILE (*Function*) 28
FLOPPY.MODE (*Function*) 28
FLOPPY.NAME (*Function*) 28

FLOPPY.SCAVENGE (*Function*) 29
FLOPPY.TO.FILE (*Function*) 28
FLOPPY.UNARCHIVE (*Function*) 28
FLOPPY.WAIT.FOR.FLOPPY (*Function*) 29
 fonts 2,8
 Interpress 18
 font directories 8
 frames, viewing 47
FX-80DRIVER 31

G
GETFILEINFO (*Function*) 39

H
 hardware
 identifying machine type 23
 host access key 13,19
 installing 15
 host ID 13
 finding value of 25
 identifying 23
 host name
 identifying 23,25

I
/install.sunosX 14,C-1
 installation, software 11
 Interlisp package 46
 Interpress files 41
 Interpress fonts, finding 18
INTERPRESSFONTDIRECTORIES (*Variable*) 18

K
 Kermit 31,44
KEYACTION (*Function*) 20
 keyboard functions 27
 keyboard template
 Sun 3 20
 Sun 4 20
 keyboard tone generator
 turning on and off 28
KEYBOARDEDITOR 31
KEYDOWNNP (*Function*) 20
kill (*UNIX Command*) 48

L
lde 7,11,14,19,25
 killing 48
ldechecksum (*Command*) 7,D-1
LDEDESTSYSOUT (*Variable*) 26,27
ldeether 7,14,19
LDEINIT (*Variable*) 17
LDESHELL (*Variable*) A-1
LDESOURCESYSOUT (*Variable*) 19
 left angle bracket 34,37
 library files, finding 18
 line feed 40,44
lisp.virtualmem 20,26,27
 Lisp development environment 11
 Lisp symbols
 set in site initialization file 17
LISP-RELEASE-VERSION (*Variable*) 23
lisp.sysout 7
LISPDIRECTORYP (*Function*) 29
 LispUsers' Modules 9
 finding 18

LISPUSERSDIRECTORIES (*Variable*) 18
 litatom 46
 local disk functions 28
 .login file 19
LOGIN (*Function*) 24
LOGOUT (*Function*) 25,26
LONG-SITE-NAME (*Variable*) 18
LONG-SITE-NAME (*Function*) 18

M
 UNIX machine type, identifying 25
MACHINE-INSTANCE (*Function*) 23
MACHINE-TYPE (*Function*) 23
MACHINE-VERSION (*Function*) 23
MACHINETYPE (*Function*) 23
 Maiko 11
 makefile 7
 Medley, exiting 26
 memory, requirements 5
 MP errors 45
MY.NSADDRESS (*Variable*) 24

N
 naming conventions
 common {DSK} and {UNIX} 34
 network address
 identifying 24
 NFS service 5,41

O
 object file 14
OPENFILE (*Function*) 39
OPENSTREAM (*Function*) 39
 operating system requirements 6

P
 packages 3
 passwords, maintaining for access 24
 pathname, Lisp 33,36
 period 34,39
 double 34,39
PLAYTUNE (*Function*) 28
 Postscript 6,9
 Press files 41
 printers 6
 default 18
PROTECTION (*File Attribute*) 40
 protocol
 Ethernet 1,6,19
 PUP 1,19,44
pstat (*UNIX Command*) 11
 PUP protocol 1,19,44
 enabling 14
PURGEDSKDIRECTORY (*Function*) 28

R
RCLKMILLISECOND (*Variable*) 29
READSYS 48
REALMEMORYSIZE (*Function*) 23
 relative pathnames 34
 relinking 14
 right angle bracket 34
RINGBELLS (*Function*) 28
rmdir (*UNIX Command*) 38
root (*UNIX Command*) 19

S

SAVEVM (*Function*) 25,27
SCAENGEDSKDIRECTORY (*Function*) 28
 semicolon 34
setenv (*UNIX Command*) 26
SETMAINTPANEL (*Function*) 28
SETPASSWORD (*Function*) 24
SETTIME (*Function*) 29
setuid (*UNIX Command*) 24
SETUSERNAME (*Function*) 24
 shell script 15
SHORT-SITE-NAME (*Variable*) 18
SHORT-SITE-NAME (*Function*) 18
 site initialization file 17
 how to find 18
 site-init.lisp 17
SIZE (*File Attribute*) 40
 Sketch files 41
 slash 34
 software installation 11
 configuring 14
 special characters 34
 square brackets 34
 stack 47
 stderr B-1
 stdin B-1
 stdout B-1
su (*UNIX Command*) 12
 Sun Type 2 keyboard 20
 Sun Type 3 keyboard 20
 Sun Type 4 keyboard 20
 Sun Workstations
 sharing 6
 SunOS
 versions supported 6,39,43
 Type 4 keyboard 22
 console messages 30
 conventions 2
 directory notations 34
 file system 33
 interface to A-1
 username 24
 SunOS process
 identifying username of 24
SUSPEND-LISP (*Function*) 27
 swap space
 allocating additional 11
 required for Medley 5
 symbols
 set in site initialization file 17
 sysout 2,7,19,45
 locations of 19,43
SYOUT (*Function*) 27,51
 sysout files 44
 system administrator 9,11,19

T

tape device type 12
 tape access 6
 tar tape 7
 TCP/IP 31,44
 TCPCHAT 31

Teleraid 45
 template
 Sun 3 keyboard 20
 Sun 4 keyboard 20
 text files 40,44
 tilde 34
 tilde-name 35
 tilde-slash 37,39
 time
 BeginDST (*Variable*) 18
 EndDST (*Variable*) 18
 SETTIME (*Function*) 29
 TIME (*Function*) 29
 timers 29
 /tmp/XXXX-lisp.log 30
TYPE (*File Attribute*) 40

U

{UNIX} 24,31,37,38,39,40
 naming conventions 37
{UNIX} (*Function*) 33
 UNIX conventions 2
 UNIX process, suspending 27
UNIX-FULLNAME (*Function*) 24
UNIX-GETENV (*Function*) 25
UNIX-GETPARM (*Function*) 25
UNIX-STREAM-CLOSE (*Function*) 1
UNIX-USERNAME (*Function*) 24
 UNIXChat A-1
 UNIXComm B-1
 URAID 45
 commands 46
 quit 46
 user IDs, maintaining for access 24
USERGREETFILES (*Variable*) 18
 /usr/local/lde 12,C-1
USERNAME (*Function*) 24

V

version, identifying machine 23
 version numbering 35
 version numbers 33,37
VIDEOCOLOR (*Function*) 28
VIDEORATE (*Function*) 28
 virtual memory 25
 finding number of pages in use 26,27
 saving with URAID 48
VIRTUALKEYBOARD 31
VMEM.PURE.STATE (*Variable*) 27
VMEMSIZE (*Function*) 26
VOLUMES (*Function*) 29
VOLUMESENIZE (*Function*) 28
 VRAID 48

W

WRITEDATE (*File Attribute*) 39

X

XNS Ethernet protocol 1,6,19
 enabling 14

[] 34

\ \ 34

\10MBWATCHER 30
\BeginDST (Variable) 18
\EndDST (Variable) 18
\MY.NSADDRESS (Variable) 24
\RCLKMILLISECOND (Variable) 29

{
 {DSK} 24,33,37,38,39,40
 special characters 34
 naming conventions 34
 {DSK}INIT. 17
 {UNIX} 24,31,37,38,39,40
 naming conventions 37
 {UNIX} (Function) 33

-
- 34
-/ 37,39
-/lisp.virtualmem 20,26,27

*
* 37
LONG-SITE-NAME (Variable) 18
SHORT-SITE-NAME (Variable) 18

. 34,39
. 34,39
.cshrc file 1,15,19
.login file 19

/
/ 34
/etc/mount (UNIX Command) 41
/install.sunosx 14,C-1
/tmp/XXXX-lisp.log 30
/usr/local/lde 12,D-1

4
4045XLPSTREAM 31

;
; 34

<
< 34,37
>
> 34

[This page intentionally left blank]