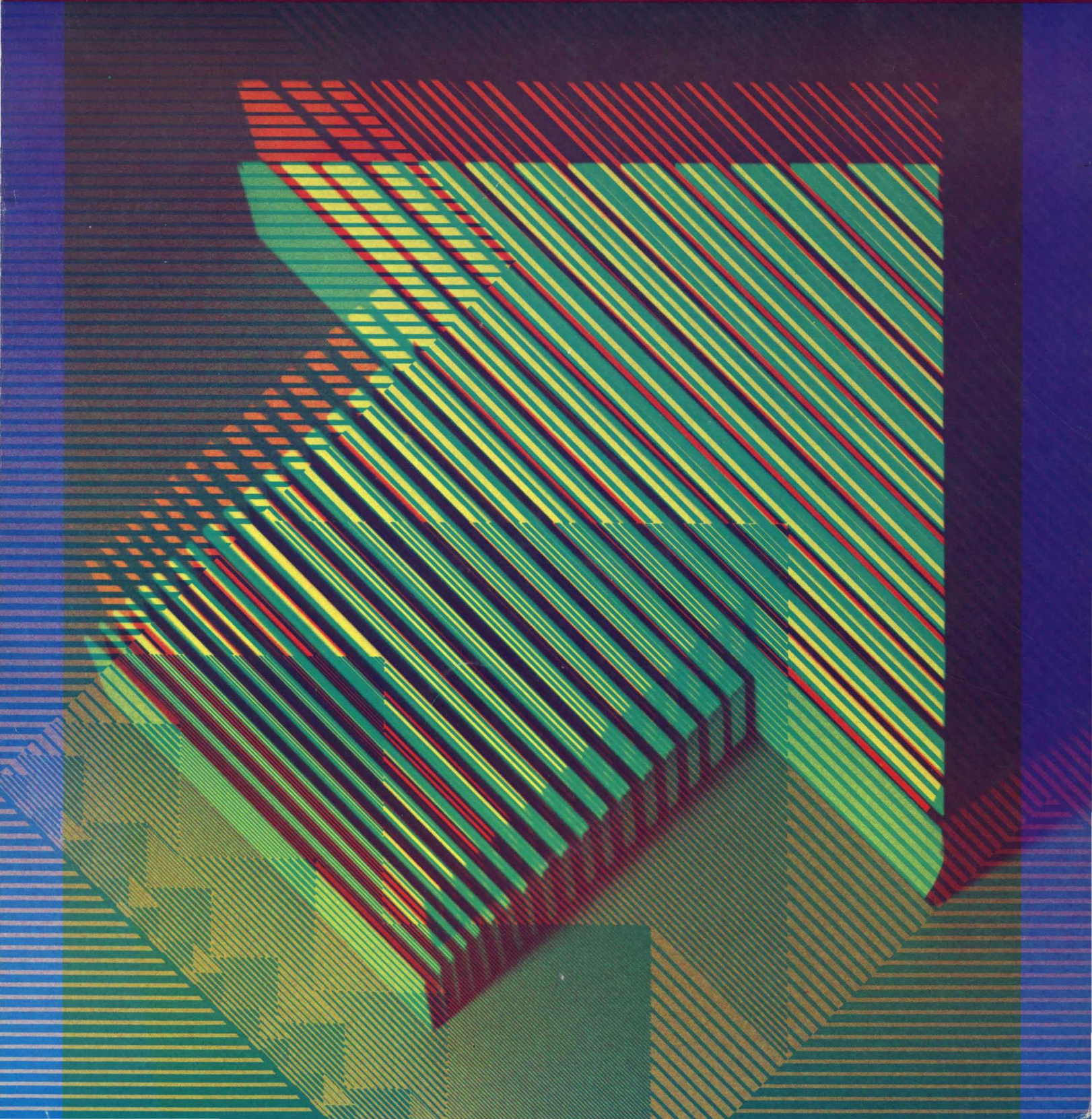


Installation and Site Operations

symbolics



Installation and Site Operations

symbolics

Installation and Site Operations

996105

March 1985

This document corresponds to Release 6.0 and later releases.

The software, data, and information contained herein are proprietary to, and comprise valuable trade secrets of, Symbolics, Inc. They are given in confidence by Symbolics pursuant to a written license agreement, and may be used, copied, transmitted, and stored only in accordance with the terms of such license.

This document may not be reproduced in whole or in part without the prior written consent of Symbolics, Inc.

Copyright © 1985, 1984, 1983, 1982, 1981, 1980 Symbolics, Inc. All Rights Reserved.
Font Library Copyright © 1984 Bitstream Inc. All Rights Reserved.

Symbolics, Symbolics 3600, Symbolics 3670, Symbolics 3640, SYMBOLICS-LISP, ZETALISP, MACSYMA, S-GEOMETRY, S-PAINT, and S-RENDER are trademarks of Symbolics, Inc.

Restricted Rights Legend

Use, duplication, or disclosure by the government is subject to restrictions as set forth in subdivision (b)(3)(ii) of the Rights in Technical Data and Computer Software Clause at FAR 52.227-7013.

Text written and produced on Symbolics 3600-family computers by the Documentation Group of Symbolics, Inc.

Text typography: Century Schoolbook and Helvetica produced on Symbolics 3600-family computers from Bitstream, Inc., outlines; text masters printed on Symbolics LGP-1 Laser Graphics Printers.

Cover design: Schafer/LaCasse

Cover printer: W.E. Andrews Co., Inc.

Text printer: ZBR Publications, Inc.

Printed in the USA.

Printing year and number: 87 86 85 9 8 7 6 5 4 3 2 1

Table of Contents

	Page
I. Site Operations	1
1. Introduction to Site Operations	3
2. Dumping, Reloading, and Retrieving	5
2.1 Performing Dumps	6
2.2 Reloading and Retrieving	10
2.3 Finding Backup Copies of Files	12
3. Using the Tape System	15
3.1 Distribution Tape Subsystem	15
3.1.1 Description of Distribution Tapes	16
3.1.2 Tape Access	17
3.1.3 Distribution Loader	17
3.1.4 Distribution Mapper	18
3.1.5 Distribution Copier	18
3.1.6 Distribution Dumper	19
3.1.7 Tape Spec Prompting	25
3.2 Carry-tape System	30
3.2.1 The Carry-tape Dumper	30
3.2.2 The Carry-tape Loader	32
3.2.3 The Carry-tape Lister	34
4. Multiple Partitions	35
4.1 Adding a Partition to LMFS	35
5. File System Maintenance Program	37
5.1 Free Records	37
5.2 Salvager	38
5.3 File System Maintenance Commands	39
6. Creating More Room on the Local Disk	43
6.1 Allocating Extra Paging Space	44
7. Saving New Versions of World Loads	45
7.1 Concepts	45
7.2 Manipulating the Label	45

7.3	Updating Software	46
7.4	Installing New Software	49
7.5	Installing New Microcode	54
8.	Instructions for Managing Disk Space on the 3640	55
8.1	Customizing and Saving the World	55
8.2	Saving Subsequent Worlds	57
9.	Booting a World	59
10.	Installing and Configuring the Mailer	61
10.1	Configuring Large Sites for Multiple Mail Servers	64
11.	Installing Printers	67
11.1	Installing an ASCII Print Server on UNIX	67
11.2	Installing an ASCII Print Server on VMS	68
12.	Registering Users and Hardware in the Site's Namespace	71
12.1	Registering Users	71
12.2	Registering Hosts	72
12.3	Registering a Tape Drive in the Namespace	72
12.4	Registering a Printer	73
12.5	Registering an ASCII Print Server on VMS	76
13.	Namespace Attributes for New Hosts	79
13.1	Namespace Attributes for a New Lispm Host	79
13.2	Namespace Attributes for a New UNIX Host	80
13.3	Namespace Attributes for a New VMS Host	80
14.	Site Related Functions and Variables	83
15.	Herald Functions and Variables	85
	II. Software Installation Guide	87
16.	Introduction to the Software Installation Guide	89
17.	Setting up a New Site	91
18.	Adding a New Machine to an Existing Site	93
18.1	Copying Worlds to Machines with Large Disks	93
18.2	Copying Worlds to Machines with Small Disks	94
18.3	Configuring Preinstalled Worlds on Machines with Large Disks	95
18.4	Configuring Preinstalled Worlds on Machines with Small Disks	96

19. Installing Release 5.3	99
20. Installing Release 6.0	101
20.1 Introduction to Release 6.0 Installation	101
20.1.1 Release 6.0 Distribution Tapes	101
20.1.2 Installing Release 6.0 on a 3640 with One 140-mbtye Disk	103
20.2 Release 6.0 Prerequisites	103
20.3 Procedures for Installing Release 6.0	104
20.3.1 Installation Checklist	106
20.3.2 Expanded Release 6.0 Installation Instructions	108
21. Related Installation Procedures	121
21.1 Preparing the Sys Host	121
21.1.1 Decide How to Set up the Sys Host	121
21.1.2 Delete Unneeded Parts of Release 5	131
21.1.3 Updating the Sys Translations File	131
21.2 Back up the Files on Your Machine	134
21.3 Booting a World	134
21.4 Loading System Sources and Documentation and Ephemeral-object Garbage Collector Diagnostic	135
21.5 Pruning the Documentation Database	136
21.6 Running the Ephemeral-object Garbage Collector Diagnostic	137
21.7 Using Disk Restore	138
21.7.1 Using Disk Restore During Installation	138
21.7.2 Release 6.0 Microcode Types	139
21.8 Set Site Command	139
21.8.1 Configuring a Namespace Server At an Established Site	140
21.8.2 Configuring a Namespace Server At a New Site Where the Sys Host is a Lisp Machine	141
21.8.3 Configuring a Non-namespace Host At an Established Site	142
21.9 Lgp Example	142
21.10 Distributing the New World and Microcode to Other Machines	142
21.11 Software Installation Terminology	143
21.12 Release 5 and Release 6 Directory Information	145
21.12.1 Description of Release 5 Directories	145
21.12.2 Release 6 Sys Host Size	147
21.12.3 More Information About Release 6 Directories	150
21.13 Field Software Support Offices	153
Index	155

List of Tables

Table 1.	Release 5 Directory Checklist	123
Table 2.	Sizes of the larger Release 6 Directories	126
Table 3.	<i>Release 6 Sources</i> Tape Checklist	129
Table 4.	<i>Release 6 Documentation and Examples</i> Tape Checklist	131
Table 5.	Size of Release 6 Directories in LMFS records	148
Table 6.	Size of documentation directories in LMFS records	149

PART I.

Site Operations

1. Introduction to Site Operations

This document provides information for various operations that are necessary to maintain Symbolics machines at your site. These procedures include the following operations:

- performing backups and dumps
- adding lmfs partitions
- using the distribution tape subsystem
- using the carry tape system
- performing file system maintenance
- saving new versions of world loads

2. Dumping, Reloading, and Retrieving

A file system can be damaged or destroyed in any number of ways. To guard against such a disaster, it is wise to periodically *dump* the file system, that is, write out the contents of the files, their properties, and the directory information onto magnetic tapes. If the file system is destroyed, it can then be *reloaded* from the tapes. Individual files can also be *retrieved* from tapes, in case a single file is destroyed, or just accidentally deleted (and expunged). Dump tapes can also be used to save a copy of all the files on a system for archival storage.

In a *complete dump*, all of the files, directories, and links in the file system are written out to tapes. This, obviously, saves all the information needed to reload the file system. However, a complete can take a long time and use a lot of tape, especially if the file system is large. In order to make it practical and convenient to dump the file system at short intervals, a second kind of dump can be done, called an *incremental dump*.

In an incremental dump, only those files and links that have been created or modified since the last dump (of either kind) are dumped; things that have stayed the same are not dumped. (All directories are always dumped in an incremental dump.) Now, if the file system is destroyed, you reload it by first reloading from the most recent complete dump and then reloading each of the incremental dump tapes made since that complete dump, in same the order they were created. Therefore, you do not need to retain incremental dump tapes made before the most recent complete dump was done; you can reuse those tapes for future dumps.

Since all incremental dumps done since a complete dump must be reloaded in order to restore the file system, doing a complete dump regularly makes recovery time faster. Doing complete dumps also lets you reuse incremental dump tapes, as described above. The more incremental dump tapes you must load at recovery time, the longer it takes to recover, and thus the more chance there is that something will go wrong. Thus, it is advantageous to take complete dumps regularly.

A *consolidated dump* is like an incremental dump, in that it only dumps files that have been created or changed recently. When a consolidated dump is requested, a *consolidation date* is specified. A consolidation date is always in the past. The consolidated dump dumps those and only those files that have been created or changed since the consolidation date. A consolidated dump is the appropriate kind to take if some event destroys recent incremental dump tapes, or they are found to be unreadable.

2.1 Performing Dumps

To perform a dump, first mount a magnetic tape on a tape drive connected to the machine. Then, in the File System Maintenance Program, click on [Local FS Maint] and invoke either [Incremental Dump] or [Complete Dump]. These commands respond by popping up a Choose Variable Values window that lets you set the parameters of the dump; the only difference among the commands is the initial value of some of the values in this window. Here are the parameters:

Dump Type There are three possible types of dump: *incremental*, *consolidated*, and *complete*. The two File System Maintenance commands initialize this field appropriately; the [Dump] command initializes it to *complete*.

Pathnames The pathname, or pathnames, specifying what is to be dumped. If there is more than one pathname, they are separated by commas. This value controls what files and directories are inspected for dumping. The type of the dump (complete, incremental, or consolidated) and the status of the individual files controls what subset of these files are actually dumped. Names of single files or links can be used to dump single files or links.

Wildcard specifications can also be used: this is the normal way to dump many files from one directory, or from a subtree. Subtrees are dumped via recursive (**:wild-inferiors**, *******) directory wildcards. The pathname you type is merged with "local:>*>*. *.*". To dump the whole file system, which is the normal default, the pathname ">*>*. *.*" is appropriate. To dump all the files in directory >foo>bar, and all of its inferiors, ">foo>bar>*>*. *.*" is the right value. To dump all the latest Lisp files in directory >abel>baker, but not any of its inferiors: ">abel>baker>*.lisp.newest". See the section "Naming of Files" in *Reference Guide to Streams, Files, and I/O*. See the section "LMFS Pathnames" in *Reference Guide to Streams, Files, and I/O*.

Tape Reel ID Every reel of tape produced by the dumper must have a Tape Reel ID, which is a string of up to eight characters. You must explicitly supply some value for this option. The reel ID is how this reel of tape is identified to the backup system; it appears in the dump maps and in any messages about the tape. The **:complete-dump-tape** or **:incremental-dump-tape** property of any file dumped is set to this value, as well. The Tape Reel ID should be physically written onto the tape so that the tape can be identified by sight.

Tape Host	The host name of the machine on which the tape drive to be used appears. This is initialized to a reasonable default, which is usually "LOCAL" if there is a tape drive present on the local host.								
Tape Drive ID	The identification number of the tape drive on which the reel of tape is mounted. This field can be left blank if the host on which the tape drive appears has only one tape drive, and thus needs no information to select the drive to be used. For hosts having multiple drives, a character string identifying the drive to be used must be entered.								
Dump deleted files	This can be either <i>Yes</i> or <i>No</i> , and says whether files marked as deleted but not yet expunged should be dumped on the backup tape. The default value is <i>No</i> ; deleted files normally are not dumped.								
Append to tape	This can be either <i>Yes</i> or <i>No</i> . The default is <i>No</i> (you cannot append to cartridge tape). When <i>Yes</i> , this option instructs the dumper to go to the end of the data on the current tape, which is assumed to be a backup dumper tape from an immediately previous dump, and begin appending new files to the end. Select <i>No</i> to force it to start writing from the beginning of the tape.								
Tape when done	This controls what the dumper does with the tape when it has finished passing over all the specified pathnames. These are the available options: <table><tr><td>Offline</td><td>Puts the tape offline, rewinding it. It declares the dump finished. This is the default.</td></tr><tr><td>Rewind</td><td>Rewinds the tape without setting it offline. It declares the dump finished. It facilitates listing or verifying the tape contents.</td></tr><tr><td>Leave</td><td>Leaves the tape positioned at the end without rewinding it or putting it offline. It declares the dump finished. It facilitates more dumping later, by using "Append to tape" in a later dumper invocation.</td></tr><tr><td>Query</td><td>At the end of the dump, all of these options are presented, and you get to choose whether to rewind and set the tape offline, rewind it, leave it at end, or dump some more files. If you choose the latter, the dumper menu is offered again, and the new files appended to this tape. The dump is</td></tr></table>	Offline	Puts the tape offline, rewinding it. It declares the dump finished. This is the default.	Rewind	Rewinds the tape without setting it offline. It declares the dump finished. It facilitates listing or verifying the tape contents.	Leave	Leaves the tape positioned at the end without rewinding it or putting it offline. It declares the dump finished. It facilitates more dumping later, by using "Append to tape" in a later dumper invocation.	Query	At the end of the dump, all of these options are presented, and you get to choose whether to rewind and set the tape offline, rewind it, leave it at end, or dump some more files. If you choose the latter, the dumper menu is offered again, and the new files appended to this tape. The dump is
Offline	Puts the tape offline, rewinding it. It declares the dump finished. This is the default.								
Rewind	Rewinds the tape without setting it offline. It declares the dump finished. It facilitates listing or verifying the tape contents.								
Leave	Leaves the tape positioned at the end without rewinding it or putting it offline. It declares the dump finished. It facilitates more dumping later, by using "Append to tape" in a later dumper invocation.								
Query	At the end of the dump, all of these options are presented, and you get to choose whether to rewind and set the tape offline, rewind it, leave it at end, or dump some more files. If you choose the latter, the dumper menu is offered again, and the new files appended to this tape. The dump is								

not declared finished at this time. Clicking on "Abort", however, declares the dump finished immediately.

- Person operating** The identification of the person doing the dump. This is entered into the backup map, as well as setting as the file author of that map. Normally, this is the same as the login ID of the user taking the dump, and that is its default value. However, for server systems, usually logged in as "File-Server" or some other user not representing a real person, this field ought to be filled in. It is important for documentation purposes and site record-keeping.
- Consolidate from** This field is only used during a consolidated dump. It is a date and time, in any acceptable Lisp Machine format, in the past. It is parsed in such a way that it is expected to be in the past. The consolidated dump dumps all, and only, files that have been created or modified since this date.
- Set date dumped** When the dumper finishes writing a tape, it marks all the files it has dumped as having been dumped on that tape at this time, and creates a *tape directory*, as described below. These measures allow the file to be retrieved later, and indicate that the file no longer needs to be dumped in incremental dumps. This is the default action, which corresponds to a value of *Yes*. If you use the dumper to make tapes that are not part of your site's backup, such as for moving software between machines, you do *not* want to indicate that the files were dumped, or to make a tape directory. Select *No* in this case. The carry system and the distribution tape system should be used, instead of the LMFS dumper, to move software between sites, as it is far more general, and system-independent.
- Restart pathname** If this pathname is supplied, that is, not blank, the pathname of every file to be dumped is compared with this pathname, and only files with pathnames "greater" than it are actually dumped. The "comparing" order is similar to that performed by the function `fs:pathname-lessp`; what is important is that it is the *same* for the same set of files every time a dump is taken or attempted.
- The purpose of this feature is to allow restarting of complete dumps that are interrupted by any sort of failure. When the dumper finishes a tape, it prints out the pathname of the last file dumped. Although this is recorded in the dump map, *the pathname and the name of the tape should be recorded on paper by the person taking the dump*, especially if a complete dump is being taken.

To restart a dump, fill out the menu as usual, but type in the pathname of the last file known to have been dumped as the value of [Restart Pathname]. The dumper scans the sub-hierarchy indicated, but does not dump files already dumped, or even progress down, seeking files to be dumped, into directories that the restart pathname indicates have already been processed. The skipping of files and directories already dumped is based on sorting order, not actually having been dumped. Thus, if files A, C, E, G, and I exist, and files A through E get dumped one day, and the dump is interrupted and restarted from E the next day, a D created in the interim is be dumped. This is reasonable because, if the complete dumper were not interrupted, it would not have dumped it, as it did not yet exist, either.

Comment

A string of arbitrary contents, written on each reel of the dump and in the dump map. This might say why the dump was taken or any other special information about this dump.

Note that you *must* supply a Tape Reel ID. When you are done, click on the [Do It] box; if you decide not to do a dump after all, click on the [Abort] box. If there is something wrong about the set of parameters you have specified, the program types out a message and present you with the Choose Variable Values window again. Otherwise, it types out a message saying that the dump has started successfully, and proceeds. While the dump is in progress, the name of the file that is being dumped is shown in the far right-hand field of the status line (this is the field that normally shows you the names of files that are being read or written).

The dumper creates a file called the *dump map*. The dump map is a character file giving a complete description of what has been dumped, directory by directory and file by file, including the time of dumping, the tape on which the file was dumped, the tape reel ID of the previous tape on which the file appears (if any), and so on. The dump map is created in the **>dump-maps** directory. Its name is constructed from the type of dump and the date and time at which it was started; the type is **map** and the version is 1. A typical dump map might have the pathname:

```
>dump-maps>complete-9/15/81-9:02.map.1
```

The dumper puts all information about the dump, the operator, the time of day, the options, and so forth, in the map. It also puts error recovery information there, and descriptions of tape-changings, as well as the number of files dumped on each tape. The dumper performs a **:finish** operation on the map file at the end of each tape, so that if the system crashes during a multitape dump, information about previous tapes is guaranteed to be intact and accessible.

The dumper also creates a file called the *tape directory* for each separate reel of each dump. This is a binary file saying what is on the tape, with more or less the same information as the dump map. This file is used when you try to locate dumped

copies of a file. See the section "Finding Backup Copies of Files", page 12. The tape directory is also created in the >dump-maps directory. Its name is the tape reel ID of the tape, its type is directory, and its version is 1. A typical directory map might have the pathname:

```
>dump-maps>INC00001.directory.1
```

The dumper dumps files successively to tape, and at the end of each tape, rewinds and unloads the tape, asking for a new tape if there are more files to be dumped. It is only after it has done this that it sets backup dates for the files and makes the dump directory.

If the dumper gets an unrecoverable error while writing a tape, it attempts to write end-of-file marks on the tape and inform you of what has happened. It gives you the option of either considering the files on that tape to have been validly dumped, in which case the dump continues on the new tape, or discarding the tape, in which case it redumps, on the (or several) new tapes, all the files that it had dumped on the bad tape. The problem and its chosen recovery are described in the dump map.

The dumper normally tries to read each tape before writing on it. This is to avoid accidentally overwriting valuable tapes. For tapes to be appended to, this is necessary. For other tapes, it is desirable. It often takes a long time to attempt to read blank tape, to prove that a new tape is really new. The dumper explains and queries if it is not confident that the tape being written on is the right one.

Some sites can choose to waive this checking. This is necessary when tape hardware is in use that cannot time out while reading blank tape, and unstoppably reads the whole tape when a new tape is checked. The site option, an attribute of the Site object for a site, **validate-lmfs-dump-tapes**, normally elected, enables suppressing this checking.

2.2 Reloading and Retrieving

Reloading is the process of moving all the files on a backup tape into a local file system. *Retrieving* is the process of moving selected files.

Reloading and retrieving can load files onto any LMFS.

Related to the functions of reloading and retrieving are two other functions, listing the contents of backup tapes and verifying their contents. The *reloader* program implements all four of these functions.

To invoke the reloader, click on [Reload/Retrieve] in the File System Maintenance program. The reloader Choose Variable Values window appears, with the following items:

Operation Wanted This controls what function the reloader is to perform.

- Reload all** The reloader is to read the tape, and reload all files on it. It will not reload files that are already present.
- Retrieve Single Files** The reloader will expect a list of wildcard pathnames, to tell it which files to reload. It will not reload files that are already present. These pathnames are specified by the menu item [Files to retrieve].
- List tape** The reloader will read the tape and describe its contents on the terminal. It will describe the dumps appearing on the tape, and which files are on the tape. This does *not* verify that the files were dumped correctly; use "Compare" for that.
- Compare** The reloader will read each file on the tape, look for the same file in the file system, and compare the two, bit for bit, reporting any discrepancies. Use this option to verify that a dump tape just created in fact contains good data.

After selecting the operation to be performed, the following information must be supplied:

Tape Spec: Local: Cart

The tape host and the tape drive. The tape host is the name of the machine that contains the tape drive to be used. Normally, for a 3600 with cartridge tape the default is "Local", which is an acceptable expression of the local machine. Otherwise, a reasonable default will be chosen, based on available tape servers at your site. The tape drive is a character string identifying the tape drive to be used on the tape host. This value needs to be supplied only if the machine selected has more than one tape drive, and needs this information to select one.

File Host:

The name of the machine onto which the files will be reloaded. Normally, this is the local machine. For the [Compare] option, this is the machine containing the files to be compared to the files on tape. Files are always reloaded into the exact place in the directory hierarchy from which they were dumped (perhaps on a different machine).

Files to retrieve (*'s ok):

One or more pathnames, which are usually wildcard pathnames. This is used when you choose [Retrieve single files]. Any file matching any one of these wildcard pathnames will be reloaded, unless it is already there. For more information, see the following sections: See the section "Naming of Files" in *Reference Guide to Streams, Files, and I/O*. See the section "Wildcard Pathname Mapping" *Reference Guide to Streams, Files, and I/O*. See the section "LMFS Pathnames" in *Reference Guide to Streams, Files, and I/O*.

When you have made your choices from the reloader Choose Variable Values window, you can mount the tape, and click on [Do It] to allow the reloader to begin. You can also click on [Abort] to not proceed with the reload, tape list, or compare.

The reloader will print information about each file it reloads or about every file on the tape, if [List tape] was selected. When it gets to the end of the tape, it will stop. If you want to do more reloading, you must mount another tape and restart the reloader. Tapes can be reloaded in any order.

The reloader does not delete or expunge files. If you are reconstructing a file system from backup tapes that include many incremental backups, you must occasionally intervene to delete and expunge unwanted old files that are reloaded, in order to ensure adequate file space.

A few notes about the comparer: Occasionally, hardware problems can cause bad backup tapes to be written, without any error having been detected by the dumper. You should always verify a backup by using the "Compare" option of the reloader to *verify* the backup tape. The comparer verifies each bit of every file on the tape, and, for files that still exist, reports any discrepancy.

If you find that an incremental backup tape is deficient, and you decide that the files must be redumped, you must take a consolidated dump, with a consolidation date equal to the time the bad incremental dump started. You must delete the backup dump tape directory ("*tape-name.directory*" in >dump-maps) by hand, if the the backup-copy finding mechanism is to be aware that the tape has been abandoned.

The dumper assumes that no undetected problems occurred. Thus, it does not use the comparer automatically. The dumper sets backup times when it finishes each tape.

2.3 Finding Backup Copies of Files

In order to retrieve a file, or files, you must know on what tape the file or files were dumped. You do not usually remember this information, since you might not anticipate losing a file.

The LMFS backup mechanism provides a way to search the binary tape directories, which are produced by the dumper, to find backup copies of files. This is the Lisp function **lmfs:find-backup-copies**, which you can invoke by [Reload/Retrieve (M)].

lmfs:find-backup-copies searches all the binary tape maps at the site for the tape locations of all backup copies of a file. It prompts for names, which it parses with respect to the local host unless you provide an explicit host name. All of the files requested must be from the same host. This applies only to Lisp Machine hosts. It looks at the binary tape maps on the specified host in the directory >dump-maps.

Normally, you specify wildcard pathnames to match. A non-wild pathname is considered to be a valid degenerate case of a wildcard pathname. The default with which all of the names are merged is "local:>**>*.*.*". The same pathname interpretation conventions as used by the reloader apply.

Here is a sample interaction with **lmfs:find-backup-copies**:

Enter file pathnames for which to search, separated by commas.

Wildcard names are allowed. The default host is LOCAL-LISPM.

Paths: f:>sys>io>*.lisp, f:>bsg>*.init, f:>lisp>*.q*

Searching for:

F:>sys>io>*.lisp.*

F:>bsg>*.init.*

F:>lisp*>*>*.q*.*

F:>LISPM>COLDL.D.QBIN.39, created 2/17/83 00:26:58, on tape fscns001

(Backup dump of 2/24/83 13:05:44)

F:>LISPM>COLDUT.QFASL.57, created 2/18/82 19:06:36, on tape fscns001

(Backup dump of 2/24/82 13:05:44)

F:>LISPM>UTILS>NEW>EVAL.QBIN.13, created 2/19/83 04:45:17, on tape fscns001

(Backup dump of 2/24/83 13:05:44)

....and so on

----- Scan complete.

3. Using the Tape System

Symbolics supports at least five different tape formats, all different and incompatible. Each format is specific to one Lisp Machine tool. Many people wonder which tool is appropriate for which application.

The *LMFS dumper and reloader* are used for snapshotting files on a local Lisp Machine file system and reloading those files on the same local Lisp Machine file system, in the same place at a later date. The intended use of LMFS backup is to reload files onto the same machine from which they were dumped.

The *Distribution dumper and loader* are intended to distribute transportable systems and libraries, defined by system declarations and pseudosystem declarations, on logical hosts, from one site to another. The distribution system specializes in finding appropriate source and object versions of systems, appropriate patch files, and so forth. It is cumbersome to use it for transporting miscellaneous files between sites or hosts.

TAPEX is a format for transferring character (source) files (only) between hosts. It is the only one of these formats which can be read or written by other than a Lisp Machine. TAPEX programs exist for TOPS-20 and Multics, as well as for the Lisp Machine (**tape:tapex**). It cannot deal with any type of file except character files, and has no features to speak of. Each individual dumping or loading requires an interaction.

Carry tape is the most general tool, and is meant to dump individual files and sets of files given by wildcard filespec, and load them at any site. Recent improvements to the wildcard facility make this a very powerful and easy-to-use tool.

IFS tape is a format specific to the 3600-family machines and is readable by the FEP. IFS tape is used to transport and distribute 3600-family FEP files and file systems, including world loads and microcode.

3.1 Distribution Tape Subsystem

The distribution tape subsystem is used at Symbolics to write, load, and inspect distribution tapes. Customers use the subsystem mostly for loading the distribution tapes they receive from Symbolics; that is, they copy the files on the distribution tapes to their own file systems. However, customers can also use the distribution tape subsystem to write their own tapes.

The distribution tape subsystem has four components:

- The *distribution dumper*, which makes distribution tapes.

- The *distribution loader*, which manages the transfer of tape files to customer file systems.
- The *distribution mapper*, which produces a listing of the contents of a distribution tape. Although the dumper produces such a listing as it creates the tape, the mapper makes it possible to view a listing if you were not the one who dumped it.
- The *distribution copier*, which copies distribution tapes.

The distribution tape subsystem runs on a Lisp Machine and requires a tape drive. The tape drive can exist on that machine, or it can be part of another machine (including a non-Lisp Machine) that is accessible through the Chaosnet and has a remote tape server program.

3.1.1 Description of Distribution Tapes

Distribution tapes contain the following:

- Descriptive information
- System source files
- Object files

The descriptive information, located at the start of the tape, includes a summary of the tape's contents and information about its creation (for example, the site, date and time, and user id of the person dumping the tape).

The objects on the tape are organized into:

- Systems
- Bands (FEP files)

Systems are either *systems*, defined by the **defsystem** special form, or *pseudosystems*, defined by the **dis:defpseudo-system** special form. Typical systems are the core, or "System", system; Zmail; and the Lisp Machine local file system (LMFS). Typical pseudosystems are the TV font library and the release documentation. Whereas systems consist of carefully coordinated source, object, and patch files whose versions are often interrelated, pseudosystems usually consist of all the latest files matching some wildcard file spec and are not subdivided into sources and objects.

Bands on the tape contain world loads. The tape can contain any number of bands, which can be copied directly into disk partitions.

All files on the tape are marked with the name of the system or pseudosystem to which they belong and are identified on the tape by logical (system-independent) pathnames.

Distribution tapes are limited to one reel.

3.1.1.1 System Source Policy

It is Symbolics policy to distribute source programs from which the original loaded object programs were made, as well as all patch-file sources and objects. These, taken together, a useful technical reference and resource.

A system source consists of a great many files because of the way in which we maintain and develop systems. When a new version of a system is loaded, the system software makes a note of the pathnames of the source files from which the newly loaded object files were made. As we develop these programs, new versions with higher version numbers are created, some incorporating new features as well as bug fixes. Sometimes when a problem is found, a patch is necessary; the developer repairs a piece of code in the current version of the system source, produces a new version of the source, and creates a patch file containing a new version of the failing function. System users can then load the compiled version of this patch into their Lisp Machine environment.

Note that at this point no version of the source program exists that corresponds exactly to the code running in the Lisp Machine environment. The source version from which the original loaded object was compiled is obsolete, because it does not contain the repaired version of the failing function. The latest version of the source is not accurate either, as it contains new features and other modifications that are not yet part of the system.

Therefore, the accurate source is a hypothetical entity partially constructed of the original source, and later patch files.

3.1.2 Tape Access

The distribution loader, distribution dumper, distribution mapper, and distribution copier all prompt for a tape spec: See the section "Tape Spec Prompting", page 25.

3.1.3 Distribution Loader

To invoke the loader, type:

```
(distribution:load-distribution-tape)
```

Keep in mind that the loader attempts to load files into places specified by logical pathnames, creating directories as needed. If you use a host other than the Lisp Machine, you might have to create directories prior to running the loader.

For a discussion of the general issues of tape access: See the section "Tape Spec Prompting", page 25.

3.1.4 Distribution Mapper

The distribution mapper produces a listing of the tape contents that is almost identical in format to that produced by the dumper. For each tape, it lists the following:

- The systems, pseudosystems, and world loads on the tape
- Creation information for the tape, including site, date and time, and user id of the operator

For each file it also lists the following:

- The logical pathnames and corresponding physical pathnames of files actually used
- The authors
- The creation dates in Greenwich Mean Time

To invoke the distribution mapper, type:

```
(distribution:map-distribution-tape)
```

A menu appears, identical to that of the distribution loader. On the menu you must specify the source of the distribution tape, including the following:

- Tape reel name, if needed by the server
- Host, if remote

The operating instructions for this menu and the tape operation are the same as for the distribution loader. The map is written to a file in the directory `sys:distribution;`. The distribution mapper prints out the name of this file when it starts and returns `nil` when it is done.

For a discussion of the general issues of tape access: See the section "Tape Spec Prompting", page 25.

3.1.5 Distribution Copier

The distribution copier makes a copy of an already written distribution tape onto another tape.

The distribution copier does not interpret the information on the tape. However, it is not a general tape-copying utility; use `tape:copy-tape` for that purpose.

To invoke the distribution copier, type:

```
(distribution:copy-distribution-tape)
```

The distribution copier offers two menus, each identical to that of the distribution loader. One menu is for the source tape, the other for the destination tape. As soon as you complete the second menu, the tape is copied to its logical end.

For a discussion of the general issues of tape access: See the section "Tape Spec Prompting", page 25.

3.1.6 Distribution Dumper

The distribution dumper writes systems on a tape.

To invoke the distribution dumper, type:

```
(distribution:write-distribution-tape)
```

The distribution dumper presents a small frame labeled *Distribution Dump Options*, which is divided into four panes. From top to bottom, the panes are as follows:

- Choose Variable Values menu
- Systems List
- Interaction window (used for displaying prompts and typing responses)
- Command Menu

Your basic interaction with the distribution dumper consists of the following steps:

1. Select systems for dumping.

See the section "The Dumper Systems List", page 20.

See the section "The Dumper Choice Boxes", page 21.

All objects on the resulting tape are part of a specified system.

2. Indicate how to access the tape drive. See the section "The Dumper Choose Variable Values Menu", page 19.
3. Press the END key or click on [Do it] to begin the dumping.
4. The dump should proceed without further intervention. However, if problems occur: See the section "Problems During Dumping", page 23.
5. The distribution dumper produces a listing almost identical to that of the mapper and places the listing in the directory sys:distribution;.

3.1.6.1 Choose Variable Values Menu

The Choose Variable Values menu, located at the top of the frame, controls access to the tape; it specifies the following parameters:

- Tape Host
- Tape reel name
- Tape Drive, if required. TD-80 drives on 3600 systems are denoted by drive identifiers 0, 1, etc. The cartridge drive, when present, is denoted by the drive identifier "cart".
- Tape Comment

To change the tape host, use either the mouse or the keyboard. This option accepts the name of any host or Local.

- Click on the Choose Variable Values menu item default tape host Local and enter a new tape host name.
- Give the H command. The command prompts for a tape host name.

For a discussion of the general issues of tape access: See the section "Tape Spec Prompting", page 25.

3.1.6.2 Dumping World Loads

The distribution dumper does not dump world loads on the 3600 family machines. Use the function **tape:write-fep-files-to-tape** instead.

tape:write-fep-files-to-tape &optional *mic-name* *Function*

Writes FEP files to tape. *mic-name* is the name of file-format microcode that precedes the microcode and world load files on distribution tapes.

When an argument is supplied within the form, the function assumes that the argument is the file-format microcode and uses stream format. When an argument is not supplied, you are prompted for a file name, which is assumed to be a microcode or world load file and which is then written out in distribution format. Thus, supplying a file-format microcode name should be used only when writing an initial microcode file to tape.

You will be prompted as to whether the first tape is in place. Put the tape in the local tape drive and then answer "Y". You will then be prompted as to whether you wish to write a file to tape; you should answer "Y". Next, enter the filename of the world load. You will also be prompted for file and restoration comments. As the file is written out, the number of blocks will be printed on the screen. When the end of the tape is reached, the following message is printed:

"starting a new tape"

and you will be prompted as to whether a new tape is in place. Put a fresh tape in the drive and type "Y" to continue. This will continue writing the file on the second tape.

This function writes only 20-Mbytes into a tape, although 3640 and 3670 tape drives are capable of writing 45-Mbytes. This provides compatibility with 3600 tape drives.

3.1.6.3 Systems List

The Systems List pane (the middle pane) displays the names of the systems selected for dumping as well as the types of files to be dumped for each system.

The distribution dumper considers systems to consist of four types of files. To make meaningful use of the distribution dumper, it is important that you understand the definitions of these file types and the distinctions between them.

- *Sources* are the source files from which a system was originally compiled.
- *NewSrc* are the newest version of the sources, possibly containing modifications and bug fixes.
- *Objects* are the object files for the system.
- *Patches* are the patch file sources, patch file objects, the patch directory, and the system version directory file.
- *Low* are the patch numbers.

Keep in mind that although the distribution dumper can dump sources, newest sources, objects, and patches in any combination, not all of these four types are available for all systems. For example, pseudosystems have only newest sources. Similarly, the distribution dumper will not offer to dump patches for systems that are not described in their system declarations as patchable.

To specify that you want to dump a given system, do one of the following:

- Use the `S` command and then, when prompted, type the name of the system.
- Use [Add System] in the Command Menu and type the name of the system.

The distribution dumper then attempts to load the system declaration (via the site file mechanism) for any system whose declaration is not already loaded into the environment. Once a declaration is loaded, it is used to identify the source and object files of the system.

When you specify a system to dump, the distribution dumper inspects the system's system version directory file to determine the current version of the system. If the original version of the system is loaded —as opposed to the newest version (or current, as determined from the system version directory file) — you are asked which (if either) you wish to dump. If you choose to dump the newest version, only latest sources, not sources, are offered for dumping.

For patchable systems, the distribution dumper determines the identity of the current sources and objects from the system's version directory file.

To delete a system from the Systems List, do one of the following:

- Click middle on the system name in the Systems List.
- Use [Delete System] in the Command Menu and either type the system name or click left on the system name in the Systems List.

3.1.6.4 Choice Boxes

When you successfully specify a system to be dumped, the Systems List displays the name of the system and up to four mouse-sensitive choice boxes. Each box selects the dumping of sources, newest sources, objects, and patches, respectively. You can click on any choice box to turn it on or off. For patchable systems, when patches are selected for dumping, the highest and lowest patch numbers to be dumped are

also displayed. These patch numbers are also mouse-sensitive: Click on them to enter new values.

The following chart sums up the default settings for different systems:

<i>Type of system</i>	<i>Default Setting</i>
Loaded systems	Sources and patches
Pseudosystems	Newest sources
All other (nonloaded)	Newest sources, objects, and patches

The default setting of these boxes depends both on the type of system being dumped and on the particular system that is loaded. The following rules illustrate this point.

- The dumper offers to dump *only* sources if the system was loaded into the current world.
- By default, the dumper dumps the original sources, not the newest sources, of loaded systems.
- The dumper does not by default dump objects for loaded systems, because it believes that these objects are already loaded into both the band that is running and the user's band.

If you wish to override the default, then click on the appropriate box to indicate this choice.

3.1.6.5 Command Menu

The Command Menu, the bottom pane, allows you to alter entries in the Systems List.

For example, you might want to modify the choice boxes for all systems to be dumped, let's say, in order to make a patch tape consisting only of patches and the release version file. To this end the Command Menu provides the [Choice Boxes] item.

<i>Mouse click</i>	<i>Action</i>
[Choice Boxes (L)]	Modifies all lines of the Systems List to implement the standard dumping policy. (See the section "The Dumper Choice Boxes", page 21.)
[Choice Boxes (M)]	Modifies all lines to set the standard options for making a patch tape. A patch tape consists of patch files only.
[Choice Boxes (R)]	Invokes a menu giving a wide variety of useful operations to be performed on all lines of the Systems List; for example, do not dump any patches and leave all other choices intact, dump latest sources in addition to original sources for all loaded systems, and so on.

Two items on the Choice Boxes Menu deserve special mention. [Clear all choices] turns off all choice boxes. [Prune systems] removes from the Systems List all systems for which no choice boxes are selected. (The dumper would ask you later about such systems if you had left them in the list.)

These powerful global operations only transform the Systems List; they do not establish modes of dumping. If no mode exists that does exactly what you want, click on appropriate individual choice boxes before starting the dump.

3.1.6.6 Giving the Go-ahead

Once you have selected all the items and are ready to begin the dump:

1. Make sure that a tape reel name has been specified. If not, do one of the following:
 - Type **R** to specify a tape reel name (or the dumper forces you to do so).
 - Use the appropriate field in the Choose Variable Values Menu.
2. Make sure that all systems to be dumped have some choice boxes selected. Delete those systems that have no choice boxes selected. You can either delete such systems singly or use [Prune systems] to delete them all.
3. Abort or begin the procedure.
 - If you decide not to write a tape, click on [Abort] in the Command menu to end interaction with the dumper *without* writing a tape. Note that you do not exit from the dumper but return to the dumper command level.
 - If you decide to begin the dump, take one of these actions:
 - Press the **END** key.
 - Click on [Do it].

Once you have begun the dump, the frame is deexposed and the dumper proceeds without need for interaction, if all goes well. The screen displays the pathname of each file dumped, or, for patch files and microcode, a more accurate description. At this point you might want to disable **MORE** processing if you are dumping many files. The dumper finishes by returning **nil**.

3.1.6.7 Problems During Dumping

You can encounter problems if you try to dump systems whose pathnames are not expressed as logical hosts and logical directories. See the section "Distribution Loader", page 17. As stated in that section, the loader must have a host-independent specification of where to load files. If you attempt to dump systems whose files are specified as being on some physical (nonlogical) host, the dumper prompts with a query of this form:

<LISPM.NEW>QGRBL.LISP is not a logical (device-independent) pathname.
There is no way that this file can be loaded at the other end unless
a logical pathname is supplied. What do you wish to do? (N or P)

The choices are:

- N** Nothing. Do not dump this file at all.
- P** Prompt. The dumper prompts for a logical pathname. You must type a valid logical pathname for a defined logical host. The file is dumped, and the tape indicates as destination pathname the logical pathname you specify. The prompt for the new, valid logical pathname offers as a default either the physical pathname at issue or the last pathname offered by you in this dialogue. Although this default might be physical, its name, type, and version are probably correct and are cross-host defaulted into the logical pathname you enter if you do not supply a name, type, or version.

The dumper takes some pains to locate, for loaded systems, the source version that was compiled to produce the loaded object file. It looks for the identical name, type, and version. When the dumper is ready to dump that file, it issues a message if it cannot find the source fitting these criteria. The message is one of the following forms:

Cannot find a copy of SYS:IO;DISK LISP with the correct source version

Cannot find any source for SYS:IO;DISK LISP

You are then asked what to do. The choices are:

- N** Nothing; do not dump anything claiming to be this file, and go on. Make no mention of the fact that this file exists, or fails to exist, on the tape.
- T** Try again. Try again to dump it. Before you type this, presumably you have moved, renamed, or created some files.
- D** Directory list. List all files, and their authors and creation dates, for all files in the relevant directory with the same name as the file whose source could not be found.
- L** Latest. Dump the latest version instead of the specified version. This option is offered only if some versions can be found. If this version has a lower version number than the specified version, you are asked to confirm your choice. The loader does not wrongly assume that this version is the missing version, but loads it for what it is.
- E** Elsewhere. Prompts for a file name. The file takes the place of

the missing file. If it is of the same name and type, at least the version number is preserved.

If a tape error is encountered during dumping, you can restart the dump on another tape drive without respecifying the files to be dumped using the restart option that appears on the restart option list at the time of the error.

3.1.6.8 Defining a Pseudosystem

dis:deftpseudo-system *name* &rest *options* *Special Form*

Defines a pseudosystem, an arbitrary collection of files to be dumped by the distribution dumper and loaded by distribution loader. *name* must be a symbol in the distribution package (**dis**). The rest of the form consists of specifications of groups of files to be dumped. At least one such specification must exist. At present, only one kind of specification is supported:

(:files *spec type*)

spec can be either a file spec or a list of file specs for logical pathnames. They may contain wildcards. *type* specifies the format of the file:

:source-file	Character or text.
:object-file	All other cases.

3.1.7 Tape Spec Prompting

3.1.7.1 Tape Specs

The concept called a *tape spec* provides a means for describing, passing around, and defaulting tape host, drive, and parameter selection.

The basic idea is that **tape:make-stream**, which is called by all tape-using programs to create tape streams, is provided by programs with partially complete information, sometimes none at all. If this information is insufficient to open a tape stream, **tape:make-stream** prompts for more information.

Only one such prompt is issued by **tape:make-stream**. In this prompt, a complete specification (a *tape spec*) for all variable parameters of the tape stream to be opened is presented as a *default*. This default is made by merging previously used parameters, that is, the last time since boot that you created a tape stream, with ones explicitly provided by the tape-using program that called **tape:make-stream**. The meaning of this default, and its use in processing your response to the prompt, is very similar to the parallel concepts in pathname merging. If you press RETURN, the default is used as is to provide all parameters to **tape:make-stream**. Otherwise, you can specify individual parameters, which are *merged*, as described below, with the default, to produce parameters to **tape:make-stream**. The merged specification is saved to produce the new default, for next time.

The default default values (the defaults after boot) are chosen from the available hardware on the local machine and the declared tape servers at the site.

For many parameters, a "null value" can be specified, which applies whatever the host chooses by default.

The following is a typical prompt:

```
Type tape host or spec (default (Return) = Local: 0, reel=joe, den=1600):
```

The defaults in this example are tape on the local machine, drive 0 (TD-80 half-inch, industry compatible tape), a reel named "joe", and density of 1600 bits per inch.

3.1.7.2 Tape Spec Parameters

The following are the supported changeable parameters with allowable synonyms:

- host (machine) The *tape host*, that is, the machine you have mounted or intend to mount the tape on. The string "local" is acceptable to specify the local machine. Any name known to the namespace system may be used to specify a machine.
- device (dev, unit) The tape drive on that machine that will be used. For Lisp Machines, numeric values of 0 to 7 specify the designated "reel-to-reel" (half-inch industry-compatible) drives, when present, and "cart" specifies 3600 cartridge tape. For other tape hosts, you must know the machine's drive specification format, if the tape host does not choose the drive you expect by default.
- reel (volume, vol) The name of the tape reel. Whether or not this parameter is necessary depends on both the program and the tape host being used. For instance, the distribution dumper requires a reel name to compute the name of the log file. Few other Lisp Machine programs do. Some tape hosts (none currently supported) might require a reel name in order to inform their operators to locate and mount the reel you want.
- density (dens, den) The tape density in bits per inch. Some tape controllers on some machines utilize this information, others ignore it. The default, if the program does not specify a density, is 1600 for non-cartridge tape.
- number-of-buffers (buffers, bufs, buffs, n-buffers, n-bufs, n-buffs) This is a performance-control parameter only applicable to local tape on a Lisp Machine. It controls the number of read-ahead/write-behind buffers used by both the industry-compatible tape stream implementations and the 3600 cartridge tape stream implementation.
- record-length (length, len, reflen, recsiz) The length of tape records, in bytes. Normally, programs specify this parameter explicitly, such that another program with which it is communicating via tape, can read what it has written or write

what the other program can read. Nevertheless, it is sometimes useful to change this parameter at tape spec merging time, in order to experiment with unknown tapes, and so forth. This parameter should not be altered casually.

granularity (gran) This specifies a number n , such that all tapes written by the stream being created are padded with zeros (the pad character is program-changeable) to be multiples of n in length. It is ignored in the case of cartridge tape. The default for this parameter depends on the tape host and its stream and hardware. For 3600 industry-compatible tape, it is 1. Other machines have various defaults, often not 1. Although supplied Lisp Machine tape applications specify a granularity of 4, it would be wrong for **tape:make-stream** to prohibit you from writing records of any length if you have a tape host not so restricted. However, tape hosts can, in general, not read tapes at granularities not multiples of their own granularity, and this causes great problems moving tapes from site to site. This parameter can be used to adjust granularity when writing tape, to help remedy this problem in some cases.

minimum-record-length (minimum, minimum-record, minimum-length, minrec, minreclen)

This specifies a number n , such that all tapes written by the stream being created will be padded with zeros (the pad character is program-changeable) to be at least n bytes in length. The default is 64. As with granularity, the goal here is to ensure compatibility between systems. Some can write records shorter than others can read, and some can even write records shorter than they themselves can read. Again, this parameter is usually managed by tape-using programs, but might need to be adjusted by hand in some cases where writing records smaller than 64 bytes is required, but modifying the program is impossible or undesirable.

3.1.7.3 How Tape Specs Are Merged

When **tape:make-stream** prompts you with a default tape spec, you type a (perhaps empty) tape spec. It merges the two to produce the final tape spec.

The tape spec you type can specify one of three types of values for each parameter:

- | | |
|----------------|--|
| Explicit value | If you specify the value of some parameter, the value you specified, assuming it is legitimate, is used. |
| Merge default | By not saying anything at all about a parameter, the default from the default shown to you is used. |
| Null value | By supplying a <i>null value</i> for a parameter, the tape host uses its own default. This is how you cancel a displayed default value |

without supplying an explicit value. A side-effect of this is that there is not an explicit value for this parameter in the next default. For the **host** parameter, a null value is the same as not saying anything.

Here are the merging rules:

Each explicit or null value given in the typed tape spec replaces the corresponding element of the default to give the new tape spec. Components not specified are carried over from the default.

However, if the typed **host** is not the **host** in the default, and the typed tape spec does not specify a **device**, the **device** from the default will *not* be used, but a null value assumed.

See the section "Tape Spec Merging Examples", page 30.

3.1.7.4 Tape Spec Syntax

The general form of a tape spec is a comma-separated list of equal-sign pairs:

```
host=LM-6, reel=foo, density=800
```

The order of parameters is not significant. Each *parameter specification* is of the form

```
parameter = value
```

The specifications are separated by commas. Spaces and tabs around the equal sign or commas are not significant. Spaces and tabs in values are considered significant parts of the value.

Values that contain commas, equal signs, quotation marks (" "), or colons can be quoted with quotation marks to preserve their integrity during parsing. Reader string slashification is operative in such strings.

A null value can be specified by any of the following:

```
reel =
reel = ()
reel = ""
```

In the first example, the equal sign is followed by the end of the line.

Some special allowances are made for specification of **host** and **device** in a less-general syntax, more like file pathname syntax.

- A tape spec consisting of only a single token is interpreted as specifying the **host** parameter by that token. The following are valid, equivalent tape specs:

```
host = lm-6
LM-6
Lisp-Machine-6
"LM-6"
```

- If a token followed by a colon, or even an empty token followed by a colon appears where a parameter specification is expected, that token is interpreted as specifying the value of the **host** parameter. A colon thus used "replaces" the comma that would have been there (although the syntax "lm-6:" is permitted for "host=lm6"). The following are valid, equivalent tape specs:

```
lm-6: reel=foo, device = "mta0:"
reel= foo, host= lm-6, device = "mta0:"
reel=foo, lm-6: device = "mta0"
reel=foo, lm-6: device = mta0
```

- A null token before a colon specifies the merge default for the **host** parameter. The utility of this is explained below.
- A single token appearing where a parameter specification is expected, that is, followed by the end of the spec or a comma, is interpreted as specifying a **device**.

Thus, the following are equivalent:

```
device=cart
:cart
```

The following, too, are equivalent:

```
LM-6: cart, reel = foo
LM-6: reel = foo, device = cart
cart, reel = foo, host = LM-6
device = cart, reel = foo, host = LM-6
```

Note again that a tape spec consisting of a single token, followed by either colon or no delimiter at all, is (in the latter case) a special case, and designates the value of the **host** parameter, not the **device** parameter.

The syntax of tape specs is quite general and flexible. It is also, in most common cases, trivial. Far and away, the most common answer to the **tape:make-stream** prompt is simply RETURN.

In most other cases, the only relevant information that **tape:make-stream** is lacking is the name of the host, which can simply be typed alone as a valid tape spec. It is for this reason that the prompt says the following:

```
Type tape host or spec
```

The next most common case is specifying a device. A single leading colon differentiates this from a host. The following example specifies both host and device:

```
LM-6: cart
```

3.1.7.5 Tape Spec Merging Examples

The following are examples of merging tape specs. It should be noted that tape specs are abstract quantities, not strings. The merging is performed on the abstract tape specs, not on strings.

old default:	Local: 0
spec:	:cart
merged result:	Local: cart
old default:	Quabbin: 0
spec:	den=3200
merged result:	Quabbin: 0, den=3200
old default:	Vixen: 0
spec:	Stony
merged result:	Stony
old default:	Quabbin: 0, den=3200
spec:	den=
merged result:	Quabbin: 0

3.2 Carry-tape System

The carry-tape system provides a means of dumping selected files or sets of files to magnetic tape (cartridge or industry-compatible) and loading them at a later time, probably at a different site. Files can be dumped from any host or set of hosts, and reloaded to any place on any host.

The carry-tape system provides a standard, system-independent interchange medium for exchanging single programs and files between sites. It is meant to fill in a gap between backup dumpers and the distribution tape system. It requires no files or declarative forms to be prepared in advance.

The carry-tape system has three components:

- The carry-tape dumper
- The carry-tape loader
- The carry-tape lister

3.2.1 The Carry-tape Dumper

tape:carry-dump *file-or-files* &key *tape-host* *density* *reel* (*report* *t*) *Function*

Dumps a file or set of files to a carry-tape. Any type of file can be dumped. Character files are dumped and reloaded using the Lisp Machine character set as an interchange medium. Binary files are dumped and reloaded with the proper byte size as long as either of the following is true:

- The file is of one of the system's known canonical types.
- The operating system on which the file resides knows and can supply the byte size.

file-or-files a pathname, filespec, or list of pathnames and/or filespecs. Wildcard pathnames or filespecs may be used. Recursive ("accordion") wildcards may be used to dump subtrees on those hosts that support them.

tape-host a host object or the name of a host object to use for tape access. **:local** specifies the local tape drive. If not specified, the standard tape host prompt and defaulting mechanism is used.

density a fixnum, specifying tape density, which may be used when the applicable default is not appropriate.

reel can be a string, specifying tape reel name for tape servers that need this information (none of the currently supported ones do).

report tells the carry-tape dumper to report its progress as it dumps files. A value of **nil** tells it not to. The default is to report to **standard-output**. Any value besides **nil** or **t** is expected to be a stream to which the reports will be written.

Currently, carry dumps must fit on one tape.

The carry-tapedumper commences by finding out all available information about the files to be dumped, verifying their existence. It then asks for confirmation, and proceeds to dump all the files specified without intervention.

```
(tape:carry-dump "swanee:>minerals>*.d*")
To be dumped:
swanee:>minerals>*.d*: 7 files
Is this right? (Y or N) Yes.
Type name of tape host (default (CR) = POINTER): scrc
Tape mounted on drive mta0:.
Dumping swanee:>minerals>abel.data.3 (5-bit bytes)
Dumping swanee:>minerals>abel.directory.7
.....
End of dump.
```

3.2.2 The Carry-tape Loader

The carry-tape loader loads files from a carry-tape. The carry system makes no attempt to copy any file properties, including author and creation date. The carry system copies only file contents, and the loader provides reasonable defaults for the target file name.

tape:carry-load	<i>&key host density reel (report t)</i>	<i>Function</i>
<i>host</i>		a host object or the name of a host object to use for tape access. :local specifies the local tape drive. If not specified, the standard tape host prompt and defaulting mechanism is used.
<i>density</i>		a fixnum, specifying tape density, which may be used when the applicable default is not appropriate.
<i>reel</i>		can be a string, specifying tape reel name for tape servers that need this information (none of the currently supported ones do).
<i>report</i>		tells the carry-tape loader to report its progress as it loads files. A value of nil tells it not to. The default is to report to standard-output . Any value besides nil or t is expected to be a stream, to which the reports will be written.

These arguments are rarely needed.

The carry-tape loader begins its operation by reporting the filespecs given to the dumper, and asks if you wish to load all of the files dumped. If only one filespec or pathname was given, it is assumed that you want to load it all, and no question is asked:

```
(tape:carry-load)
Type name of tape host (default (CR) = APS0): beta
Tape mounted on drive mta0:.
Carry dump made by DCF.
Dump taken at 11/13/82 09:05:22.
Dumped on machine EAGLE.
Dumped: e:>trees>apple.orchard
```

The set of files dumped as a result of each filespec given to the dumper is called a *group*. If many groups were dumped, the loader lists the filespecs of each group at the start of its operation, and asks for instructions about which groups are to be loaded (selectively) and which groups are to be skipped:

```
The following groups of files were dumped:
e:>trees>apple.orchard
e:>animals>whales>tails.tales
e:>baseball>runs>foul.*
-----
Load all these files? (ABORT to get out) (Y, Q, or M)
```

The possible responses are:

- Y Ignore distinctions of group, and proceed as described below.
- Q Query about each individual group, and proceed as below for those groups that are selected for loading.
- M Same as Q, but present a multiple-choice menu instead of querying for each group.

If you do not want to load anything, ABORT will work at any time.

The carry-tape loader can either query for the target location of each file to be loaded, or proceed in semi-automatic mode, in which the host and directory from which each file was dumped are used as a key to target loading of subsequent files from that host and directory. The name, type, and version of each file to be loaded are developed automatically from the name, type, and version of the file that was dumped, by means of the same mechanism used by ordinary file copying.

The normal action of the carry-tape loader is to query for each file, with a query of the following form:

```
Load SWANEE:>minerals>rock5.data.6 into BULLWINKLE:/usr2/jones/rock5.data?
(Y, N, O, or A)?
```

The following responses apply:

- Y, SPACE Load the file into the place specified. The host and directory shown remain the default target directory for all files from this host and directory at the sending site.
- N Do not load this file at all. The host and directory shown remain

the default target directory for all files from this host and directory at the sending site, in spite of this.

- O** Prompt for another place in which to put this file. The host and directory into which this file is then loaded become the default for all subsequent files from the same host and directory at the sending site. You are queried again at the time such files are encountered.
- A** Load the file into the place specified. All further files from the same host and directory at the sending site are then automatically loaded into the same host and directory as this file without querying you.

3.2.3 The Carry-tape Lister

The carry-tape lister describes what is on a carry-tape. Once started, it does not interact in any way.

tape:carry-list &key <i>host density reel (report t)</i>	<i>Function</i>
<i>host</i>	a host object or the name of a host object to use for tape access. :local specifies the local tape drive. If not specified, the standard tape host prompt and defaulting mechanism is used.
<i>density</i>	a fixnum, specifying tape density, which may be used when the applicable default is not appropriate.
<i>reel</i>	can be a string, specifying tape reel name for tape servers that need this information (none of the currently supported ones do).
<i>report</i>	tells the carry-tape lister to report its progress as it lists files. A value of nil tells it not to. The default is to report to standard-output . Any value besides nil or t is expected to be a stream, to which the reports will be written.

These arguments are rarely needed.

4. Multiple Partitions

The Lisp Machine File System (LMFS) allows the use of multiple partitions residing on one or more disk drives. The selection of partitions to be used by LMFS is determined by a database called the *file system partition table* (FSPT).

The FSPT is a FEP file named >fspt.fspt on the boot drive. The FSPT is optional: If not present, LMFS behaves as previously and uses lmfs.file on the FEP boot drive. The FSPT is a simple character database containing the actual pathnames (in the FEP file system) of the partitions to be used for file system access.

Each partition in the file system knows how many partitions make up the file system. Only the FSPT, which is used only at LMFS startup time, indicates the locations of these partitions. That is, the file system databases in the actual partitions do not contain drive and partition numbers or FEP pathnames. Thus, when LMFS is down, partitions can be moved around using Copy File (m-x); as long as the FSPT is edited to indicate their new locations, LMFS comes up (when required) using the moved partitions.

The FSPT is edited only to move partitions around or to add a partition. When you add partitions to the file system, the file system automatically rewrites the FSPT database to include the locations of new partitions.

Do not delete file partitions from your LMFS. Each LMFS partition contains pointers to all other file partitions in the LMFS. Deleting a file partition leaves the other partitions with pointers to a nonexistent file.

If you want to reduce the size of your LMFS, you must completely backup your LMFS, delete the entire existing LMFS and create a new one. The user files can then be restored into this new LMFS from the backup tapes.

Never delete a FEP file that is entered in the FSPT. If you do so, even if you remove the corresponding FSPT entry, the file system will not be able to initialize itself properly and will not be able to be brought up. This means that your file system will be irrevocably inconsistent and inaccessible.

4.1 Adding a Partition to LMFS

Partitions can be added to LMFS with [Local FS Maint] on the File System Editor menu. Select this item to get a menu of file-system maintenance operations. The [Initialize (R)] command yields a menu of initialization options, which offers [New File System] and [Auxiliary Partition] as a choice. [New File System] is similar to [Initialize (L)]; it initializes a partition to be the basis of a file system.

When you add a new partition or a partition on another disk, the disk should be free of errors and properly initialized and formatted, and the partition should exist.

To add another partition, use [Auxiliary Partition]. Enter the pathname of the FEP file to be used as the new partition. The default presented, which is correct for [New File System], is never correct for adding a partition. Then use [Do It]. The system then performs much verification and error checking, roughly as much as when initializing a new partition. It must not be interrupted while performing these actions. When finished, it adds the partition and edits the FSPT automatically.

5. File System Maintenance Program

The File System Maintenance Program includes the File System Editor as well as other commands for manipulating backup tapes, running the salvager, and doing other local file system maintenance operations. See the section "File System Editor" in *Reference Guide to Streams, Files, and I/O*. Although it includes commands for dealing with remote file systems, including the interface to the File System Editor, the File System Maintenance program is intended for manipulating the local file system, and is the only place most where of its functionality is available.

The program uses a frame with a command menu pane at the top and a large pane beneath it. Commands are given by clicking on the command menu pane. The large pane can either be a *Lisp Interaction Window* or a *File System Editor* window, depending on whether or not you are in the File System Editor. When it is in the former state, it is a Lisp Listener, and you can type Lisp forms at it and have them evaluated and printed. When it is in the latter state, you are in the File System Editor and can click on the pane to get File System Editor commands.

5.1 Free Records

The basic unit of allocation in the File System is the *record*. A record is 1152 32-bit words, or four disk blocks. Each file system object is made from an integral number of records. At any time, each record is *in use* (representing an existing file system object) or *free* (not representing anything and free to be used in new objects). When the file system needs to find a new free block to create or grow an object, it does not search through the records looking for a free one, because that would require many disk operations and be very slow. Instead, the file system keeps a redundant data structure called the *free record map*, kept in several blocks in a known location in the file system partition. The map has one bit for each record in the file system; this bit marks whether the record is in use or is free. The file system can find a free record quickly by examining this map.

If the file system crashes, or something else goes wrong, the contents of the free record map can become inconsistent with the contents of the file system itself. For each record, there are two different errors that might happen. The first error is that the record might be in use, representing some of an object, but be marked as "free" in the map; the second error is that the record might be free, but be marked as "in use" in the map. The first error is much worse than the second. If the first error happens, then the file system might use the record for a new object even though it is currently representing some existing object, which could destroy the existing object. The second error is less severe: the record simply is not allocated even though it could be. Such a record is said to be *lost*.

The file system is written so that a crash can only create the second kind of error and not the first. While the file system is operating, it maintains a *free buffer* in its data structures in virtual memory. The free buffer is a pool of records that are not actually in use, but that are marked as being in use in the free map on the disk. When it needs to allocate a record, it draws on one of these; when it frees up a record, it adds the record to this buffer. When the buffer gets too big, some records are removed from the buffer and marked as "free" in the map on the disk; when the buffer runs low, some more records are marked as "in use" in the map on the disk, and are added to the buffer. So, if the machine is cold booted, or the file system crashes, the records that are in the buffer will be lost, but no errors of the first kind will be created. The size of the buffer is maintained at about 30 records, so each crash will lose 30 records. Logging out of the machine, or using the [Flush Free Buffer] command, will flush the entire free buffer and mark the records as "free" in the map on the disk. After the buffer has been flushed, you can cold boot the machine without losing any blocks.

Lost records can be found again by the salvager. See the section "Salvager", page 38.

The [Free Records (L)] command first types out a line for each block of the file map, telling you which records are covered by that block, the number of such records, and how many are marked as free. [Free Records (L)] also tells you how many free records (marked as "in use" in the map) are in the free buffer, and finally types out a grand total of the number of free, used, and total records in the file system.

To find out how many records are actually in use, click on [Free Records (M)] to prepare a printable report of record use throughout the file system. This has to pass over every object in the file system, and so it takes some time, especially on large file systems. The discrepancy between the answer of this function and the answer of [Free Records (L)] tells you how many lost records there are; if there are a lot, you might want to run the salvager.

5.2 Salvager

The *salvager* is a program that reads every record of the file system and finds and fixes inconsistencies and errors. It can fix two classes of problems. First of all, it can see which records are in use and which are free, and set the free record map to correctly reflect the current state of the file system. This is how lost records are recovered. Secondly, it can find objects that are stored in a file system partition but are not part of any directory. Such objects are called *orphans*; they exist only if some problem has occurred, such as a file system crash during the creation of a file, or an unanticipated directory failure of some sort. The salvager can find such objects and put them back into the directory hierarchy (*repatriate* them).

To run the salvager, give the [Salvage] command. If you have a local file system of

multiple partitions (occupying multiple FEP files), you will be presented with a menu of partitions to process. The normal default is to process all partitions. In any case, you will be presented with a second menu, asking whether or not you wish to repatriate orphans, and whether or not you wish to "check records" after the salvage. (For a description of the [Check Records] command: See the section "File System Maintenance Commands", page 39.) Do *not* attempt to repatriate if you have chosen *not* to salvage any partitions.

The salvager always reconstructs the free record maps. The salvager takes about two minutes per thousand records of file partition.

When the salvager is repatriating an orphan and it cannot find the directory in which the orphan is supposed to reside, it creates a new directory as an inferior of the directory >repatriations, with a name like lost-1 or lost-2. After a repatriating salvager run, you should examine this directory. When the salvager repatriates an object, it types out a message saying that it did so. (If you walk away from the machine while the salvager is running, one of these messages might cause a MORE pause; you might want to disable MORE pauses before you start the salvager.)

Note: the salvager always considers storage occupied by orphans to be "in use" for purposes of the free record map, even if it is not repatriating the orphan. If many orphans exist, they can use up a great deal of disk space, but normally they do not occur.

5.3 File System Maintenance Commands

The following list explains all of the commands in the program. Some commands are described as "typing out" certain information. If the large pane is a *Lisp Interaction Window*, the information is simply printed on that window; if it is a *File System Editor*, then the information is typed out over the file system editor information using a "typeout window". You can flush the typeout and restore the display of the File System Editor by typing any character or by clicking on the [Flush Typeout] menu item.

- | | |
|---------------------|---|
| [Tree edit root] | Enter the File System Editor, using the root directory as the base directory. See the section "File System Editor" in <i>Reference Guide to Streams, Files, and I/O</i> . This puts the large pane into the <i>File System Editor</i> state. |
| [Tree edit any] | Enter the File System Editor; you will be prompted for the name of the base directory. See the section "File System Editor" in <i>Reference Guide to Streams, Files, and I/O</i> . This puts the large pane into the <i>File System Editor</i> state. |
| [Tree edit Homedir] | Enter the File System Editor, using your home directory as the base directory. See the section "File System Editor" in |

- Reference Guide to Streams, Files, and I/O*. This puts the large pane into the *File System Editor* state.
- [Free records] Type out information about the number of free records in the local file system. The interesting part is the last line, which tells you how many records are marked as free and how many are marked as used, and the sum of these numbers, which is the total number of records in the file system.
- [Free Records (M)] prepares a directory-by-directory usage report of record use, indicating how many records are in use by files in each directory. You will be prompted for the name of a file in which to place the report. See the section "Free Records: File System Maintenance Program", page 37.
- [Salvage] Run the salvager. See the section "Salvager", page 38.
- [Reload/Retrieve] Invoke the reloader to recover files off a backup tape, or list or verify the contents of a backup tape. See the section "Reloading and Retrieving", page 10. [Reload/Retrieve (M)] searches for backup copies of a file. See the section "Finding Backup Copies of Files", page 12.
- [Incremental Dump] Do an incremental dump of the local root directory. You will be offered a menu to adjust all parameters. See the section "Dumping, Reloading, and Retrieving", page 5.
- [Complete Dump] Do a complete dump of the local root directory. You will be offered a menu to adjust all parameters. See the section "Dumping, Reloading, and Retrieving", page 5.
- [List FEP FS Root] List the boot unit FEP file system's root directory. See the function **print-disk-label**, page 45.
- [Print Herald] Print information about the running system.
- [Flush Free Buffer] Write the internal pool of free disk records back to the disk. This happens when you log out. After doing this, you can cold boot without losing records. See the section "Free Records: File System Maintenance Program", page 37.
- [Server Errors] Print out all the error messages associated with errors encountered by the file server. When such errors occur, you get a message that begins as follows:
- [File Server got an error: ...]
- The message contains descriptive information about the error.
- [Server Shutdown] Shut down file servers at future time, or reschedule or cancel a shutdown.
- This command lets you shut down a file server cleanly. You would run this command only on a Lisp Machine that was

acting as a file server for other users. Clicking on [Server Shutdown (L)] means that you plan to shut down the file system soon. You are asked for a short message to be sent to people using the file server, which you can use to explain why it is being shut down and when it will return. You are also asked when you want the shutdown to take place; the default is five minutes. All users of the file system are sent periodic messages warning them that the server is going to be shut down. Finally, when the time comes, all Chaosnet servers on the machine are closed, and creation of new servers is disabled. At this point servers are shut down, and the machine can be cold booted or whatever else you want to do. While the shutdown is "in progress" (the messages are being sent), you can cancel the shutdown by clicking on [Server Shutdown (M)] or reschedule it by clicking on [Server Shutdown (R)].

[Server Shutdown] shuts down the network server; it does not affect the local operation of the file system itself. It shuts down all servers, not just file servers, since anything that requires the file servers to be shut down also requires that all servers be shut down.

- [Maintenance] Put up a menu of potentially dangerous operations, intended to be used only by people who are thoroughly familiar with the file system, just below the main command menu. These operations are described below.
- [Lisp Window] Put the large pane back into the *Lisp Interaction Pane* state.
- [Flush Typeout] When the large pane is in the *File System Editor* state, and you use one of the above command that "types out" information, then the information will appear on top of the File System Editor window, and you will be told "Type any char to flush:". You can use this command to flush typeout and restore the File System Editor window, too; the command is useful because you can give it without removing your hand from the mouse. You can also use this command to proceed from MORE pauses.
- [HELP] Type out general information about the File System Maintenance program.
- [QUIT] Bury the frame, possibly reselecting the window you were using before you started using the File System Maintenance program.

The following items appear in the *Maintenance Operations* menu. They are intended only for users who are thoroughly familiar with the operations of the file system; misuse of these operations can destroy or damage the file system.

- [Initialize] Create a new file system. This operation asks you several questions and prints out information to make sure you really want to initialize the FEP file that would contain the file system. These verifications are to ensure that you do not accidentally destroy any previous file system residing there. [Initialize] takes about a minute for each four thousand records (a record is four 256-word disk blocks). [Initialize] queries about FEP file size.
- [Initialize (M)] presents a menu of initialization options. This is how you can add new FEP files to the running file system. For a description of this operation: See the section "Adding a Partition to LMFS", page 35.
- Auxiliary partitions should never be initialized. The partitions are automatically initialized by the system when they are created.
- [Close All Files] Call **fs:close-all-files**. This has nothing to do with the Lisp Machine File System as such; it closes any files, locally or remotely, on any file system to which the machine is connected. This is occasionally useful for cleaning up after problems, but you can cause new problems for any programs in the machine that are validly using files at the time.
- [Active Structure Edit] Edit the active file system data structure. This displays "active" internal data structure as a scroll window, and is intended for use by those debugging local file system problems.
- [Expunge All FS] Expunge all directories on the local file system. Displays a message about how much space was recovered.
- [Check Records] Check each record in the local file system for consistency, and notify you about any problems. (This is also available from the salvager). This option scans the hierarchy, going through the directories, making sure that each directory entry really describes a file that agrees with it, and that each record of each file is validly identified as a part of that file.

6. Creating More Room on the Local Disk

There are two file systems available on the Symbolics computer: the Lisp Machine File System (LMFS) and the FEP File System (FEP FS). LMFS is a general-purpose, highly flexible file system, suitable for everyday use. Currently, only the Symbolics processor understands how to operate on LMFS files. The FEP FS is a simple, basic file system that both the Symbolics computer and front-end processors understand how to access. The FEP FS is used mainly to store world loads, microcode loads, paging files, boot files, and file system partitions that LMFS uses to store its structure and data. The FEP FS is not a good place for users to store their files; that is what LMFS is for.

Sometimes the Save World command or the **disk-save** or **si:receive-band** function might inform you that you have run out of FEP file system space. For systems with 167-Mbyte or more of storage, you should delete and expunge old, unneeded world loads, and then resume from the Save World/**disk-save** "out of room" error or retry the **si:receive-band** operation. You should not delete any world loads from a 140-Mbyte system. See the section "Instructions for Managing Disk Space on the 3640", page 55.

It is wise to keep a large (25K-30K), noncritical world load on the Symbolics computer's disk, where it is available for the FEP Disk Restore command to use in case all world loads become nonfunctional.

Sometimes, writing a file out to a Lisp Machine File System (LMFS) produces an "out of room" error. This means that the present allocation of that particular LMFS is not large enough to accommodate your request for space. It might help to expunge directories with deleted files in them to remove unneeded versions of files, using the Zmacs command **Dired** (**m-x**).

If you still do not have enough space after you have deleted and expunged the unnecessary files, you might consider creating an auxiliary file partition. You should only consider doing so on systems that have at least 280 Mbytes of storage. There is no room in the FEP file system to allocate an auxiliary file partition for 140-Mbyte systems, and allocating an auxiliary file partition on a 167-Mbyte system might inhibit the creation of large world loads.

Even for 280-Mbyte systems, you are trading off world load space for file space when you create auxiliary partitions. Be sure to reserve enough FEP file system space for two large world loads (about 60K blocks total): the world you are currently running from and a spare world load for the FEP Disk Restore command to use.

For details on how to create auxiliary file partitions: See the section "LMFS Multiple Partitions", page 35. Once you have created an auxiliary file partition, you should never delete it. Deleting it would lose all the data contained in that partition and make the entire Lisp Machine File System unusable.

If you run out of room while writing a LMFS file and then create a new partition to increase the LMFS space, you cannot resume the file operation that failed. Instead, you must ABORT that operation and then retry the operation.

6.1 Allocating Extra Paging Space

Programs that use large amounts of virtual memory might require you to allocate additional paging space, to perform better or to perform at all. Only systems with at least 280 Mbytes of disk storage have enough room to permit additional paging files to be allocated without adversely affecting the maintenance of worlds on the machine. In order to add a paging file to your virtual memory set, you must first create a FEP file. Below is a function that creates a FEP file of a given length.

```
(defun create-fep-file (name length)
  (with-open-file (stream name :direction :block :if-exists :error)
    (send stream :allocate length)))
```

The code below creates an additional 20K-block paging file on unit 0, using the above function:

```
(create-fep-file "fep0:>page1.page" 20000.)
```

After creating the extra paging file, any boot files should be modified to use this new paging file. A typical boot file might look something like this:

```
clear machine
load microcode >tmc5-mic.mic.319
load world >Dist-6-0.load
set chaos 401
start
```

After creating the new paging file, edit your boot file to look something like this:

```
clear machine
load microcode >tmc5-mic.mic.319
load world >Dist-6-0.load
clear paging
add paging >Page.page
add paging >Page1.page
set chaos 401
start
```

It is safe to delete extra paging files, but only if they are not in active use. You cannot change a paging file that is being used without booting. Anytime you change the paging area you have set up, first boot without adding the paging file to be deleted. Be sure to cold boot by hand, and *do not* type the Add Paging command for the extra paging partition you intend to delete.

7. Saving New Versions of World Loads

7.1 Concepts

The **make-system** and **load-patches** functions load software into the Lisp world. This takes time; it is wasteful for everyone to sit through this loading of software every time the software is to be used. Usually someone loads up software into a Lisp world and then saves away the whole Lisp world in a FEP file on a disk. This section explains how to do this and other things.

For information on the FEP file system: See the section "FEP File System Overview" in *User's Guide to Symbolics Computers*.

7.2 Manipulating the Label

print-disk-label &optional (*unit* **si:*boot-unit***) (*stream* **standard-output**) *Function*

Prints onto *stream* a description of the FEP files on *unit*. *unit* can be one of the following:

- An integer smaller than 20., interpreted as a disk unit number on the local host.
- An integer larger than 19., interpreted as the Chaosnet address of a remote host. Prints the contents of unit 0 on that host.
- A symbol, interpreted as the name of a remote host. Prints the contents of unit 0 on that host.
- A string of the form "*host|unit*", where *host* is the name or Chaosnet address of a remote host and *unit* is an integer representing a disk unit number on that host.

unit defaults to the value of **si:*boot-unit***, usually 0.

print-disk-label first prints an estimate of the number of free blocks and the proportion of blocks used on *unit*. It then prints a summary of the files on *unit*, one line per file. For each file, it prints the file name, the length in blocks and in bytes, the byte size, the creation date, the comment, and the author.

7.3 Updating Software

Of the procedures described in this section, the most common one is to use **load-and-save-patches**.

load-and-save-patches &rest *keyword-args* *Function*

load-and-save-patches first disables network services and MORE processing and then loads any patches that need to be loaded and any new versions of the site files, calling **load-patches** with arguments of **:noselective** and any other keywords provided as *keyword-args*. Valid *keyword-args* are:

- :verbose** **:verbose** prints an explanation of what is being done. This is the default.
- :selective** For each patch **:selective** displays the patch comment and then asks you whether or not to load the patches. The choices are Y, N, P, or H: yes, no, proceed, or highest. Answering P turns off selective mode for any remaining patches to the current system. H means highest patch number to load. If you do not specify a limit, it loads all patches from the present level for a given system.
- :silent** **:silent** turns off both **:selective** and **:verbose**. In **:silent** mode all necessary patches are loaded without printing anything and without querying the user.
- :nowarn** **:nowarn** suppresses any warnings generated while a patch file is being loaded, such as that produced when a symbol belonging to one package is being defined by a file belonging to a different package. It also suppresses the subsequent query to the user. **:nowarn** does not imply **:noselective**.

If no one is logged in, it logs in anonymously. If any patches have been loaded, **load-and-save-patches** prompts for the name of a FEP file in which to save the world load and then calls **disk-save** to actually save the resulting world load. If no patches have been loaded, it restores network services to their state before **load-and-save-patches** was called, and logs out if it has logged in anonymously.

Call **load-and-save-patches** *before* you log in in order to avoid putting the contents of your init file into the saved world load.

disk-save &optional *destination-file* *Function*

Saves out the current Lisp environment into a FEP world-load file.

```
(disk-save "purple")
(disk-save "fep0:>purple.load")
(disk-save "fep1:>heavy.load")
```

destination-file is the pathname of the FEP world-load file in which the Lisp environment is to be saved. It defaults to a file on FEP: (the boot unit) with a file type of .load and a file name of *Release-major-minor* (for released systems) or *System-major-minor* (for unreleased systems).

disk-save asks for confirmation only if the file to be written is the same file from which the currently running world was booted.

When the FEP file system does not have enough room to save the world-load file, **disk-save** tells you how many more blocks you need and offers a choice of actions that you can take on the root directory of the FEP file system of the unit on which you tried to save:

```
Estimated size is N blocks.
Not enough room in FEP filesystem, need at least N more blocks.
Run Dired, Expunge directory, List directory, or Selectively
delete load files (D, E, L, or S)?
```

- D Runs Dired on the directory. If you try to run Dired, **disk-save** warns:

Running Dired will substantially increase the size of the saved world. You should, therefore, boot again before saving if you run it. Run Dired anyway? (Y or N)
- E Expunges the directory.
- L Lists all files in the directory.
- S Selectively deletes load files. It goes into a loop asking about each ".load" file in directory (except the file you are currently running). It asks whether to delete each one, expunging after each successful deletion until you have enough room. This is probably the only option you need to use.

If a world-load file to be written already exists, **disk-save** offers a choice of superseding the file, overwriting the file, or providing a new pathname.

disk-save displays the herald for the world to be saved. It asks whether you want to add an additional comment to the herald or change the comment if already present. The value of the variable **si:system-additional-info** is set to the string you type, and **print-herald** displays this string in parentheses at the end of the first line of the herald.

disk-save constructs a title for the world load based on the comment and

versions of the systems loaded in the environment; it defaults to the value returned by (**si:system-version-info t**), but you can supply an alternate title. The title is a property of the world-load file; **print-disk-label** displays this title.

disk-save then displays a few messages telling you what it is doing: It runs the initializations in the **si:before-cold-initialization-list**, logs out, and runs the initializations in the **si:system-shutdown-initialization-list** (see below). Then the machine seems to act as it does when it cold boots, although actually it is copying from the paging file to a FEP file rather than the other way around. When it finishes it displays the herald message as if cold booting had just completed.

(Note that the entries on the **system-shutdown** list should all be for subsystems that are required for almost anything else to run. Currently there are entries for the network, the TIME system, and the Lisp Machine file system. User programs should add themselves to the **before-cold** list rather than to **system-shutdown**.)

disk-save offers to update the FEP boot file to load the world that has just been saved.

Do not save patched systems that have had the editor or the compiler run. This works, but it makes the saved system a lot bigger. You should try to do as little as possible between the time you cold boot and the time you save the world load, in order to produce a clean saved environment.

si:login-history

Variable

The value of **si:login-history** is a list of entries, one for each person who has logged into this world since it was created. This makes it possible to tell who **disk-saved** a band with something broken in it. Each entry is a list of the user ID, the host logged into, the Lisp Machine on which the world was being executed, and the date and time.

si:print-login-history prints this history in a legible way.

si:print-login-history &optional (*whole-history* **si:login-history**)

Function

Prints one line for each time the **login** function has been called in this world load. Each line contains the name of the user that logged in, the name of the machine on which the world load was running at that time, and the date and time. If you cold boot, log in, and then call **si:print-login-history**, the last line refers to your own login and all previous lines refer to logins that were done before running **disk-save**.

This information is useful to determine how many times a world load has been disk-saved, on what machines it was disk-saved, and who disk-saved it.

The first couple of lines do not contain any date or time, because they were

made during the initial construction of the world load before it found out the current time. Names of users at other sites that are not in the local site's namespace search list are qualified with the site's namespace name and a vertical bar. The user SCRC|LISP-MACHINE is the dummy user used by **si:login-to-sys-host** at SCRC, the site where new world loads are created.

7.4 Installing New Software

Install microcode with **si:install-microcode**.

si:install-microcode *from-file-or-version* &optional *to-file-or-version* *Function*
boot-file-to-update

Installs microcode from a system file into a file in the FEP file system.

from-file-or-version is a microcode version number (in decimal). The file resides in the logical directory *sys:l-ucode*;

to-file-or-version rarely needs to be supplied. It defaults to a file on FEP:> (the root directory of the boot disk) whose name is based on the microcode name and version. If supplied, *to-file-or-version* is either a pathname (string) of a file on FEP:>, or an integer *n*, which stands for the file TMC5-MIC.MIC.*n* on FEP:>.

The logical directory *sys:l-ucode*; includes multiple types of microcode for each version number. The correct microcode to install depends upon the particular hardware configuration of your machine. When your machine is shipped, the default microcode filename is correct, but if your machine is upgraded (for example, an FPA board is installed) you might need to override the default used by **si:install-microcode** to get the correct type for your configuration. Below is an example of how you would get the microcode for a 3600 running 6.0, with no console upgrade but an FPA board installed:

```
(si:install-microcode "tmc5-fpa-mic.mic.319")
```

The correct microcode types for each system and hardware configuration are shown below. The names in this table omit the suffix *mic.n* that you must include to indicate the version of the required microcode. The version number must be followed by a period. Microcode version 319. is required for Release 6.0.

3600

	tmc5	ifu
No FPA	tmc5-mic.	ifu-mic.
FPA	tmc5-fpa-mic.	ifu-fpa-mic.

3670/3600 with console upgrade

	tmc5	ifu
--	-------------	------------

No FPA	tmc5-io4-mic.	ifu-io4-mic.
FPA	tmc5-io4-fpa-mic.	ifu-io4-fpa-mic.
3640		
	tmc5	ifu
No FPA	tmc5-io4-st506-mic.	ifu-io4-st506-mic.
FPA	tmc5-io4-st506-fpa-mic.	ifu-io4-st506-fpa-mic.

If you use the wrong microcode for your configuration, your machine will not boot, except in the case where your system has an FPA and you use a non-FPA microcode. In this case, the machine functions normally, but does not make use of the FPA at all.

boot-file-to-update specifies whether to update the boot file with the new microcode version number. It accepts one of these values (the default value is **nil**):

<i>Value</i>	<i>Action</i>
nil	Prompts for a boot file to update.
pathname	Does not prompt but uses pathname as the boot file to update.
:no-boot-file-update	Does not prompt or update.

The following function copies a world from another machine:

si:receive-band *source-host source-band &optional target-band subset-start subset-n-blocks* *Function*

Copies a file named *source-band* from the FEP file system of a remote *source-host* to a file named *target-band* on the FEP file system of the local machine.

source-band is a pathname that defaults to a file on FEP: (the boot unit) with a file type of .load and a version of .newest. The default pathname for *target-band* is *source-band*. If *target-band* already exists and is the file from which the current world was fast-booted, **si:receive-band** creates a new version. If *target-band* exists but is not the file from which the world was fast-booted, **si:receive-band** offers a choice of overwriting, superseding, or writing a new version.

subset-start and *subset-n-blocks* can have the following combinations of values:

<i>subset-start</i>	<i>subset-n-blocks</i>	<i>Action</i>
nil	nil	Copy entire file
integer	nil	Copy from block <i>subset-start</i> to end
integer	integer	Copy <i>subset-n-blocks</i> starting at block <i>subset-start</i>
list of integers	nil	Copy only blocks listed in <i>subset-start</i>

The default is **nil** for both *subset-start* and *subset-n-blocks*.

If both machines are on the same Ethernet cable and are running at least Release 5.0, **si:receive-band** uses a fast transfer protocol that copies about five megabytes per minute (about four minutes for a typical world load file). This is likely to be about three times as fast as slow transfer.

If both machines are running at least Release 5.0, **si:receive-band** checksums the transfer or the entire file after copying the file. If bad blocks are found it offers to retransfer the bad blocks. If you elect not to retransfer, it returns a list of the bad blocks. You can use this list as the *subset-start* argument to **si:receive-band**, **si:transmit-band**, **si:checksum-band**, or **si:compare-band**. See the function **si:checksum-band**, page 52.

If either machine does not support checksumming, you can compare the two files using **si:compare-band**.

si:receive-band prints an integer for each 100 blocks it has transferred. It puts up a display on the remote machine in the status line, indicating that the remote machine is serving the local machine.

si:transmit-band transfers a FEP file from a local machine to a remote machine.

To copy a world to another machine, use **si:transmit-band**.

si:transmit-band *source-band target-host &optional target-band* *subset-start subset-n-blocks* *Function*

Copies a file named *source-band* from the FEP file system of the local machine to a file named *target-band* on the FEP file system of a remote *target-host*.

source-band is a pathname that defaults to a file on FEP: (the boot unit) with a file type of .load and a version of .newest. The default pathname for *target-band* is *source-band*. If *target-band* already exists and is the band from which the world on *target-host* was fast-booted, **si:transmit-band** refuses to perform the transfer. If *target-band* already exists but is not the band from which the remote world was fast-booted, **si:transmit-band** overwrites it.

subset-start and *subset-n-blocks* can have the following combinations of values:

<i>subset-start</i>	<i>subset-n-blocks</i>	Action
nil	nil	Copy entire file
integer	nil	Copy from block <i>subset-start</i> to end
integer	integer	Copy <i>subset-n-blocks</i> starting at block <i>subset-start</i>
list of integers	nil	Copy only blocks listed in <i>subset-start</i>

The default is **nil** for both *subset-start* and *subset-n-blocks*.

If both machines are on the same Ethernet cable and are running at least Release 5.0, **si:transmit-band** uses a fast transfer protocol that copies about five megabytes per minute (about four minutes for a typical world load file). This is likely to be about three times as fast as slow transfer.

If both machines are running at least Release 5.0, **si:transmit-band** checksums the transfer or the entire file after copying the file. If bad blocks are found it offers to retransfer the bad blocks. If you elect not to retransfer, it returns a list of the bad blocks. You can use this list as the *subset-start* argument to **si:transmit-band**, **si:receive-band**, **si:checksum-band**, or **si:compare-band**. See the function **si:checksum-band**, page 52.

If either machine does not support checksumming, you can compare the two files using **si:compare-band**.

si:transmit-band prints an integer for each 100 blocks it has transferred. It puts up a display on the remote machine in the status line, indicating that the remote machine is serving the local machine.

si:receive-band transfers a FEP file from a remote machine to a local machine.

It is good practice to compare the copied world to the original. Use **si:checksum-band** or **si:compare-band**.

si:checksum-band *source-host source-band &optional target-band subset-start subset-n-blocks* *Function*

Checksums a file named *source-band* from the FEP file system of a remote *source-host* with a file named *target-band* on the FEP file system of the local machine. Prints a report of any differences in the files. Both machines must be running at least Release 5.0.

The checksum is per block, transmitted over a reliable byte stream connection. **si:checksum-band** is considerably faster than **si:compare-band**; a checksum for a typical world load file takes about three minutes. However, it is not guaranteed to be 100 percent accurate because it uses a hashing algorithm.

source-band is a pathname that defaults to a file on FEP: (the boot unit) with a file type of .load and a version of .newest. The default pathname for *target-band* is *source-band*.

subset-start and *subset-n-blocks* can have the following combinations of values:

<i>subset-start</i>	<i>subset-n-blocks</i>	Action
nil	nil	Checksum entire file
integer	nil	Checksum from block <i>subset-start</i> to end
integer	integer	Checksum <i>subset-n-blocks</i> starting at block <i>subset-start</i>
list of integers	nil	Checksum only blocks listed in <i>subset-start</i>

The default is **nil** for both *subset-start* and *subset-n-blocks*.

si:checksum-band prints an integer for each 100 blocks it has checksummed. It puts up a display on the remote machine in the status line, indicating that the remote machine is serving the local machine.

si:checksum-band returns a list of the bad blocks. You can use this list as the *subset-start* argument to **si:transmit-band**, **si:receive-band**, **si:checksum-band**, or **si:compare-band**.

si:receive-band transfers a FEP file from a remote machine to a local machine. **si:transmit-band** transfers a FEP file from a local machine to a remote machine.

si:compare-band *source-host source-band* &optional *target-band* *subset-start subset-n-blocks* *Function*

Compares a file named *source-band* from the FEP file system of a remote *source-host* with a file named *target-band* on the FEP file system of the local machine. Prints a report of any differences in the files.

source-band is a pathname that defaults to a file on FEP: (the boot unit) with a file type of .load and a version of .newest. The default pathname for *target-band* is *source-band*.

subset-start and *subset-n-blocks* can have the following combinations of values:

<i>subset-start</i>	<i>subset-n-blocks</i>	<i>Action</i>
nil	nil	Compare entire file
integer	nil	Compare from block <i>subset-start</i> to end
integer	integer	Compare <i>subset-n-blocks</i> starting at block <i>subset-start</i>
list of integers	nil	Compare only blocks listed in <i>subset-start</i>

The default is **nil** for both *subset-start* and *subset-n-blocks*.

si:compare-band prints an integer for each 100 blocks it has compared. It puts up a display on the remote machine in the status line, indicating that the remote machine is serving the local machine.

si:compare-band returns a list of the bad blocks. You can use this list as the *subset-start* argument to **si:transmit-band**, **si:receive-band**, **si:checksum-band**, or **si:compare-band**.

si:checksum-band performs a checksum comparison of two FEP files.
si:receive-band transfers a FEP file from a remote machine to a local machine. **si:transmit-band** transfers a FEP file from a local machine to a remote machine.

7.5 Installing New Microcode

When an existing system is to be used with a new microcode, certain changes need to be made to the system, and it should then be saved out with the changes. Usually new microcode is released only along with a new system, so you hardly ever have to do this. The Debugger has a table of errors that are detected by microcode. The symbols in this table are used when debugging other machines and by certain metering programs. The Debugger updates these tables when you install a new microcode.

The Debugger automatically updates its table from a file in the `sys:l-ucode;` directory when you boot the machine with the new microcode.

8. Instructions for Managing Disk Space on the 3640

Since the 140 megabyte disk drive of the 3640 contains a smaller paging file than the 3600 or 3670, you must manage your 3640 FEP file system differently. For a complete description of paging files: See the section "FEP File Types" in *Reference Guide to Streams, Files, and I/O*.

This section describes the different procedures that you follow to manipulate paging space when:

- Loading the world.
- Customizing and saving that world load.
- Saving future world loads.

The disk of your 3640 contains a world load file, a large paging file (called *Page.page*), and an auxiliary file that is the same size as the world load file. You will use the auxiliary file in one way for normal operation and in another way when putting a new world on the disk.

WARNING: If your system does not contain an auxiliary file (use the **print-herald** function to look for the file named *Aux.page*), call your field representative.

In normal operation, you boot a world load file and use both *Page.page* and the auxiliary file for paging. In this case, you call the auxiliary file *Aux.page*.

When you want to create a new world or transfer a new world to the disk, you boot your world load file and use only *Page.page* for paging. Instead of using the auxiliary file for paging, you rename it and use it as a world repository. Once you have successfully created the new world, you rename the old world load file to *Aux.page* and use it as your auxiliary paging file.

The auxiliary file is always actively in use, either as:

- A paging file (in normal operation)
- The target file for new world load

8.1 Customizing and Saving the World

The shipped configuration assumes the auxiliary file (*Aux.page*) is to receive your site's customized world load and so contains just one actual paging file:

```
distribution-world.LOAD.1  
30,000
```

```
Aux.page.1      30,000
Page.page.1     45,000
```

A customized, normal configuration uses the auxiliary file as a paging file and so contains two paging files:

```
New-world.load  30,000
Aux.page.1      30,000
Page.page.1     45,000
```

To create your customized world, follow these instructions:

1. Boot the distribution world using the correct microcode. Use only Page.page.1 for paging and reserve the auxiliary file. You should to initially boot by hand rather than use the boot file so that you can set your chaos address:

```
Clear Machine
Load Microcode microcode-file-name
Load World world-load-file-name
Clear Paging
Add Paging >Page.page
Set Chaos-Address this-machine's-chaos-address
Start
```

2. Login by using **si:login-to-sys-host**, for example:

```
(si:login-to-sys-host)
```

3. Rename the auxiliary file to whatever name you wish, for example:

```
Rename File FEP:>Aux.page.1 FEP:>New-world.load.1
```

4. Customize the booted world and then save it into your new world load file:

```
Save World FEP:>New-world.load.1
```

Since you are asking to save the world into an existing file, you are prompted for an action with which to proceed. The correct answer is Overwrite. Then you are asked if you want to update the boot file. Answer yes. The Set Chaos line that you manually typed is added to the boot file at this time.

5. Rename the distribution world to be the auxiliary file:

```
Rename File FEP:>distribution-world.LOAD.1 FEP:>Aux.page.1
```

6. At this point, you should edit the boot file, FEP:>Boot.boot, to add the auxiliary file as an additional paging file. After the line Add Paging >Page.page, add:

```
Add Paging >Aux.page
```

Your edited boot file should look like this:

```

Clear Machine
Load Microcode microcode-file-name
Load World world-load-file-name
Clear Paging
Add Paging >Page.page
Add Paging >Aux.page
Set Chaos-Address this-machine's-chaos-address
Start

```

Save the edited version.

7. Log out and halt the machine.
8. Boot the new world using the boot file.

8.2 Saving Subsequent Worlds

Whenever you wish to create a new world on your 3640 disk, you must follow a similar procedure to that shown above.

1. Boot manually, and do *not* type the Add Paging >Aux.page command, since you will be saving the latest world into it:

```

Clear Machine
Load Microcode microcode-file-name
Load World world-load-file-name
Clear Paging
Add Paging >Page.page
Set Chaos-Address this-machine's-chaos-address
Start

```

2. Login by using **si:login-to-sys-host**, for example:

```
(si:login-to-sys-host)
```

3. Rename the auxiliary file to whatever name you wish, for example:

```
Rename File FEP:>Aux.page.1 FEP:>Newer-world.load.1
```

4. Either customize the booted world and save it into your new world load file, or transfer the world from some other machine:

```
Save World FEP:>Newer-world.load.1
```

or:


```
(si:receive-band "other-machine-name" "remote-world-name"  
"FEP:>Newer-world.load.1")
```

Since you are asking to save the world into an existing file, you are prompted for an action with which to proceed. The correct answer is Overwrite. Then you are asked if you want to update the boot file. Answer yes.

5. Rename the old world to the auxiliary file to be used for paging:

```
Rename File FEP:>Old-world.LOAD.1 FEP:>Aux.page.1
```

6. Log out and halt the machine.
7. Boot the machine using the new boot file.

9. Booting a World

To boot a world load, type the following to the FEP:

```
Fep>clear machine
Fep>load microcode microcode-file-name
Fep>load world world-load-file-name
Fep>set chaos-address this-machine's-chaos-address
Fep>Start
```

If you encounter any errors with a particular command, you should try that command again. To avoid typing all those commands every time you boot a world, you can use a file named >Boot.boot. This file contains text similar to the above example. The Boot command activates the file; for example, the command Boot >boot.boot tells the FEP to execute the commands in the file >Boot.boot.

10. Installing and Configuring the Mailer

When you want to offer store-and-forward mail service on a Symbolics host (or on several hosts), you must update the host object corresponding to the hosts that will offer the service.

This installation is actually a further customization of the new release world for your site; that is, the particular world that is run on the host providing mail service is different than the worlds running on the other machines at the site. The difference is that the host providing mail service runs with a larger world that contains the **mailer** system.

For an overview on changing objects in the namespace database: See the section "Registering Users and Hardware in the Site's Namespace", page 71.

1. Type: Edit Namespace Object Host, and answer the prompt with the name of the host that is to provide mail service. If this host is going to send and receive mail over telephone lines, and if you have loaded a Dialnet registry that included this host, the namespace editor asks whether you want to edit the DIAL or the local namespace view of this host. Choose the local view. For more information on Dialnet: See the section "Symbolics Dialnet" in *Networks*.
2. Using the menu, add the following service triplets to the host object definition:
MAIL-TO-USER CHAOS CHAOS-MAIL
STORE-AND-FORWARD-MAIL CHAOS CHAOS-MAIL
3. If the host has an address on the DIAL network, also add the following triplets; otherwise, skip this step:
MAIL-TO-USER DIAL SMTP
STORE-AND-FORWARD-MAIL DIAL SMTP
4. After you have added the services in the previous steps, click on [Save] to save the changes.
5. Wait for confirmation that the changes have been saved, and then click on [Quit] to exit from the namespace editor.

The mail services have now been registered into your namespace database, for the host you specified.

The next procedure installs the Mailer software onto the machine that is going to be the mail server, which is the same host as specified above.

1. You should free up enough disk space in the FEP file system of the server machine to accommodate the following material:

- There should be enough space for a world about 10% larger than the world already there. (The world with the Mailer loaded is saved back onto the disk at the end of this procedure.) If mail service is to be provided by a Symbolics 3640: See the section "Instructions for Managing Disk Space on the 3640", page 55.
- The Mailer sources and binaries occupy about 100 LMFS blocks.
- The Mailer databases each require a minimum of 20 LMFS blocks. The size of a database increases approximately linearly with the number of messages stored by the mail server (the host); an "average" piece of mail occupies about 2 LMFS blocks.

You can use the Dired (M-X) Zmacs command or SELECT F (the File System Editor) to delete unused or excess files in the same way that you freed up enough space for the customized new release world.

2. Using the Create Directory command, create the following directories on the host:

```
>Mail>
>Mail>Static>
>Mail>Dynamic>
>Mail>Hardcopy>
>Mail-Archive>
```

3. Edit the file >MAIL>STATIC>MAILBOXES.TEXT. This file defines mailing lists served by this mail server, as well as local delivery paths for this server. An example follows:

```
; -*- Mode: Lisp; Package: Mailer; Lowercase: Yes -*-

;; This file belongs on White only.

;; The mail addresses "Postmaster" and "File-Server" are resolved with respect
;; to individual hosts. Edgar gets all mail addressed to these "non-people" users,
;; will be the person who takes care of dead letters, etc.

(define-local Postmaster Edgar)
(define-local File-Server Edgar)
(define-local Mail-Server Edgar)
(define-local Lisp-Machine Edgar)

;;; People who get mail delivered on White
(deliver-local Robert Edgar Henry Malcolm Peterson Davies Richard Smiley Nelson)
```

```

;;; THE basic mailing list for bugs. Each person listed gets a copy of each bug
;;; report, and a copy goes to the files.
(define Bug-LispM
  (list Edgar Davies Henry
        "white:>Mail-Archive>Bug-Reports.text"))

;;; Miscellaneous mailing lists

;;; Good to have a mailing list that allows you to reach everyone
(define seventh-crisis
  (list Robert Edgar Henry Malcolm Peterson Davies Richard Smiley Nelson))

;;; Simple mailing list, commented to show particular preferences
(define sports-followers
  (list Richard      ;; golf
        Smiley      ;; gymnastics
        Nelson)     ;; tennis

;;; Another simple list.
;;; Put yourself on this list if you want to hear about tape bargains
(define tapers
  (list Richard Henry))

```

Write out the file, and stay in the editor for the next step.

4. Edit the file >MAIL>STATIC>OPTIONS.LISP. Set it up to look like this:

```

;;; -*- Mode: LISP; Package: MAILER; Base: 10; Syntax: ZetaLisp -*-

(setq mailer:deferred-delivery-times t) ;; send dial mail immediately
(setq mailer:failed-mail-reply-file t) ;; archive failed mail

```

Write out the file. For a description of the options: See the section "Files and Directories Used by the Mailer" in *Communicating with Other Users*.

5. To install the Mailer system, you should first cold boot the host. After cold booting the host, you disable services, log in to the sys host, and then use the Load System command to make the Mailer system. After using Load System, you should then save the world with the Save World command or the **disk-save** function.

Installing the mailer system is shown in the following example:

```

Command: Halt Machine RETURN
Fep>boot RETURN
.
.
.
Command: Disable Services RETURN

```

Command: (si:login-to-sys-host)

Command: Load System Mailer

Files to be loaded:

.
.
.

Command: Save World *name-of-file-to-save-it-in*

If the host machine is doing nothing but loading the Mailer, the Load System step takes about two minutes. You have now configured the newly saved world to be a mail server for the site.

6. Boot the new world using the FEP Boot command. Initialize and start the mailer by invoking **mailer:launch**. Shortly thereafter, SELECT 0 will bring up the Mailer Log window. Test the mailer by sending messages from various machines at the site.

10.1 Configuring Large Sites for Multiple Mail Servers

The store-and-forward mailer supports *forwarding tables* to help coordinate mail delivery at sites with several mail servers.

One particular mail server is configured to be in charge of forwarding-table maintenance. The forwarding tables themselves are written by the this host to the file systems of all the other mail servers at the site. This asymmetry is, in a sense, a further customization of the particular mail server that writes the forwarding tables. The customization is usually done by placing a **setq** of the variable **mailer:forwarding-table-hosts** in the init file that is run when the server is booted. For example:

```
(SETQ MAILER:FORWARDING-TABLE-HOSTS
  ("MANFRED" "NATASHA" "BORIS"))
```

Here, the hosts Manfred, Natasha, and Boris receive new forwarding tables from the host to which this init file belongs. The forwarding table for a given host is written in the file >MAIL>DYNAMIC>FORWARDING.TEXT on that host's local LFMS.

If you want to run the identical init file on all the server machines at a site, the following example may be instructive. Here, a sys host (Fearless) runs the Mailer and is responsible for writing out the forwarding tables. The File-Server init file, which all file servers use, includes the following lines:

```
(DEFMACRO FILE-SERVER-ONLY-ON (HOSTS &BODY BODY)
  '(WHEN (OR ,@(LOOP FOR HOST IN HOSTS
    COLLECT '(SEND NET:*LOCAL-HOST* :PATHNAME-HOST-NAMEP ,(STRING
      HOST)))) ,@BODY))

(FILE-SERVER-ONLY-ON (FEARLESS)
  (SETQ MAILER:FORWARDING-TABLE-HOSTS
    '("MANFRED" "NATASHA" "BORIS")))
```

The file MAILBOXES.TEXT on Fearless contains the names of all the mailing lists for this network. In addition to the usual forms defining mailing lists, the file contains forms like the following:

```
;; What follows is a global table of mail addresses for the network.
;; There is one entry for each host, listing all of the mail addresses to be
;; forwarded to that host. Each entry is broken into two sublists,
;; one for mailing lists and one for individuals. This is the only
;; place in which this table should be edited.
;; The forwarding tables for all other hosts are generated from this one.
```

```
(DELIVER-TO NATASHA
```

```
;;; The mailing list file on Natasha is >mail>static>mailboxes.text.
```

```
;; Individuals
Andy Bob Charles David Edgar

;; Lists
ASAS Audio Audiophiles
BBoard Bikers Bleeding-Hearts Bridge
...)
```

Similar **deliver-to** forms are supplied for Boris and all other hosts with store-and-forward-mailers.

When the Mailer on Fearless is booted (that is, when **si:enable-services** or **mailer:launch** is called), or when the Mailer notices that the local MAILBOXES.TEXT file has changed and has been stable for at least 10 minutes, it reads in its MAILBOXES.TEXT file (Fearless will never have a FORWARDING.TEXT file) and then writes out FORWARDING.TEXT files on all the other Mailer hosts. Those hosts eventually read in the new FORWARDING.TEXT files and their own MAILBOXES.TEXT files, merge the two sets of definitions, and carry on.

The FORWARDING.TEXT file that Fearless generates for Boris includes forms for hosts with store-and-forward mailers, such as NATASHA:

```
; Mailbox forwarding table for BORIS.
; Written 2/24/85 15:33:54 by Mail-Server running on FEARLESS.
; From F:>Mail>Static>Mailboxes.text created on 2/24/85 15:25:24.
; This table is automatically generated by a program. Do not edit it.
```


(DELIVER-TO NATASHA

Andy Bob Charles David Edgar
ASAS Audio Audiophiles
BBoard Bikers Bleeding-Hearts Bridge

If Boris gets incoming mail for these individuals or lists, the mail is forwarded to Natasha. There is no entry for Boris in this list, since those entries come from the MAILBOXES.TEXT file on Boris.

11. Installing Printers

This section describes the procedures for installing ASCII print servers. The section **Registering a Printer** describes the procedures to register the printer after the hardware has been installed.

11.1 Installing an ASCII Print Server on UNIX

The ASCII print server allows someone on a Lisp Machine to hardcopy a file on an ASCII printer, such as a line printer or a DECwriter, on some UNIX system.

To bring up an ASCII printer named Stumpy on host Cupid:

1. Create the printer object named Stumpy (using **tv:edit-namespace-object**) and give it the following attributes:

```
Type ASCII
Format ASCII
Default-font FAKE9
Header-font FAKE9
host CUPID
```

See the function **tv:edit-namespace-object**.

2. Edit the host object Cupid and add the attributes:

```
Service HARDCOPY CHAOS LGP
Printer STUMPY
```

On the UNIX (BSD) 4.2 network tape, there is a server which answers the "LGP" contact name on the Chaosnet. This server simply gets some information for the banner page and then calls upon the "lpr" spooling system to do the work with the following line:

```
exec1("/usr/ucb/lpr", "LGP", "-l", "-T", nargs[FILE_NAME], 0);
```

Refer to *4.2BSD Line Printer Spooler Manual* (in the /usr/doc documents directory) and manual pages for lpr and printcap for background information on how this all works.

As explained in the above documents, there are several ways for a system manager to configure the lpr system. The exec1 command above uses the -l argument to allow binary interpretation by the spooler. This is appropriate for an LGP printer. The -T gives the spooler the name to use on the banner page. Because there is no -P specification, lpr will use its default device. In the distribution system, this uses /dev/lp as the device and spools in the directory /usr/spool/lp. It is easy to modify the

LPR configuration to use a different default device and/or a different spooling directory.

To get the LGP server to use a different printer (here, "myprinter"), first install a printcap entry for the printer. This includes specifying a spooling directory for the printer, so make that directory if it doesn't already exist. Then change the `execl` statement to the following:

```
execl("/usr/ucb/lpr", "LGP", "-l", "-P", "myprinter", "-T",
      nargs[FILE_NAME], 0);
```

Be sure to link `/dev/myprinter` to the appropriate hardware device. For example, by linking `/dev/console` to `/dev/myprinter`, making the directory `/usr/spool/myprinter`, and installing the following entry in `/etc/printcap` one can use the console DECwriter as a remote printer with the above change to `execl`:

```
myprinter|local line printer:\
      :lp=/dev/myprinter:sd=/usr/spool/myprinter:lf=/usr/adm/lpd-errs:
```

If you plan to use an ASCII printer regularly from a Lisp Machine, you should probably include the following in your `lisp-init.l` file:

```
(login-forms
 (setq press:*user-hardcopy-page-headings* nil))
```

11.2 Installing an ASCII Print Server on VMS

The ASCII print server allows someone on a Lisp Machine to hardcopy a file on an ASCII printer, such as a line printer or a DECwriter, from some VMS system.

To bring up an ASCII printer named Stumpy on host Comet:

1. Create the printer object named Stumpy (using `tv:edit-namespace-object`) and give it the following attributes:

```
Type ASCII
Format ASCII
Default-font FAKE9
Header-font FAKE9
host COMET
```

See the function `tv:edit-namespace-object`.

2. Edit the host object Comet and add the attributes:

```
Service HARDCOPY CHAOS LGP
Printer STUMPY
```

3. Edit the file `SYS$MANAGER:CHSTART.COM` to include the print server in the `INSTALL` commands. Insert the line:

```
CHAOS$RFC:LGP/OPEN/SHARED/HEADER/PRIV=(TMPMBX,SYSPRV)
```

after the command that runs the INSTALL utility.

4. You should also add the following commands to the SYSS\$MANAGER:SYSTARTUP.COM file:

```
$ ASSIGN/SYSTEM <spooled device> <printer name>  
$ SET/DEVICE=<spooled device> <spooled device>  
$ INIT/QUEUE <spooled device>  
$ START/QUEUE <spooled device>
```

where <spooled device> is the name of the device that will be a spooled printer and <printer name> is the name of the network printer object.

If you plan to use an ASCII printer regularly from a Lisp Machine, you should probably include the following in your lisp-init.l file:

```
(login-forms  
 (setq press:*user-hardcopy-page-headings* nil))
```


12. Registering Users and Hardware in the Site's Namespace

After you bring up the new release, you can perform the following steps as part of your site administration activities:

- Register users, hosts, printers, and networks in the site's namespace
- Install printers
- Move the new release to other Lisp Machines at the site
- Install new releases distributed in patch tape format
- Install new releases distributed in world load format
- Install world loads from other sites

You should reflect any changes, such as new users or changes in the site's hardware, in the namespace database. Register new hosts and printers in the namespace database before connecting them to the network or supporting host. Register new users in the namespace database either before they use the new release or as part of the process when they log in for the first time. Whether you are registering new users or new hardware, the process is most easily done by copying and modifying an existing object of the same type using the namespace editor (invoked by the Edit Namespace Object command or the **tv:edit-namespace-object** function), and then saving the new object. Once it has been saved it is part of your site's configuration and all machines running the new release know about the new object the next time they boot, or sooner in some cases. (See the section "Namespace System" in *Networks*.)

To use the namespace editor to create and update objects, click on *Namespace* on the System Menu or select a Lisp Listener and type the Edit Namespace Object command or the **tv:edit-namespace-object** function.

12.1 Registering Users

The easiest registration strategy is to create your entry by copying someone else's entry. To copy another entry, use the Edit Namespace Object command and then the namespace editor's [Copy] command. If you are the first person at your site to register, copy the user object for user LISP-MACHINE.

Individual users can run the the Edit Namespace Object command or the **tv:edit-namespace-object** function for themselves the first time that they use a new release. If they have not created an appropriate user object, then logging into the new release fails because the system does not find the user object in the namespace database. Should this be the case, the system offers to create the user object with **tv:edit-namespace-object**. Use the namespace editor's [Copy] command and copy the user object for user LISP-MACHINE.

For an overview on changing objects in the namespace database: See the section "Registering Users and Hardware in the Site's Namespace", page 71.

12.2 Registering Hosts

To create the new host object in the site namespace, type:

```
Edit Namespace Object Any RETURN
```

Click on [Create] and then specify the name of the new host. Or, use the form:

```
(tv:edit-namespace-object :host New-Host :create t)
```

To determine the service attributes required for your host type (LISPM, UNIX, or VMS): See the section "Namespace Attributes for New Hosts", page 79.

12.3 Registering a Tape Drive in the Namespace

To register a tape drive for a Symbolics machine, use the Edit Namespace Object command to add the tape drive to the namespace database. Specify tape chaos rtape as the last Service: Set: entry in the host object. See the section "**service: Host Object Attribute**" in *Networks*. For example:

```
Edit Namespace Object :host Janis RETURN
```

pops up namespace menu that displays all the attributes of the host Janis. Add the tape service by clicking on Set: of the last Service: Set: entry, then type:

```
tape chaos rtape RETURN
```

You can also use the **tv:edit-namespace-object** function to invoke the namespace menu for editing the printer and host objects; for example:

```
(tv:edit-namespace-object :host "Janis")
```

12.4 Registering a Printer

In addition to the hardware installation of your printer, you must register the printer in the namespace database. This involves creating a printer object and updating the host object corresponding to the host to which the printer is connected.

This installation is actually a further customization of the new release world for your site; that is, the particular world that is run on the host controlling the printer is different than the worlds running on the other machines at the site. The difference is that the host controlling the printer runs with a larger world that contains the **print** system.

For an overview on changing objects in the namespace database: See the section "Registering Users and Hardware in the Site's Namespace", page 71.

1. Type: Edit Namespace Object Any and when the menu appears click on [Create]. Then, click on [Printer]. The editor asks for the name of the printer object you want to create. Enter the name of the printer. This should be the actual name of the printer, not the pretty name. For example, a printer might have a pretty name of Echo Lake, but the actual name is Echo-Lake.
2. Click on [Global-name] next to the "Type*:" field. Enter the type of printer, such as LGP or ASCII. Fill in the "Site" and "Pretty Name" fields with the name of the site and the pretty name for the printer, respectively.

Set the interface field to the type of interface that connects the printer to the host. If you are hooking up the LGP via a serial interface, specify serial. If you are hooking up the LGP via a DR11-C, specify DR11-C. In the "Host" field, specify the host to which you are connecting the LGP.

For the "Fonts Widths File" entry, specify the physical pathname of the font widths information file for the printer, if one exists. To get the physical pathname of the directory containing the most current fonts width information for LGPs, use the form:

```
(send (fs:parse-pathname "sys: fonts; lgp-1; fonts.width.newest")
:translated-pathname)
```

which returns, for example:

```
#<LMFS-PATHNAME "T:>sys>fonts>lgp-1>fonts.width">
```

To construct the entry, prepend the name of your site followed by a vertical bar, for example, MYSITE|T:>sys>fonts>lgp-1>fonts.width.

The details of attaching a printer to a host are recorded in the "Interface Options" attribute of the printer object, as are any important characteristics of the particular printer.

If the printer is attached to the 3600-family processor by a serial line, the printer object should reflect this by an Interface Option attribute listing the baud rate and unit number (the number of the serial interface on the processor's IO bulkhead). Find the "Interface Options" field. Click on the leftmost [Global-name] and enter baud; next, click on *Token* to the immediate right and enter the appropriate baud rate as a decimal number, for example, 9600 or 1200. Click on Interface Options' [Global-name] field again and enter UNIT, then click on the adjacent *Token* and enter the unit number.

If the printer is a Symbolics LGP, and contains Rev-2A proms, find the "Interface Options" attribute for the printer (if not there already from, for example, setting a serial printer's baud rate and unit number). Click on the leftmost [Global-name] field and enter PROM-VERSION, then click on the adjacent *Token* and enter REV-2A. Do *not* specify this option if you are unsure of the proms in your printer, as the wrong thing in this field truncates screen hardcopy on the right.

The fields described above are sufficient to describe an LGP printer. Other printers might require that additional fields be supplied in order to operate correctly. Please consult with your Symbolics Field Office Software Support representative for details.

3. Click on [Save] to save the printer object.
4. Click on [Edit], then click on [Host]. When the namespace editor asks which host you want to edit, specify the host to which you intend to connect the printer.
5. Find the "Spooled Printer" field. Click on [Printer] and enter the name of the printer.
6. To specify the spooling directory for the printer, in the "Spooled Printer" field for the printer, click on [Global-name] and type home-directory, then click on the *Token* next to it and type the spooling directory name.
7. Find the last "Service" field. Click on [Global-name]. Type the service name, Hardcopy. Click on the [Global-name] that just popped up to the right, in the same "Service" field. Type the network by which requests will be queued; usually this is Chaos. Click on the [Global-name] that just popped up to the right. Enter the contact name; for LGPs this is LGP.
8. A new service field has just been created below the one you edited. Click on [Global-name]. Type the service name, Hardcopy-status. Click on the [Global-name] that just popped up to the right, in the same "Service" field. Type the network by which requests will be queue. This network name will be the same name as entered in the previous field. Click on the [Global-name] that

just popped up to the right. Enter the contact name. For LGPs this would be LGP-STATUS.

9. The printer has now been registered into your namespace database. The next step installs the printer server software onto the machine that is going to be the printer server, which is the same host as specified above.

You should free up enough disk space in the FEP file system of the server machine to hold a world that is approximately 10 percent larger than the new release world customized for your site. You can use the Dired (m-x) Zmacs command or SELECT F to select the File System Editor to delete unused or excess files in the same way that you freed up enough space for the customized new release world.

To install the print system, you should first cold boot the host. After cold booting the host, you disable services, log in to the sys host, and then use **make-system** to make the print system. After using **make-system**, you should then save the world with the Save World command or the **disk-save** function.

Installing the print system is shown in the following example:

```
Command: Halt Machine RETURN
Fep>boot RETURN
.
.
.
Command: Disable Services RETURN

Command: (si:login-to-sys-host)

Command: (make-system 'print)

Files to be loaded:

SYS: PRINT; REQUEST.BIN.8
SYS: PRINT; SPOOLER.BIN.7
SYS: PRINT; DEVICE.BIN.7
SYS: PRINT; LGP.BIN.7

Making Print patchable
Print version 38. loaded

Command: Save World name-of-file-to-save-it-in
```

10. You have now configured the newly saved world to be a print server for the site. Boot the new world using the FEP Boot command and test the printer by hardcopying files or screen images from various machines at the site.

12.5 Registering an ASCII Print Server on VMS

The ASCII print server allows someone on a Lisp Machine to hardcopy a file on an ASCII printer, such as a line printer or a DECwriter, from some VMS system.

To bring up an ASCII printer named Stump-Pond on host Comet:

1. Create the printer object named Stump-Pond (using the Edit Namespace Object command or the **tv:edit-namespace-object**) function and give it the following attributes:

```
Type ASCII
Format ASCII
Default-font FAKE9
Header-font FAKE9
host COMET
```

For an overview on changing objects in the namespace database: See the section "Registering Users and Hardware in the Site's Namespace", page 71.

2. Edit the host object Comet and add the attributes:

```
Service HARDCOPY CHAOS LGP
Spooled-Printer STUMP-POND
```

3. Read the tape containing the VMS ASCII print server software into your VMS system using BACKUP. The tape was written via this command:

```
$ BACKUP CHAOS$DISK:[CHAOS...]MT:ASCLGP.BCK/DENSITY=1600
```

The following commands suffice to retrieve the software:

```
$ MOUNT/FOREIGN MTAn
$ BACKUP MT:ASCLGP.BCK disk:[*...]*.*.*
```

4. Edit the file SYSS\$MANAGER:CHSTART.COM to include the print server in the INSTALL commands. Insert the following line after the command that runs the INSTALL utility:

```
CHAOS$RFC:LGP/OPEN/SHARED/HEADER/PRIV=(TMPMBX,SYSPRV)
```

5. You should also add the following commands to the SYSS\$MANAGER:SYSTARTUP.COM file:

```
$ ASSIGN/SYSTEM <spooled device> <printer name>
$ SET/DEVICE=<spooled device> <spooled device>
$ INIT/QUEUE <spooled device>
$ START/QUEUE <spooled device>
```

where <spooled device> is the name of the device that will be a spooled printer and <printer name> is the name of the network printer object.

If you plan to use an ASCII printer regularly from a Lisp Machine, you should probably include the following in your init file:

```
(login-forms  
  (setq press:*user-hardcopy-page-headings* nil))
```


13. Namespace Attributes for New Hosts

To create the new host object in the site namespace, type:

```
Edit Namespace Object Any RETURN
```

Click on [Create] and then specify the name of the new host. Or, use the form:

```
(tv:edit-namespace-object :host New-Host :create t)
```

Now add the attributes as shown in the sample template for your host type:

13.1 Namespace Attributes for a New Lisp Host

```
System Type*: LISPM
Service: Set: CHAOS-STATUS CHAOS-SIMPLE CHAOS-STATUS
Global-name
Service: Set: SHOW-USERS CHAOS NAME Global-name
Service: Set: TIME CHAOS-SIMPLE TIME-SIMPLE Global-name
Service: Set: UPTIME CHAOS-SIMPLE UPTIME-SIMPLE
Global-name
Service: Set: LOGIN CHAOS TELNET Global-name
Service: Set: LOGIN CHAOS SUPDUP Global-name
Service: Set: LOGIN CHAOS 3600-LOGIN Global-name
Service: Set: SEND CHAOS CONVERSE Global-name
Service: Set: SEND CHAOS SEND Global-name
Service: Set: NAMESPACE CHAOS NAMESPACE Global-name
Service: Set: NAMESPACE-TIMESTAMP CHAOS-SIMPLE
NAMESPACE-TIMESTAMP Global-name
Service: Set: LISPM-FINGER CHAOS-SIMPLE LISPM-FINGER
Global-name
Service: Set: FILE CHAOS QFILE Global-name
Service: Set: FILE CHAOS NFILE Global-name
Service: Set: Global-name
Address: Pair: CHAOS nnnnn Global-name
```

In the Address attribute line, *nnnnn* is a valid 5-digit octal Chaos address.

13.2 Namespace Attributes for a New UNIX Host

System Type*: UNIX
Service: Set: FILE CHAOS QFILE *Global-name*
Service: Set: TIME CHAOS-SIMPLE TIME-SIMPLE *Global-name*
Service: Set: MAIL-TO-USER CHAOS CHAOS-MAIL *Global-name*
Service: Set: STORE-AND-FORWARD-MAIL CHAOS CHAOS-MAIL
Global-name
Service: Set: SHOW-USERS CHAOS NAME *Global-name*
Service: Set: TAPE CHAOS RTAPE *Global-name*
Service: Set: SEND CHAOS SEND *Global-name*
Service: Set: LOGIN CHAOS SUPDUP *Global-name*
Service: Set: LOGIN CHAOS TELNET *Global-name*
Service: Set: NAMESPACE CHAOS NAMESPACE *Global-name*
Service: Set: NAMESPACE-TIMESTAMP CHAOS-SIMPLE
NAMESPACE-TIMESTAMP *Global-name*
Service: Set: UPTIME CHAOS-SIMPLE UPTIME-SIMPLE
Global-name
Service: Set: CHAOS-STATUS CHAOS-SIMPLE CHAOS-STATUS
Global-name
Service: Set: *Global-name*
Address: Pair: CHAOS *nnnnn Global-name*

In the Address attribute line, *nnnnn* is a valid 5-digit octal Chaos address.

13.3 Namespace Attributes for a New VMS Host

System Type*: VMS
Service: Set: FILE CHAOS QFILE *Global-name*
Service: Set: SHOW-USERS CHAOS NAME *Global-name*
Service: Set: TAPE CHAOS RTAPE *Global-name*
Service: Set: UPTIME CHAOS-SIMPLE UPTIME-SIMPLE
Global-name
Service: Set: TIME CHAOS-SIMPLE TIME-SIMPLE *Global-name*
Service: Set: CHAOS-STATUS CHAOS-SIMPLE CHAOS-STATUS
Global-name
Service: Set: LOGIN CHAOS TELNET *Global-name*
Service: Set: NAMESPACE CHAOS NAMESPACE *Global-name*
Service: Set: NAMESPACE-TIMESTAMP CHAOS-SIMPLE
NAMESPACE-TIMESTAMP *Global-name*
Service: Set: SEND CHAOS SEND *Global-name*
Service: Set: MAIL-TO-USER CHAOS CHAOS-MAIL *Global-name*
Service: Set: *Global-name*
Address: Pair: CHAOS *nnnnn Global-name*

In the Address attribute line, *nnnnn* is a valid 5-digit octal Chaos address.

14. Site Related Functions and Variables

In Release 6.0, the **send** function can be used to obtain the site name, as shown in the following example:

```
(send net:*local-site* :name)
:FISHERY
```

In Release 6.0, the **send** function can be used to obtain the specified option. The following example uses **send** to return the timezone option.

```
(send net:*local-site* :get :timezone)
:EST
```

In Release 6.0 the **defvar** and **add-initialization** functions should be used, rather than the **define-site-variable** function.

si:login-to-sys-host

Function

This function logs into the sys host as the default user. This is useful during installation procedures when you must do certain operations after cold booting that require the machine to be logged in.

chaos:my-address

Variable

The value representing the machine's Chaosnet address as used by the world load.

net:set-site *site-and-namespace-name* &key *ignore-existing-site*

Function

This function is used to change the site at which the current running band is operating. *site-and-namespace-name* is an argument whose value is a keyword symbol that names the site (such as **:ACME**). If *ignore-existing-site* is non-null, any cached information about the site is ignored.

si:site-initialization-list

Variable

Holds the value of the *site initialization* list, which is a special system initialization list. The **:site** keyword (as an element of the third argument to **add-initialization**) means that the initialization list to be added to is the site initialization list, and that the form should be evaluated now as well as added to the list (as with the **:now** keyword). Site initializations are run whenever the site parameters are changed; changes could be the namespace attributes of the local site or whether the site is local. Thus, by adding initializations to this list, you can specify things to do when the site parameters are changed, in case you write a program that is sensitive to the site at which it is run. Usually it is more convenient for you to use the **define-site-variable** special form to modify things as the site changes. But if it is not powerful enough, you can use this initialization list directly.

15. Herald Functions and Variables

The herald is the multiline greeting message that is displayed when you cold boot or warm boot, after you call the **print-herald** or **disk-save** functions or the FEP Disk Restore command. When you cold or warm boot, your machine displays a full-screen herald, which gives some basic information about your machine and how to operate it. When you display the herald with the **print-herald** function, you see an abbreviated herald.

print-herald &key *as-if-band verbose* *Function*

Prints out the herald message to **standard-output**. The herald message is what the machine prints when it is cold booted. It shows you the name of the FEP file for the current world load, any comment added to the herald, a measure of the physical memory and swapping space available, the versions of the systems that are running, the site name, and the machine's own host name.

:as-if-band is used by **disk-save** to supply the name of the FEP file of the saved world. **:verbose** controls the system version information displayed: if **t**, the version numbers of all systems are displayed; if **nil**, the version numbers of only those systems that differ from the release are displayed.

si:system-additional-info *Variable*

If you provide an additional comment to the herald using the Save World command or the **disk-save** function, **disk-save** sets this variable in the saved environment. The value of this variable should always be a string.

The following functions and variables are not actually used in printing the herald, but provide the same kind of information as does **print-herald**.

si:describe-system-versions &optional (*stream standard-output*) *Function*

Prints the major and minor version numbers of the release and of any systems whose version number is not exactly the released version number. The output is sent to *stream*, which defaults to **standard-output**.

si:get-system-version &optional (*system "System"*) *Function*

Returns three values. The first two are the major and minor version numbers of the version of *system* currently loaded into the machine. The third is the status of the system, as a keyword symbol: **:experimental**, **:released**, **:obsolete**, or **:broken**. *system* defaults to **System**. This returns **nil** if that system is not present at all.

si:print-system-status-warning &optional (*system "System"*) *Function*

If *system*'s status is **:experimental**, prints out a warning reminding the user

to load patches. If *system*'s status is **:broken**, prints out a warning cautioning the user that the system may not work. Otherwise, it does nothing.

PART II.

Software Installation Guide

16. Introduction to the Software Installation Guide

The Software Installation Guide presents installation instructions for different types of installations:

Installing a new release on a 3640

To install a new release on a 3640 with only one 140-Mbyte disk, you must allocate enough room on your disk for installing a new release. For instructions on how to do this: See the section "Instructions for Managing Disk Space on the 3640", page 55. These instructions should be used if the 3640 is the first machine at a site, if it is a new machine at an existing site, or if Release 6.0 is being installed on an existing machine at an existing site. If you do not have a 3640 or if you have a 3640 with one or more 167-Mbyte disks or two or more 140-Mbyte disks, these steps are unnecessary.

Installing the first Symbolics machine at a site

To install the first Symbolics machine at a site: See the section "Setting up a New Site", page 91.

Adding a new machine to an existing site

To add a new machine to an existing site: See the section "Adding a New Machine to an Existing Site", page 93.

Upgrading to Release 5.3

To run Release 5 and Release 6.0 concurrently, the Release 5.3 patches must be installed on any machines that will continue to run Release 5. See the section "Installing Release 5.3", page 99.

Installing Release 6.0 at an existing site

To install Release 6.0 on existing machines at an existing site: See the section "Installing Release 6.0", page 101.

To perform these installations, it is strongly recommended that you be familiar with the use of Symbolics machines. You should at least understand how to use the following:

- the Lisp software system
- the Zmacs editor
- the Front End Processor (FEP)

If you are not familiar with these: See the document *User's Guide to Symbolics Computers*. The User's Guide provides an introduction to Symbolics computers and presents the most commonly used information. For an explanation of some of the

terms that are used in the software installation process: See the section "Software Installation Terminology", page 143.

In general, the instructions in this document apply to 3600s, 3670s, and 3640s. The two exceptions are managing disk space on a 3640 and determining the correct microcode. Whenever the procedure differs because of the model, the documentation describes the necessary steps for you to follow.

Before enabling the ephemeral-object garbage collector in Release 6.0, it is recommended that you run the EGC diagnostic. See the section "Running the Ephemeral-object Garbage Collector Diagnostic", page 137.

17. Setting up a New Site

Before setting up a new site, you must register the site in a global registry maintained by Symbolics. To register your site, please contact Symbolics Software Support at 800-441-6010.

In outline, the procedure for setting up a new site is as follows:

1. Register your site, if you have not already done so.
2. Decide which machine at your site will be the namespace server. The namespace server must be a Symbolics computer.
3. Decide what parts of the Symbolics software sources and online documentation you want to have available online. (This determines the amount of disk space the sys host must have). For more information: See the section "Preparing the Sys Host", page 121.
4. Decide which machine will be the sys host (the file server where system sources and other information are stored centrally). The file server can be a Symbolics computer or a VAX computer running VAX/VMS or UNIX.
5. Bring up Release 6.0 on the namespace server. When you use the Set Site command, follow the procedure for configuring a namespace server at a new site. See the section "Procedures for Installing Release 6.0", page 104. You start at step 10 for the Release 6.0 installation procedures. See the section "Set Site Command", page 139.
6. If the sys host is a Symbolics computer, bring up Release 6.0 on the sys host.
7. If there are additional machines at the site, bring them up after the namespace server and sys host. See the section "Adding a New Machine to an Existing Site", page 93.

It is possible for a site to consist of only one machine if its disks are large enough to hold the desired system sources, online documentation, and system files (world loads, paging files, and so on).

18. Adding a New Machine to an Existing Site

New Symbolics computers are shipped from the factory with release 6.0 preinstalled on the FEP disk. However, you must perform a configuration procedure, described in this section, before using the machine. This procedure is only part of the procedure needed if the machine is the first one at a new site. (See the section "Setting up a New Site", page 91.)

Before performing the configuration procedure, register the machine as a new host in the site's namespace. See the section "Registering Hosts", page 72.

The configuration procedure for adding a new machine to an existing site adds site-specific information and customizations to the preinstalled software. The site-specific information tells the machine its site's name and the location of the site's namespace server. The customizations include, for example, the installation of optional software products.

The procedure can be done either by copying an already configured world from another machine at the site or by booting the preinstalled world, adding the site-specific information, and saving that world again. If all your site's machines should have identical characteristics (for instance, the same optional software installed), it is easiest to copy a world from another machine.

The procedure differs for machines that have less than 167 Mbytes of total disk capacity (for example, model 3640 computers with single 140-Mbyte disks).

You need to perform only one of the procedures described here, depending on your circumstances.

18.1 Copying Worlds to Machines with Large Disks

Perform the procedure as follows if you want to copy a world from another machine at the site and the new machine's disk capacity is at least 167 Mbytes:

1. Boot the distribution world on the new machine:

```
Clear Machine
Load Microcode microcode-file-name
Load World distribution-world-file-name
Set Chaos-Address this-machine's-chaos-address
Start
```

For example, *microcode-file-name* might be >tmc5-io4-mic.mic.319 and *distribution-world-file-name* might be >release-6-0.load. A possible value of *this-machine's-chaos-address* might be 402.

2. Enter the Set Site command and follow the instructions for configuring a non-namespace host at an established site. See the section "Set Site Command", page 139.
3. Use the Copy World command on the new machine, to transfer a configured world from another machine at the site. Your machine's disk should have enough unused space for the copied world (about 30,000 blocks). (If the preinstalled software you booted is a version earlier than release 6.0, you can use the function **si:receive-band** instead of Copy World.) For example:

```
Copy World RED|FEP0:>configured-world.load
```

Here, the already configured world on host RED is copied to the machine on which this command is entered.

4. Edit (create) the file FEP0:>boot.boot so that it contains the commands used in step 1 and so that the Load World command uses the pathname for the newly copied world. The file should look like this:

```
Clear Machine
Load Microcode microcode-file-name
Load World configured-world-file-name
Set Chaos-Address this-machine's-chaos-address
Start
```

5. Halt the machine and boot the new world. You can then delete and expunge the preinstalled world, to free up space on your disk.

18.2 Copying Worlds to Machines with Small Disks

Perform the procedure as follows if you want to copy a world from another machine at the site and the new machine's disk capacity is less than 167 Mbytes:

1. Boot the distribution world on the new machine manually, using this sequence:

```
Clear Machine
Load Microcode microcode-file-name
Load World distribution-world-file-name
Clear Paging
Add Paging >page.page
Set Chaos-Address this-machine's-chaos-address
Start
```

Here, *distribution-world-file-name* is the name of the world distributed on the disk (for example, >release-6-0.load). The command sequence leaves the file FEP0:>aux.page.1 unused; it will be used to receive the copied world.

2. Rename the auxiliary file to whatever name you want for the copied world; for example:

```
Rename File FEP0:>aux.page.1 FEP0:>configured-release-6.load.1
```

3. Enter the Set Site command and follow the instructions for configuring a non-namespace host at an established site. See the section "Set Site Command", page 139.
4. Now use the Copy World command on the new machine, to transfer a configured world from another machine at the site. (If the preinstalled software you booted is a version earlier than release 6.0, you can use the function **si:receive-band** instead of Copy World.) For example:

```
Copy World RED|FEP0:>configured-world.load FEP0:>configured-release-6.load
```

Here, the already configured world on host RED is copied to the renamed auxiliary file. Since you are copying the world into an existing file, you are asked how you want to proceed; the correct answer is Overwrite.

5. The machine's disk now has the configured world; rename the distribution world-load file to aux.page:

```
Rename File FEP0:>distribution-world-file-name FEP0:>aux.page
```

6. Edit (create) the file FEP0:>boot.boot so that the Load World command uses the pathname for the newly copied world. The file should look like this:

```
Clear Machine
Load Microcode microcode-file-name
Load World configured-world-file-name
Clear Paging
Add Paging >page.page
Add Paging >aux.page
Set Chaos-Address this-machine's-chaos-address
Start
```

This boot file uses the copied, configured world and uses both the standard file page.page and the renamed world load (aux.page) as paging space.

7. Halt the machine and reboot it using the new boot file.

18.3 Configuring Preinstalled Worlds on Machines with Large Disks

If there is no other machine at the site with a world suitable for copying (for instance, if all other machines have a Fortran compiler saved in their worlds, but you want this one not to have it), you can configure the new machine with the distributed world and the Set Site command. Proceed as follows if the machine has at least 167 Mbytes of disk space:

1. Boot the distribution world on the new machine. (See the section "Booting a World", page 59.)

2. Enter the Set Site command and follow the instructions for configuring a non-namespace host at an established site. See the section "Set Site Command", page 139.
3. Make any other customizations you want for this world and then enter the Save World command to save the configured world. For example:


```
Save World FEP0:>configured-release-6.load
```
4. Edit the file FEP0:>boot.boot so that it loads the saved, configured world.
5. Halt and reboot the machine, using the new boot file. You can now delete and expunge the distribution world load, to free up disk space.

18.4 Configuring Preinstalled Worlds on Machines with Small Disks

If there is no other machine at the site with a world suitable for copying (for instance, if all other machines have a Fortran compiler saved in their worlds, but you want this one not to have it), you can configure the new machine with the distributed world and the Set Site command. Proceed as follows if the machine has less than 167 Mbytes of disk space:

1. Boot the distribution world on the new machine manually.

```
Clear Machine
Load Microcode microcode-file-name
Load World distribution-world-file-name
Clear Paging
Add Paging >page.page
Set Chaos-Address this-machine's-chaos-address
Start
```

Here, *distribution-world-file-name* is the name of the world distributed on the disk (for example, >release-6-0.load). The command sequence leaves the file FEP0:>aux.page.1 unused; it will be used to receive the copied world.

2. Rename the auxiliary file to whatever name you want for the customized world; for example:


```
Rename File FEP0:>aux.page.1 FEP0:>configured-release-6.load.1
```
3. Enter the Set Site command and follow the instructions for configuring a non-namespace host at an established site. See the section "Set Site Command", page 139.
4. Enter the Save World command to save the configured world. For example:


```
Save World FEP0:>configured-release-6.load
```

5. The machine's disk now has the configured world; rename the distribution world-load file to aux.page:

```
Rename File FEP0:>distribution-world-file-name FEP0:>aux.page
```

6. Edit or create the file FEP0:>boot.boot so that the Load World command uses the pathname for the newly copied world. The file should look like this:

```
Clear Machine  
Load Microcode microcode-file-name  
Load World configured-world-file-name  
Clear Paging  
Add Paging >page.page  
Add Paging >aux.page  
Set Chaos-Address this-machine's-chaos-address  
Start
```

This boot file uses the copied, configured world and uses both the standard file page.page and the renamed world load (aux.page) as paging space.

7. Halt and reboot the machine, using the new boot file.

19. Installing Release 5.3

Release 5.3 patches should be installed if you are using a 6.0 namespace server and are planning to run Release 5 at any point in the future on the same network. All the machines that will continue to use Release 5 must be upgraded to Release 5.3 before Release 6.0 is installed. Release 5.3 should not be installed on the namespace server.

The steps for upgrading to Release 5.3 are as follows:

- Backup the LMFS and verify the tape for the machine on which the Release 5.3 will be installed. This step is very important to maintain the integrity of your system. Performing backups ensures that you can restore your file system if any errors should occur. See the section "Back up the Files on Your Machine", page 134.
- Insert the Release 6.0 documentation tape onto the cartridge tape drive.
- To load the patch files, use **dis:load-distribution-tape**, as shown in the following example:

```
(dis:load-distribution-tape)
```

Specify the tape host and press the RETURN key. This pops up a menu specifying the tape contents. For more information about this function: See the function **dis:load-distribution-tape**, page 145.

Click on [Load] for the following:

```
System 242.358
Zmail 83.61
Hardcopy 20.10
```

Click on [Skip] for all the rest.

Click on [Do It] at the bottom of the menu.

- Use **load-and-save-patches** to save these patches into the world. If your machine runs out of room, you will be prompted with a set of options. If you are unsure of which option to choose, it is suggested that you choose the "S" option: "Selectively delete load files". However, you should not delete the current world that you are running. When the patches have been saved into the world, the new world will appear as

```
Release-5-3.load.1
```

You will be asked about updating the default boot file to reflect the 5.3 world. Answer [Yes].

- Remove the tape from the tape drive.
- This 5.3 world can now be copied to other machines with **si:transmit-band**; **si:receive-band** can be used from the other machines to copy the 5.3 world to those machines.

20. Installing Release 6.0

Before installing Release 6.0, any machine that will continue to run Release 5 should be upgraded to Release 5.3. To run Release 5 in a network with a Release 6.0 namespace server, Release 5.3 must be used. See the section "Installing Release 5.3", page 99.

After upgrading machines to Release 5.3, if this was necessary, you can install Release 6.0.

20.1 Introduction to Release 6.0 Installation

This section provides an introduction to the Release 6.0 installation instructions.

It is recommended that you run Release 6.0 on all machines on a network. The namespace server should be the last machine to be upgraded to Release 6.0.

If you must run Release 5 on any machine in your network, it is necessary to update that machine to Release 5.3 before Release 6.0 is installed on the namespace server. The Release 5.2 namespace system is incompatible with Release 6.0; therefore you can not run 5.2 on any machine in a network with a Release 6.0 namespace server. The Release 5.3 patches are distributed on the Release 6.0 documentation tape.

Release 6.0 has a command processor that accepts commands and then executes the command. These commands can be used instead of Lisp forms in Lisp Listeners and **break** loops. After Release 6.0 is installed, you can get a list of the available commands by typing Help to the command prompt. For information about the command processor, See the section "Communicating with the Lisp Machine" in *User's Guide to Symbolics Computers*.

20.1.1 Release 6.0 Distribution Tapes

The Release 6.0 distribution includes the following tapes:

- 2 world load tapes that contain the Release 6.0 microcodes and world load. The labels on these tapes are:

Release 6.0 Distribution World 1/2 IFS Format
Copyright (c) 1985, 1984, 1983, 1982, 1981, 1980
Symbolics, Inc. All Rights Reserved. Portions
Copyright (c) 1980, Massachusetts Institute of
Technology. All rights reserved.
Release 6.0 Dist. World 1/2 3/22/85 TD 995276

Release 6.0 Distribution World 2/2 IFS Format
Copyright (c) 1985, 1984, 1983, 1982, 1981, 1980
Symbolics, Inc. All Rights Reserved. Portions
Copyright (c) 1980, Massachusetts Institute of
Technology. All rights reserved.
Release 6.0 Dist. World 2/2 3/22/85 TD 995276

- 1 source distribution tape that contains the system sources. The label on the tape is:

Release 6.0 Sources Distribution Format
Copyright (c) 1985, 1984, 1983, 1982, 1981, 1980
Symbolics, Inc. All Rights Reserved. Portions
Copyright (c) 1980, Massachusetts Institute of
Technology. All rights reserved.
Release 6.0 Sources 3/22/85 TD 995277

- 1 documentation distribution tape that contains the 5.3 patches, the EGC-DIAG (a diagnostic), documentation, examples, and the graphic-lisp system. The label for this tape is:

Rel 6.0 Doc&Examples Copr.(c)1985,1984,1983,1982,
1981,1980 Symbolics,Inc.All Rights Reserved.
Portions of font library Copr.(c)1984 Bitstream Inc.
All Rights Reserved.Portions Copr.(c)1980 Massachu-
setts Institute of Technology.All Rights Reserved.
Rel 6.0 Doc&Examples 3/22/85 TD 995278

- VAX customers only: Chaosnet distribution tape for VAX/VMS. The label on the VAX/VMS tape is:

Rel. 6 VAX/VMS V3 Network Software
Copyright (c) 1985, 1984, 1983, 1982
Symbolics, Inc. All Rights Reserved.
995176 3/22/85 Backup Format
9 Track, 1600 BPI

- Chaosnet distribution tape for UNIX. The label on the UNIX tape is:

Rel. 6 Unix 4.2BSD Network Software
Copyright (c) 1985, 1984, 1983, 1982
Symbolics, Inc. All Rights Reserved.
995179 3/22/85 TAR Format
9 Track, 1600 BPI

- Some layered products are being distributed with Release 6.0. Any layered product tapes you receive are also in distribution tape format. If you did not receive a complete shipment, please contact your local sales representative.

20.1.2 Installing Release 6.0 on a 3640 with One 140-mbyte Disk

If you have a 3640 with a 140-Mbyte disk, it is necessary to manipulate the available space on the disk before you can begin the installation. For information on how to do this: See the section "Instructions for Managing Disk Space on the 3640", page 55.

20.2 Release 6.0 Prerequisites

This section provides a set of prerequisites for installing Release 6.0. Please be sure that all the prerequisites have been satisfied before attempting to install Release 6.0.

As was explained at the beginning of the document, you should be familiar with the operation of Symbolics computers before attempting to installing Release 6.0. If you are not familiar with them, you should familiarize yourself with them before beginning this installation. See the document *User's Guide to Symbolics Computers*. This book provides an introduction to Symbolics computers and presents the most commonly used information.

If you have a 3640, you will need to use different procedures to manage your paging space. See the section "Instructions for Managing Disk Space on the 3640", page 55.

The prerequisites are as follows:

- Install Release 5.2 on the system to be converted if you have not already done so. If you are unable to install Release 5.2, contact your local office for help.
- At least one machine at the site must have a cartridge tape drive.
- Familiarize yourself with the concepts and terminology associated with the namespace database: See the section "Software Installation Terminology", page 143.
- Make sure that the appropriate file servers for this release have been installed on any timesharing hosts at the site referenced by the Lisp Machine. The documentation cannot be installed without this.
- Back up and verify any local file system(s) on the Lisp Machines to be converted; ideally the backup should be completed just before the conversion starts. See the section "Back up the Files on Your Machine", page 134.
- Allocate sufficient space on your SYS host for the system sources. If you want

to run Release 5 and 6.0 side-by-side at your site, you should set up an alternative source hierarchy for 6.0; this alternative hierarchy should have sufficient space to accommodate the 6.0 sources. See the section "Preparing the Sys Host", page 121.

- Allocate sufficient space for the Release 6.0 distribution in the FEP file system of the machine on which the release will first be installed. The Release 6.0 distribution world needs at least 24,480 (decimal) blocks; the customized world created from the distribution world needs 30,000 (decimal) blocks. The microcode file needs 200 (decimal) blocks.
- You should ensure that all the Lisp Machines are at the appropriate ECO level. Release 6.0 requires at least V17 proms. Release 5 also required V17 proms; thus if you were running Release 5, you will have this version. If you use serial lines, you will need at least V22 proms. If you are unsure of your configuration, check with your local Customer Service representative for this information.
- Determine which file server at your site is going to be the SYS host for the sources. This server can either be another Lisp Machine with a larger disk (474 Mbyte or 335 Mbyte) or a VAX host. Please be sure that the host you choose has room for the 30 Mbytes of information that the Source tape and Documentation and Examples tape contain if you intend to load both tapes. The Source tape alone requires about half as much room.

After you have restored the new release from tape, back up this world and microcode before saving any major applications onto it. Symbolics might distribute additional patch tapes to you and cannot guarantee that you can safely load patches on top of applications.

The following sections provide additional information which you might find useful:

- For information about how to register users and hardware: See the section "Registering Users and Hardware in the Site's Namespace", page 71.
- For templates containing namespace attributes for new Lisp Machine, UNIX, and VMS hosts: See the section "Namespace Attributes for New Hosts", page 79.

20.3 Procedures for Installing Release 6.0

This section describes the procedures for installing Release 6.0. Before installing Release 6.0, it is essential that certain prerequisite conditions be met. See the section "Release 6.0 Prerequisites", page 103. You should only install Release 6.0 if you meet the prerequisites described there.

It is recommended that you run Release 6.0 on all your machines. However, if you decide to continue running Release 5 on some of your machines, you will need to upgrade those machines to Release 5.3; the patches for 5.3 are included with the Release 6.0 distribution.

This set of procedures does not describe all the possible situations that exist at user sites. If all the prerequisites are met, you should be able to install Release 6.0 with these procedures. If you have an unusual situation at your site that is not covered by these procedures, you should contact your local Field Software Support representative for help in installing Release 6.0. See the section "Field Software Support Offices", page 153.

20.3.1 Installation Checklist

The following checklist is a series of steps to follow to install Release 6.0 on the first machine at a site. This checklist should be used to provide an overall view of the Release 6.0 installation process. It is not intended to be a complete description of the installation process. For detailed information and examples for each step shown in this checklist: See the section "Expanded Release 6.0 Installation Instructions", page 108. For example, for detailed information on step 3 in this checklist, refer to Step 3 in the expanded description section.

If you are installing Release 6.0 on a 3640 with one 140-Mbyte disk, you must first allocate enough space in the FEP file system for the new release. See the section "Instructions for Managing Disk Space on the 3640", page 55.

If Release 5 will be running on any machines after Release 6.0 is installed on the namespace server, those machines must be upgraded to Release 5.3. See the section "Installing Release 5.3", page 99.

The procedures as follows:

1. Install Release 5.3 on any machines that will continue running Release 5.
2. Boot your machine with Release 5.2.
3. Login to the SYS host using **si:login-to-sys-host**.
4. Edit the following files:
 - SYS.TRANSLATIONS file
 - FEPn:>BOOT.BOOT file
 - SYS:PRESS-FONTS file (only if you have an LGP on your network)
5. Back up the LMFS on your machine and verify the tape.
6. VAX customers only. Install the new Chaosnet package, which is on a separate tape.
7. Preallocate space in the FEP system for the new Release 6.0 files.
8. Logout and halt your machine.
9. Disk restore both Release 6.0 microcode and world load files from the distribution tapes.
10. Boot the Release 6.0 world. When the machine comes up, you will notice two distinct differences: the Release 6.0 Lisp Listener display is significantly

different than the Release 5 Lisp Listener and Release 6.0 supports a command processor. For more information about using the command processor (CP): See the section "Communicating with the Lisp Machine" in *User's Guide to Symbolics Computers*.

11. Use the CP Set Site command to configure the world to your site.
12. Use the CP Save World command to save the world with your local site information.
13. Boot the newly saved world.
14. Use **tape:write-fep-files-to-tape** to produce a backup copy of the new Release 6.0 world.
15. Load the files from the documentation tape and the source tape. The documentation tape contains the 5.3 patch files, the ephemeral-object garbage collector diagnostic files, and the documentation files. You must load the ephemeral-object garbage collector diagnostic files. Loading the documentation files and the sources files is optional. Use **dis:load-distribution-tape** to load the files from the tapes.
16. At this point you can either distribute the configured Release 6.0 world or you can modify it further for your site.
17. Optionally, you can install any layered products and make any site modifications (such as make-systems) that you might have. This is necessary only if you have additions to make to the Release 6.0 world.
18. Distribute the modified world to other machines.
19. Run the diagnostic before enabling the ephemeral garbage collector. **NOTE:** You must run the diagnostic on each machine that will be running 6.0 before enabling the ephemeral garbage collector.

At this point, assuming that all steps have been followed and all machines have been upgraded to either 5.3 or 6.0, the installation is complete.

20.3.2 Expanded Release 6.0 Installation Instructions

The following checklist provides an expanded set of instructions for installing Release 6.0 on the first machine at a site. The steps in this checklist provide details and examples for the steps in the installation process.

If you are installing Release 6.0 on a 3640 with one 140-Mbyte disk, you must first allocate enough space in the FEP file system for the new release. See the section "Instructions for Managing Disk Space on the 3640", page 55.

If Release 5 will be running on any machines after Release 6.0 is installed on the namespace server, those machines must be upgraded to Release 5.3. See the section "Installing Release 5.3", page 99.

The procedures are as follows:

1. *Install Release 5.3 on any machines that will continue running Release 5.*

See the section "Installing Release 5.3", page 99.

2. *Boot your machine with Release 5.2.*

If you need information about booting your machine: See the document *User's Guide to Symbolics Computers*.

3. *Login to the SYS host using **si:login-to-sys-host**.*

Login as the default system user, using **si:login-to-sys-host**, as shown in the following example:

```
(si:login-to-sys-host)
```

This function logs into the SYS host as the default user and is useful during installation procedures when you must do certain operations after cold booting.

4. *Edit the following files:*

SYS.TRANSLATIONS file

The SYS translations file specifies the pathnames for various system files. This file must be updated to reflect the Release 6.0 files and directories and pathnames. For information about editing the SYS.TRANSLATIONS file: See the section "Updating the Sys Translations File", page 131. For complete information about planning your SYS host: See the section "Preparing the Sys Host", page 121.

FEPn:>BOOT.BOOT file

Edit your default boot file to reflect the Release 6.0 distribution world and microcode 319:

```

Clear Machine
Load Microcode FEP0:>TMC5-IO4-ST506-MIC.MIC.319
Load World FEP0:>Release-6-0.load.1
Set Chaos-Address 12345                (Puffin)
Start

```

NOTE: Use the correct microcode for your machine. See the section "Release 6.0 Microcode Types", page 139.

SYS:PRESS-FONTS file {only if you have an LGP on your network}

If you have the Fonts Widths File entry in the namespace object pointing to SYS: PRESS-FONTS; you must change that entry because this directory no longer exists on the tape being distributed. See the section "Lgp Example", page 142.

5. *Back up the LMFS on your machine and verify the tape.*

It is important that you back up any local file system(s) on the Lisp Machines to be upgraded before the conversion starts. For information on backing up your file system: See the section "Performing Dumps", page 6.

To verify the tape, click on [Local FS Maint] in the File System Editor, which pops up a menu. Click on [Reload/Retrieve], which pops up another menu. Click on [Compare] to compare the files on tape to the files stored on the machine.

6. *VAX customers only. Install the new Chaosnet package, which is on a separate tape.*

If you use VAX/VMS or VAX/UNIX Chaosnet, you must install the new Chaosnet package before proceeding with the installation. The Chaosnet software is on a separate distribution tape that is included with Release 6.0. For instructions about installing the Chaosnet software, refer to the appropriate document:

Symbolics VAX/VMS Chaosnet Software Installation Guide (for VAX/VMS V3.x)
 Symbolics UNIX Chaosnet Software Installation Guide

These documents are included with your Release 6.0 shipment.

7. *Preallocate space in the FEP system for the new Release 6.0 files.*

These instructions do not apply to 3640's with single 140-Mbyte disks.

In order to load the *Release 6.0 World Load* tape set, you must have a world load file at least 24480 blocks long. You can reuse an existing world, or you

can create a new file of the required size. You must also have a microcode file that is at least 200 blocks long and which has the same name as the microcode for the machine. See the section "Release 6.0 Microcode Types", page 139.

If you are reusing a world, first rename it to >Release-6-0.Load.1. If it needs to be larger, you can grow it with the function below.

```
(defun grow-fep-file (name length)
  (with-open-file (stream name :direction :block :if-exists :overwrite)
    (send stream :allocate length)))
```

The code below grows a world to the required size for Release 6.0, using the above function:

```
(grow-fep-file "fep0:>Release-6-0.load.1" 24480.)
```

If possible, allocate a different file than the Release 5 world you are currently using, so that you are able to boot the machine if something goes wrong in the middle of the procedure.

If you are installing the release on a machine without a tape drive, via the network, it is essential that you do not reuse your current world file. Failure to observe this precaution may necessitate a service visit to restore software to your machine.

8. Logout and halt your machine.

To logout, type the following to a Lisp Listener:

```
(logout)
```

To halt your machine, type the following to a Lisp Listener:

```
(si:halt)
```

9. Disk restore both the Release 6.0 microcode and world load files from the distribution tapes.

Symbolics distributes microcode and world loads for this release on two tapes. The first tape contains the following:

- A copy of several different microcodes; a different microcode is supplied for each configuration
- The first part of the new release world

The second tape contains the second part of the new release world.

Installing the new release requires (in the simplest case) reuse of one FEP

microcode file and two FEP world load files. The steps below assume two FEP world load files can be made available for reuse, either because they contain old information (an obsolete world, for example) or because they are duplicated on some other machine and can be retrieved from that machine after the new release installation procedure has been completed.

After installation is complete, the new release requires one world load and one microcode's worth of space in the FEP file system.

The FEP world load file reused to load the new release world distributed from Symbolics must be at least 24,480 (decimal) disk blocks in size. The microcode file should be 200 (decimal) disk blocks in size. The second FEP world load file that will be reused will contain a new release world that you have customized for the site and should be at least 30,000 (decimal) disk blocks in size.

In general, disk restoring the Release 6.0 world and microcode is done as follows. Insert the first Release 6.0 world tape into the cartridge tape drive. Use the FEP command Disk Restore to restore the Release 6.0 microcode and the Release 6.0 world tape. You will need to insert the second Release 6.0 world tape during the restoration.

This is described in detail in the following set of procedures:

- Before halting the machine, write down the names of the FEP files to be reused. You must create the microcode file before you can disk restore the microcode. For example, if you are restoring microcode FEP0:>tmc5-io4-st506-mic.mic.319, you must have a file with this name in the FEP file system before you can restore the microcode.

- Write down the Chaosnet address for the machine; you will need it to boot the new release world manually. To determine the Chaosnet address, use the following form:

```
(hostat neti:*local-host*)
```

- Logout and halt the machine:

```
(logout)  
(si:halt)  
Fep>
```

- Insert the first tape into the cartridge tape drive. Clear the machine and load the default microcode that is already on the machine:

```
Fep>Clear Machine  
Fep>Load Microcode
```

The Clear Machine command should take less than a minute. The more memory on the machine, the longer this step takes to clear the memory.

After the machine clears, a "Machine halted" message is displayed; the FEP prompt then returns.

- Restore the microcode from tape with the FEP Disk Restore command.

Fep>Disk Restore

This command asks several questions about the tape.

- The first question:

Have you done a SET DISK-TYPE for each disk unit that does not have a valid label block on it?

asks whether all disks to be used during the restoration have valid labels on them; when installing a new release, you should always answer Y (for Yes).

- The next question:

Is there a microcode file at the beginning which should be skipped?

asks whether there is microcode at the start of the tape that should be skipped. Always answer N when installing a new release.

- Then, after descriptive information is printed out for each microcode type, you are asked if you want to restore that microcode. This part of the dialogue is repeated for each possible microcode type, that is:

- 3600, 3640, and 3670
- 3600, 3640, and 3670 with IFU
- all of the above both with and without FPA microcode

Answer Y for the microcode that is appropriate for your machine; answer N for all the other microcodes. For a list of all the types of microcode: See the section "Release 6.0 Microcode Types", page 139.

- When you answer yes to the restore prompt, you are prompted for the name of the existing file into which the microcode should be restored:

To file...

Enter the name of the microcode file chosen for this purpose earlier. Case is unimportant; you can use either upper- or lowercase.

The restoration of the microcode should take less than a minute after the microcode file name is given. One or two numbers will be printed during the actual restoration.

This example shows part of a sample dialogue produced by using the

FEP Disk Restore command when restoring a world load from tape onto a 3640 running Release 5.2. In this example, for the sake of brevity we do not show the dialogue for most of the types of microcode. Answer Y to restore the proper microcode; for this example, tmc5-io4-st506-mic.mic.319 is used. Remember that you must choose the right microcode type to accommodate your machine type:

```
Halt machine
Fep>clear machine
Fep>load microcode fep0:>tmc5-io4-mic.mic.296
Fep> disk restore
Have you done a SET DISK-TYPE for each disk unit that does not
have a valid label block on it? (Y or N) Y
Is there a microcode file at the beginning which should be skipped?
(Y or N) N
File restore. Information from tape:
```

```
Filename: SYS:L-UCODE; tmc5-mic.mic.319
Length: 120735 bytes (105 blocks)
Created: 1/29/85 02:28:17
Author: pepper
Comment: tmc5-mic 319 (Rel 6.0)
Restoration Comment: Microcode for the 3600.
If need for this machine,
please load into FEP0:>tmc5-mic.mic.319.
Restore it? (Y or N) N
File Restore. Information from tape:
```

continuation of microcode restoration requests

```
Filename: SYS:L-UCODE; tmc5-io4-st506-mic.mic.319
Length: 121525 bytes (106 blocks)
Created: 1/29/85 00:49:36
Author: pepper
Comment: tmc5-io4-st506-mic 319 (Rel 6.0)
Restoration Comment: Microcode for the 3640.
If need for this machine,
please load into FEP0:>tmc5-io4-st506-mic.mic.319.
Restore it? (Y or N) Y
To file FEP0:>tmc5-io4-st506-mic.mic.319
0 50 100
File Restore. Information from tape:
```

continuation of microcode restoration requests


```

Filename: SYS:L-UCODE; ifu-io4-st506-fpa-mic.mic.319
Length: 121338 bytes (106 blocks)
Created: 1/29/85 01:54:20
Author: pepper
Comment: tmc5-io4-st506-fpa-mic 319 (Rel 6.0)
Restoration Comment: Microcode for the 3640 with FPA.
If need for this machine,
please load into FEP0:>ifu-io4-st506-fpa-mic.mic.319
Restore it? (Y or N) N
File Restore. Information from tape:

```

```

.
.
.

```

NOTE: You should answer only as you are prompted for each microcode type and avoid typing ahead of the questions. If you type ahead answering N, you could easily answer no to the next question about restoring the world load. If you do that, you will have to begin the Disk Restore procedure from the beginning. The last microcode type for which you are prompted is:

```
ifu-io4-st506-fpa-mic.mic.319
```

Next, this example continues with the new release restoration by loading the first part of the world load. Answer Y when asked if the world load file should be restored, then supply the name of the FEP file that will hold the new release world distributed from Symbolics. This file name is *not* the name of the FEP file that will hold your customized world. Just as in the microcode restoration, after every three hundred blocks of restoration the block number is displayed as a progress report.

```

Filename: FEP0:>Release-6-0.load.1
Length: 26802432 bytes (24488 blocks)
Created: 3/18/85 01:58:03
Author: pepper
Comment: release-6.0 Distrib
Restoration Comment: Distribution World for Release 6.0.
Requires 24480 blocks.
Restore it? (Y or N) Y
To file FEP0:>Release-6-0.load.1
0 50 100 150 200...

```

This restoration takes much longer, perhaps 10 minutes. Because the new release world cannot fit on one tape, it eventually runs out of restoration tape. At that time a message is printed and the standard FEP command loop is resumed.

- Place the second world load tape in the drive and use the Disk Restore

command again. Answer Y again to the label question; answer N to the microcode question; then, answer Y to the restoration query. Answer the question about the target file with the same FEP file name used in restoring the first part of the tape.

```
End of Restoration Tape Encountered.
Fep> disk restore
Have you done a SET DISK-TYPE for each disk unit that
does not have a valid label block on it? (Y or N) Y
Is there a microcode file at the beginning that should
be skipped? (Y or N) N
File Restore. Information from tape:
```

[Continuation from previous tape]

```
Filename: FEP0:>Release-6-0.load.1
Length: 28200960 bytes (24480 blocks)
Created: 3/18/85 01:58:03
Author: pepper
Comment: release-6.0 Distrib
Restoration Comment: Distribution World for Release 6.0.
24488 blocks.
Restore it? (Y or N) Y
To File: FEP0:>Release-6-0.load
15200 15250 15300...24450 0
End of Restoration Tape Encountered.
Fep>
```

At the end of this process (which should take no more than five minutes) the restoration completes and the FEP prompt appears.

10. *Boot the Release 6.0 world. You should use the boot file that you edited in step 4, unless this machine was shipped with Release 6.0, in which case you can just use the FEP Boot command.*

When the machine comes up, there will be two immediate differences between the Release 5 and Release 6.0 Lisp Listener windows:

- The Release 6.0 herald is significantly different than the Release 5 Lisp Listener.
- The Release 6.0 Lisp Listener window shows a command prompt. At this point, you can begin using commands, rather than having to type Lisp forms. For more information about using the command processor (CP): See the section "Communicating with the Lisp Machine" in *User's Guide to Symbolics Computers*.

11. *Use the CP Set Site command to configure the world to your site.*

The Set Site command starts a dialogue to set the current site to be *site name*. This command is used to configure the software and identify your machine before you use a new world load. It should be the first thing you type to your machine after booting the new software.

When a new world is booted for the first time, the herald gives the machine name as *DIS-LOCAL-HOST*. You are prompted in the course of the Set Site dialogue for a name for the machine.

An example of using the Set Site command during the installation of Release 6.0 at an existing site is shown below.

Use this dialogue if you are installing new software on a machine at your site that is not the namespace server. What you type is underlined. RETURN means press the key labeled return.

```
Command: Set Site (a site name) downunder
Define a new site named DOWNUNDER (as opposed to looking for an existing
definition of DOWNUNDER on disk)? (Y or N) No
```

You answer no because your site is already defined.

```
What host is the namespace server for DOWNUNDER (default: local): Koala
Chaosnet address for KOALA: 401
Host responds as KOALA, ok? (Y or N) Yes.
The local host is now KANGAROO
Command:
```

You are now ready to use Save World to save the configured Lisp world and reboot the machine.

For additional information about using Set Site is provided in the section: See the section "Set Site Command", page 139.

12. *Use the CP Save World command to save the world with your local site information.*

In the following example the Save World command saves the current world in the file Rel6.load.1:

```
Save World Rel6.load.1
```

13. *Boot the newly saved world.*

14. *Use **tape:write-fep-files-to-tape** to produce a backup copy of the new Release 6.0 world.*

It is important to backup the world at this point. This world represents a Release 6.0 world load that you know can be booted and that has your local site information. If you should later destroy your Release 6.0 world, this

backup will provide a bootable Release 6.0 world that includes your local site configuration.

See the function **tape:write-fep-files-to-tape**, page 20.

15. *Load the files from the documentation tape and the source tape. The documentation tape contains the 5.3 patch files, the ephemeral-object garbage collector diagnostic files, and the documentation files. You must load the ephemeral-object garbage collector diagnostic files. Loading the documentation files and the sources files is optional.*

Type **(dis:load-distribution-tape)** to the Lisp Listener to invoke the distribution loader. The distribution loader uses a menu to control how the sources are loaded. The menu offers choices of how to load each system on the tape. Specify that you want to load all systems (the default) on the menu.

If you experience any problems loading the sources, the distribution loader offers various solutions. Additionally, the HELP key might be useful, and, if you still cannot determine the problem, please call the Symbolics Field Office Software Support representative for your site.

See the section "Loading System Sources and Documentation and Ephemeral-object Garbage Collector Diagnostic", page 135.

16. *At this point you can either distribute the configured Release 6.0 world or you can modify it further for your site.*

At this point the newly disk-saved world is ready to be distributed to other machines on the network. You can either distribute this world or you can continue to modify it by making site modifications or installing layered products. Your decision depends on how the machines on your network are used. For example, some machines might run the unmodified Release 6.0 world, while others might run Release 6.0 with a layered product.

See the section "Distributing the New World and Microcode to Other Machines", page 142.

17. *Optionally, you can install any layered products and make any site modifications (such as make-systems that you might have. This is necessary only if you have additions to make to the Release 6.0 world.)*

For information about installing layered products, refer to the installation instructions that accompany the product.

To distribute the modified worlds to other machines, use the Save World and Copy World commands to save the world and copy it to other machines. See the section "Save World Command" in *User's Guide to Symbolics Computers*. See the section "Copy World Command" in *User's Guide to Symbolics Computers*.

18. *Distribute the modified world to other machines.*

To distribute the new Release 6.0 world and the accompanying microcode, use the Copy World Command to copy the world to the specified machine. In the following example, the world Rel6.load.1 would be copied to the host Puffin.

```
Copy World Rel6.load.1 Puffin
```

For more information: See the section "Copy World Command" in *User's Guide to Symbolics Computers*.

To distribute the new microcode, use the **si:install-microcode** function on the machine that will receive the new microcode.

```
(si:install-microcode 319.)
```

If you have a machine that is running Release 5, use **si:receive-band** to copy the world from another machine to your machine. See the function **si:receive-band**, page 50.

19. *Run the diagnostic before enabling the ephemeral garbage collector. NOTE: You must run the diagnostic on each machine that will be running 6.0 before enabling the ephemeral garbage collector.*

To run this diagnostic, do the following:

- If the diagnostic has not yet been loaded from the documentation tape, use **dis:load-distribution-tape** to load this system. The name of the system is EGC-DIAG.
- Copy the diagnostic from the SYS host to your local machine, using the following form:

```
(si:install-microcode "tmc5-egc-diag-mic.mic")
```

When you are asked about updating your boot file, answer NO.

- Halt your machine, using the Halt Machine command.
- Type the following commands to the FEP:

```
Fep>Clear Machine
Fep>Load Microcode FEP:>TMC5-EGC-DIAG-MIC.MIC
Fep>Load World FEP:> world-load-file-name
Fep>Start
```

world-load-file-name is the name of the world load. The diagnostic can be run with either Release 5.3 or Release 6.0.

- The diagnostic will print out any errors it discovers, and then, finally, will report the total number of errors in the total test. The final line printed is the total.
 - If this number is 0, it indicates that no errors were found in the hardware being tested and the ephemeral-object garbage collector can be used on this machine.
 - If this number is not zero, you should call your local Customer Service Engineer and have the DP and FEP boards checked and possibly replaced. In the meantime, it is important that you do not run the ephemeral-object garbage collector. The ephemeral-object garbage collector is disabled by default.
 - In some cases, the diagnostic finds errors that make it impossible for the diagnostic to continue. When this happens, Lisp halts itself and displays the message "Lisp halted itself" and you are returned to the FEP. Use the Show Status command:

```
FEP>show status
```

Several lines of information are displayed; within this text will be statements phrases about GC parity errors.

This indicates that you have hardware problems. You should call your local Customer Service Engineer and have the DP and FEP boards checked and possibly replaced. In the meantime, it is important that you do not run the ephemeral-object garbage collector.

At this point, assuming that all steps have been followed and all machines have been upgraded to either 5.3 or 6.0, the installation is complete.

21. Related Installation Procedures

21.1 Preparing the Sys Host

When preparing to run Release 6, you must decide what parts of Release 5 you are going to retain and how much disk space you are going to use for Release 6. This section gives you the information to make these choices for your individual site.

Here is an overview of the procedure:

1. Decide how you want to set up the hierarchies. Fill out the checklist in Table 2 on page 126.

See the section "Decide How to Set up the Sys Host", page 121.

See the section "What Parts of Release 5 Do I Need?", page 121.

See the section "What Disk Space Do I Have?", page 124.

See the section "What Part of the Release 6 Distribution Do I Need?"

2. Delete unneeded portions of the Release 5 hierarchy. See the section "Delete Unneeded Parts of Release 5", page 131.
3. Edit SYS: SITE; SYS.TRANSLATIONS to encompass the new situation. See the section "Updating the Sys Translations File", page 131.

21.1.1 Decide How to Set up the Sys Host

Before setting up the SYS host, the following decisions should be made:

1. What parts of the Release 5 hierarchy should be retained?
2. What disk space should be available for system software, data, and sources?
3. What part of the Release 6 distribution is needed?

Finally, you have to determine whether all the files will fit in the available space. If not, one of the above three items has to be adjusted.

21.1.1.1 What Parts of Release 5 Do I Need?

The answer to this depends on how much Release 5 will be used at the site after installing Release 6. If a group at your site is planning to complete a development project in Release 5.2, you might need to keep the entire heirarchy. At the other extreme, if you plan not to run Release 5 at all, you might need to move the contents of SYS: SITE; to a new home; you can then delete the rest of the Release 5 hierarchy.

The following table contains a checklist; it is suggested that you fill out this checklist as you read this section. If you need more information to fill in this checklist: See the section "Description of Release 5 Directories", page 145.

Release 5 Directory Checklist

Circle either *Keep* or *Delete* for each directory below.
Note that you must keep SYS: SITE;.

Directory	Size	Action	
sys: *;	7239 total.		
sys: cc;	49	<i>Keep</i>	<i>Delete</i>
sys: cube;	95	<i>Keep</i>	<i>Delete</i>
sys: debugger;	102	<i>Keep</i>	<i>Delete</i>
sys: demo;	53	<i>Keep</i>	<i>Delete</i>
sys: distribution;	38	<i>Keep</i>	<i>Delete</i>
sys: doc;	2605	<i>Keep</i>	<i>Delete</i>
sys: examples;	195	<i>Keep</i>	<i>Delete</i>
sys: hardcopy;	57	<i>Keep</i>	<i>Delete</i>
sys: io;	213	<i>Keep</i>	<i>Delete</i>
sys: iol;	125	<i>Keep</i>	<i>Delete</i>
sys: l-bin;	14	<i>Keep</i>	<i>Delete</i>
sys: l-compiler;	101	<i>Keep</i>	<i>Delete</i>
sys: l-fep;	24	<i>Keep</i>	<i>Delete</i>
sys: l-sys;	154	<i>Keep</i>	<i>Delete</i>
sys: l-ucode;	286	<i>Keep</i>	<i>Delete</i>
sys: fonts; lgp-1;	583	<i>Keep</i>	<i>Delete</i>
sys: fonts; tv;	210	<i>Keep</i>	<i>Delete</i>
sys: lmfs;	206	<i>Keep</i>	<i>Delete</i>
sys: lmfs-patch;	4	<i>Keep</i>	<i>Delete</i>
sys: lmsym;	40	<i>Keep</i>	<i>Delete</i>
sys: lmtape;	65	<i>Keep</i>	<i>Delete</i>
sys: network;	213	<i>Keep</i>	<i>Delete</i>
sys: patch;	32	<i>Keep</i>	<i>Delete</i>
sys: print;	63	<i>Keep</i>	<i>Delete</i>
sys: site;	3	<i>Keep</i>	
sys: sys;	316	<i>Keep</i>	<i>Delete</i>
sys: sys2;	358	<i>Keep</i>	<i>Delete</i>
sys: ubin;	6	<i>Keep</i>	<i>Delete</i>
sys: ucadr;	100	<i>Keep</i>	<i>Delete</i>
sys: window;	321	<i>Keep</i>	<i>Delete</i>
sys: zmail;	240	<i>Keep</i>	<i>Delete</i>
sys: zwei;	341	<i>Keep</i>	<i>Delete</i>

Table 1. Release 5 Directory Checklist

21.1.1.2 What Disk Space Do I Have?

The Release 6 distribution, including both the sources and documentation tape, is slightly smaller than the Release 5 distribution. If you have all of Release 5 online, you should have little problem finding space for Release 6.

In Release 6, it is no longer necessary to keep all of the distribution on the same physical host. This means that if you do not have sufficient disk space on the host you used as your SYS host in Release 5, you can now look at other hosts on your network and consider using them to hold files for Release 6. This is especially important in finding room for the on-line documentation, which allows use of the Document Examiner.

You can find out how much disk space is free on a Symbolics system by listing any directory on that system. The first or second line of the directory listing will include the number of free records. To find the amount of space available on a VAX, consult your system manager.

Sufficient space for installing Release 6 can come from three places:

- Free space available on your SYS host.
- Space reclaimed by deleting Release 5 materials.
- Free space found on other hosts.

The *Release 5 sources* tape took the following space, most of which can be reclaimed with little inconvenience:

- LMFS - 4350 records
- UNIX - 34330 blocks
- VMS - 35031 blocks

The *Release 5 Documentation and Examples* tape took the following space, which should be reclaimed to make room for the Release 6 documentation and examples.

- LMFS -
 - sys:doc;** 2605 records
 - sys:examples;** 151 records

- UNIX -
 - sys:doc;** 22730 blocks
 - sys:examples;** 969 blocks

- VMS -
 - sys:doc;** 23410 blocks
 - sys:examples;** 1300 blocks

To convert to units of VMS or Unix blocks, multiply LMFS record sizes by 8. The result usually slightly overstates the disk space required.

If you plan to use disk space on more than one host, see Table 2 and work out an allocation of directories to each host. Your task will be made simpler by putting as few directories on Unix as possible. However, you should also take into account loading on the hosts involved and frequency of access of the various directories. Documentation lookup will be much faster if the documentation files are kept on a lightly loaded host, especially a non-timesharing host. This is important because documentation access is very interactive, and the documentation database may be the most heavily used set of files in the distribution.

Write down your choice of host for each directory in the space provided in Table 2. For more detailed information about what each directory is used for: See the section "More Information About Release 6 Directories", page 150.

Directory	LMFS Records	Notes: host chosen
sys: **;	6652	
sys: debugger; **;	94	
sys: doc; **;	2060	
sys: examples; **;	150	
sys: fonts; **;	991	
sys: io; **;	309	
sys: iol; **;	89	
sys: l-ucode; **;	1240	
sys: lmtape; **;	80	
sys: network; **;	234	
sys: patch; **;	77	
sys: sys; **;	102	
sys: sys2; **;	256	
sys: window; **;	315	
sys: zwei; **;	361	
<i>Small Directories</i>	294	

Table 2. Sizes of the larger Release 6 Directories

For more detailed information: See the section "Release 6 Sys Host Size", page 147.

21.1.1.3 What Part of the Release 6 Distribution Do I Need?

The Release 6 distribution consists of two tapes titled *Release 6 Sources* and *Release 6 Documentation and Examples*. We list here the major parts of each tape, the systems that make up these major parts, and an estimate of how much disk space is needed for each major part.

The system descriptions in italics for each entry correspond with the systems that must be selected from the menu when the tape is loaded.

The Release 6 Sources tape contains:

You should use this section as a reference aid in filling out the checklist for this tape. See Table 3, page 129.

- Legal notices. These should always be loaded.

Trade Secret Notice Latest files.

Documentation Font Notice Latest files.

These require 2 LMFS records.

- Microcodes for different hardware configurations, and associated files.

Microcode Auxiliaries Rel 6 0 Latest files.

These require about 1300 LMFS records.

- Fep debuggers, for examining the stack when a machine halts due to an error. You should be sure to load these, and install the proper version on each system's FEP filesystem.

Fep Debug Tools Latest files.

These require about 32 LMFS records.

- A selected set of system sources useful for finding out more about the system.

System 271 Latest sources, patches 271.1 to 271.0.

Lmfs Tools 6 0 Latest files.

Tape 41 Latest sources, patches 41.1 to 41.0.

These require about 2081 LMFS records.

- TV Font files for fonts not pre-loaded into the world.

Tv Fonts 6 0 Latest files.

These require about 300 LMFS records.

- Various patch directories, to facilitate future updates. These should always be loaded.

COMMON-LISP 4 Patches 4.1 to 4.0, PATCHES ONLY.

Zmail 107 Patches 107.1 to 107.0, PATCHES ONLY.

Hardcopy 46 Patches 46.1 to 46.0, PATCHES ONLY.

LMFS 57 Patches 57.1 to 57.0, PATCHES ONLY.

Basic Sage 15 Patches 15.1 to 15.0, PATCHES ONLY.

These require about 10 LMFS records.

- Printer software. This is needed to support spooling to printers, or any printing to LGP printers.

Print 38 Latest sources, system build objects, patches 38.1 to 38.0.

Hardcopy Fonts 6 0 Latest files.

These require about 60 LMFS records of code, plus about 655 LMFS records for the font files.

- Store and forward mailer. This provides the ability to have mailing lists and other mailer services.

Mailer 42 System build objects, patches 42.1 to 42.3.

Dialnet Registry 6 0 Latest files.

This requires about 45 LMFS records.

Release 6 Sources Tape Checklist

Circle either *Load* or *Skip* for each system below.

<i>Load</i>	Trade Secret Notice Latest files.
<i>Load</i>	Documentation Font Notice Latest files.
<i>Load</i>	Microcode Auxiliaries Rel 6 0 Latest files.
<i>Load</i>	Fep Debug Tools Latest files.
<i>Load Skip</i>	System 271 Latest sources, patches 271.1 to 271.0.
<i>Load</i>	COMMON-LISP 4 Patches 4.1 to 4.0, PATCHES ONLY.
<i>Load</i>	Tv Fonts 6 0 Latest files.
<i>Load</i>	Zmail 107 Patches 107.1 to 107.0, PATCHES ONLY.

<i>Load</i>	Hardcopy 46 Patches 46.1 to 46.0, PATCHES ONLY.
<i>Load</i>	LMFS 57 Patches 57.1 to 57.0, PATCHES ONLY.
<i>Load</i>	Lmfs Tools 6 0 Latest files.
<i>Load Skip</i>	Tape 41 Latest sources, patches 41.1 to 41.0.
<i>Load</i>	Basic Sage 15 Patches 15.1 to 15.0, PATCHES ONLY.
<i>Load Skip</i>	Print 38 Latest sources, system build objects, patches 38.1 to 38.0.
<i>Load Skip</i>	Hardcopy Fonts 6 0 Latest files.
<i>Load Skip</i>	Mailer 42 System build objects, patches 42.1 to 42.3.
<i>Load Skip</i>	Dialnet Registry 6 0 Latest files.

Table 3. *Release 6 Sources* Tape Checklist

The Release 6 Documentation and Examples tape contains:

- Release 5.3. You need to load this from a Release 5 world, rather than from a Release 6 world. You should use this section as an aid in filling out the checklist for this tape. See Table 4, page 131.

System 242 Patches 242.354 to 242.358, PATCHES ONLY.

Zmail 83 Patches 83.60 to 83.61, PATCHES ONLY.

Hardcopy 20 Patches 20.9 to 20.10, PATCHES ONLY.

These take around 50 LMFS records.

- Garbage Collector diagnostic. You must load this.

Egc Diag Latest files.

This requires 24 LMFS records.

- Legal notice. This should always be loaded.

Trade Secret Notice Latest files.

This requires one LMFS record.

- Usage examples. This is a set of programs which serve to illustrate how to program in Zetalisp, use the window system, the mouse, and other useful topics.

Examples Rel 6 0 Latest files.

Zmail Examples Latest files.

Graphic Lisp 7 Latest sources, system build objects, patches 7.1 to 7.0.

This requires about 140 LMFS records.

- Online documentation. This is needed in order to look up documentation in the document examiner. If you put this system onto a VAX, be sure to install the new file server software onto the VAX *before* loading this onto the VAX.

Documentation Database 3 System build objects, patches 3.1 to 3.0.

Documentation Fonts Latest files.

This requires about 2082 LMFS records.

- One empty system. Due to an oversight, one system was included with no files.

DISTRIBUTED-UTILITIES .

This takes no LMFS records.

Release 6 Documentation and Examples Tape Checklist

Circle either *Load* or *Skip* for each system below, for loading Release 6.0.

<i>Skip</i>	System 242 Patches 242.354 to 242.358, PATCHES ONLY.
<i>Skip</i>	Zmail 83 Patches 83.60 to 83.61, PATCHES ONLY.
<i>Skip</i>	Hardcopy 20 Patches 20.9 to 20.10, PATCHES ONLY.
<i>Load</i>	Egc Diag Latest files.
<i>Load</i>	Trade Secret Notice Latest files.
<i>Load Skip</i>	Examples Rel 6 0 Latest files.
<i>Load Skip</i>	Zmail Examples Latest files.

<i>Load</i>	<i>Skip</i>	Graphic Lisp 7 Latest sources, system build objects, patches 7.1 to 7.0.
<i>Load</i>	<i>Skip</i>	Documentation Database 3 System build objects, patches 3.1 to 3.0.
<i>Load</i>	<i>Skip</i>	Documentation Fonts Latest files.
	<i>Skip</i>	DISTRIBUTED-UTILITIES .

Table 4. *Release 6 Documentation and Examples* Tape Checklist

21.1.2 Delete Unneeded Parts of Release 5

To delete the release 5 sources you use the Delete File (m-X) command in the editor. *You must be running Release 5 during this operation, to ensure that you will be operating on the right files.* Use the checklist of directories you prepared earlier (see Table 1) and delete the files, using wildcards to delete all the files in the directories. For example:

```
m-X Delete File
SYS:IO;*. *.*
```

This deletes all the files, but does not delete the directories. The directories themselves take up little space, and can be safely left. However, they can also be deleted if you wish, by deleting the directory files on the physical host, by whatever technique is needed for deleting directories on the host involved.

21.1.3 Updating the Sys Translations File

Before installing Release 6.0 at this site, you need to prepare the file which will tell the new world about the new SYS host. To do this, edit the file SYS: SITE; SYS.TRANSLATIONS to know about where the Release 6 sources will be kept. We recommend naming the top level directory "Rel-6", or as close as you can approach this on the individual operating system.

The first step is to *conditionalize* the SYS.TRANSLATIONS file as shown below. Doing this allows you to continue using Release 5 after installing Release 6. This allows Release 5 and Release 6 to coexist, each with their own translations.

Release 6 gets the translation for SYS: SITE from the site object, not from this file, so Release 5 and Release 6 continue to share the site directory.

You may be lucky and already have your site directory named >sys>site>. If so, we recommend leaving it with that name. There is no software restriction as to the name of the site directory in Release 6, but it is helpful to give it a name independent of any individual release.

Note that if you are planning to continue to use Release 5 for a time, it considerably more convenient *not* to change the name of the site directory until you are no longer using Release 5.

If you are not planning to continue to run Release 5, you can create a directory for SYS: SITE; which is not part of any particular release hierarchy. For example, create the directory >sys>site>, and copy the files from SYS: SITE; into this directory, and edit SYS.TRANSLATIONS there. If you do this, you can still boot Release 5 worlds, but you cannot run the namespace server in Release 5, or do other things which depend on reading information from the site directory, such as **make-system** of an unknown system.

In the example below, note that the attribute line has been modified to have a Syntax: ZetaLisp; and a Base: 10; attribute. If this is not done, a minor warning message results when configuring the Release 6 world.

Old:

```
;; -*- Mode: LISP; Package: FILE-SYSTEM; -*-
(set-logical-pathname-host "SYS"
 :physical-host "ACME-YUKON"
 :translations '(("FONTS; TV;" ">Rel-5>tv-fonts>")
                 ("FONTS; LGP-1;" ">Rel-5>lgp-1-fonts>")
                 ("LMFS-PATCH;" ">Rel-5>lmfs>patch>")
                 ("*;" ">Rel-5>*>")
                ))
```

New:

```
;; -*- Mode: LISP; Package: FILE-SYSTEM; Syntax: ZetaLisp; Base: 10; -*-
(select (or (si:get-release-version) (si:get-system-version))
 ;; Translations for Release 5
 (5 (set-logical-pathname-host "SYS"
 :physical-host "ACME-YUKON"
 :translations '(("FONTS; TV;" ">Rel-5>tv-fonts>")
                 ("FONTS; LGP-1;" ">Rel-5>lgp-1-fonts>")
                 ("LMFS-PATCH;" ">Rel-5>lmfs>patch>")
                 ("*;" ">Rel-5>*>")
                )))
 ;; Translations for Release 6
 (6 (set-logical-pathname-host "SYS"
 :translations '(("SYS:**;*.*.*" "ACME-YUKON:>Rel-6>**>*.*.*)" )))
 (otherwise (ferror "~D unknown release or system version.~
 Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
 (or (si:get-release-version) (si:get-system-version))))))
```

The Release 6 version of the translations is made much simpler by the use of the "wild-inferiors" wildcard, which allows matching to any level of directory. This translation is adequate for whenever all the files are kept on a single host. All directories and subdirectories on SYS: are mapped to subdirectories of ACME-YUKON:>Rel-6>. However, if for space considerations, you decided to put the documentation files on ACME-VMS, some of the larger source directories on ACME-UNIX, and the rest on ACME-YUKON, this example would look like this:

```

;;; -*- Mode: LISP; Package: FILE-SYSTEM; Syntax: ZetaLisp; Base: 10; -*-
(select (or (si:get-release-version) (si:get-system-version))
  ;; Translations for Release 5
  (5 (set-logical-pathname-host "SYS"
    :physical-host "ACME-YUKON"
    :translations '((("FONTS; TV;" ">Rel-5>tv-fonts>")
      ("FONTS; LGP-1;" ">Rel-5>lgp-1-fonts>")
      ("LMFS-PATCH;" ">Rel-5>lmfs>patch>")
      ("*;;" ">Rel-5>*>")
      )))
  ;; Translations for Release 6
  (6 (set-logical-pathname-host "SYS"
    :translations
    '((("SYS:DEBUGGER;*.*.*" "ACME-UNIX://usr//rel6//debugger//*.*.*")
      ("SYS:IO;*.*.*" "ACME-UNIX://usr//rel6//io//*.*.*")
      ("SYS:IO1;*.*.*" "ACME-UNIX://usr//rel6//io1//*.*.*")
      ;; LMTAPE; has subdirectories
      ("SYS:LMTAPE;*.*.*" "ACME-UNIX://usr//rel6//lmtape//*.*.*")
      ("SYS:LMTAPE;*.*.*" "ACME-UNIX://usr//rel6//lmtape//*/*.*.*")
      ("SYS:LMTAPE;*.*.*" "ACME-UNIX://usr//rel6//lmtape//*/*/*.*.*")
      ("SYS:NETWORK;*.*.*" "ACME-UNIX://usr//rel6//network//*.*.*")
      ("SYS:SYS;*.*.*" "ACME-UNIX://usr//rel6//sys//*.*.*")
      ("SYS:SYS2;*.*.*" "ACME-UNIX://usr//rel6//sys2//*.*.*")
      ("SYS:WINDOW;*.*.*" "ACME-UNIX://usr//rel6//window//*.*.*")
      ("SYS:ZWEI;*.*.*" "ACME-UNIX://usr//rel6//zwei//*.*.*")
      ;; All documentation goes on ACME-VMS
      ("SYS:DOC;**;*.*.*" "ACME-VMS:DISK$USER:[REL6.DOC...]*.*.*")
      ;; Examples go on ACME-VMS, too.
      ("SYS:EXAMPLES;**;*.*.*" "ACME-VMS:DISK$USER:[REL6.DOC...]*.*.*")
      ("SYS:**;*.*.*" "ACME-YUKON:>Rel-6>**>*.*.*"))))
  (otherwise (ferror "~D unknown release or system version.~
    Edit SYS:SITE;SYS.TRANSLATIONS to add this system."
    (or (si:get-release-version) (si:get-system-version))))))

```

Note that because `SYS: LMTAPE;` has subdirectories, and because Unix does not support "wild-inferiors" multiple-directory-level wildcards, there is an entry for each additional level of directory. If this were not done, these directories would have been placed on ACME-YUKON, because the last entry in the translation table would be the only match. The following directories include subdirectories:

```

sys: cl;
sys: doc;
sys: examples;
sys: fonts;
sys: hardcopy;
sys: lmfs;
sys: lmtape;

```

```
sys: mailer;  
sys: patch;  
sys: print;  
sys: sage;  
sys: zmail;
```

Some of the subdirectories can in turn contain an additional level of subdirectory. The system software does not currently use more than three levels of logical directory. Thus we listed all three of the following in the example above.

```
sys: lmtape;  
sys: lmtape; *;  
sys: lmtape; *; *;
```

Once you have completed the editing of SYS: SITE; SYS.TRANSLATIONS and written it out, there is no need to update the Release 5 worlds, as the information for them has not been changed. Release 6 now gets the new information automatically when you configure it for your site.

For more information on logical hosts: See the section "Logical Pathnames" in *Reference Guide to Streams, Files, and I/O*.

21.2 Back up the Files on Your Machine

It is important that you back up any local file system(s) on the Lisp Machines to be upgraded before the conversion starts. For information on backing up your file system: See the section "Performing Dumps", page 6.

To verify the tape, click on [Local FS Maint] in the File System Editor, which pops up a menu. Click on [Reload/Retrieve], which pops up another menu. Click on [Compare] to compare the files on tape to the files stored on the machine.

21.3 Booting a World

To boot a world load, type the following to the FEP:

```
Fep>clear machine  
Fep>load microcode microcode-file-name  
Fep>load world world-load-file-name  
Fep>set chaos-address this-machine's-chaos-address  
Fep>Start
```

If you encounter any errors with a particular command, you should try that command again. To avoid typing all those commands every time you boot a world, you can use a file named >Boot.boot. This file contains text similar to the above

example. The `Boot` command activates the file; for example, the command `Boot >boot.boot` tells the FEP to execute the commands in the file `>Boot.boot`.

21.4 Loading System Sources and Documentation and Ephemeral-object Garbage Collector Diagnostic

Symbolics distributes the ephemeral-object garbage collector diagnostic, the documentation and the examples, as well as the 5.3 patches, on the documentation tape. The system sources are on the source tape. The ephemeral-object garbage collector must be loaded, but loading the documentation and examples is optional.

To load the ephemeral-object garbage collector diagnostic, the documentation, and the examples, do the following:

- Insert the documentation tape into the cartridge tape drive.
- Use **dis:load-distribution-tape**, which pops up a menu showing the systems on the tape.
- Click on `Skip` for the 5.3 patches at the beginning of the file.
- Click on `Load` for `EGC-DIAG`; this is the ephemeral-object garbage collector diagnostic.
- Click on `[Load]` for the following to load examples and the graphic-lisp system:
 - "Examples-rel-6-0"
 - "zmail-examples"
 - "Graphic-Lisp"

These are loaded into the `sys: examples; directory`. Please refer to `sys:examples;-readme-.text` for instructions on how to run the examples and set up the graphic-lisp system.

If you do not want to load these, Click on `[Skip]`.

- To load the documentation, click on `[Load]` for the following:
 - "DOC"
 - "Documentation-fonts"
 - "distributed-utilities"

These are loaded into the sys: doc; directory. See the section "Release 6 Documentation and Examples Tape Checklist", page 130.

- It is important to note that Release 6.0 has a new on-line Document Examiner, and to use this, the documentation must be loaded. See the section "Using the Online Documentation System" in *User's Guide to Symbolics Computers*. If you decide you need more space on the documentation file server, you can remove parts of the documentation. See the section "Pruning the Documentation Database", page 136.
- Symbolics distributes the system sources on one distribution tape. New release sources can only be loaded from a machine running new release software because of the current handling of logical pathnames. To load the sources, do the following:
 - Insert the documentation into the cartridge tape drive.
 - Use **dis:load-distribution-tape**, which pops up a menu showing the systems on the tape.
 - Click on [Load] to load sources; click on [Skip] to not load the sources.
 - Click on [Do It] when completed.

The sources are then loaded into the sys: site; directory. See the section "Release 6 Sources Tape Checklist", page 128.

21.5 Pruning the Documentation Database

After the documentation is installed at your site, you have the option of removing documents from the documentation database. To remove documents from the documentation database, use the function **sage:prune-online-document-set**. This invokes a menu of major sections in the database. To remove a document from the database, click on [Remove]. This causes the files containing that documentation to be deleted from the documentation database file server. In this way, space can be saved on the file server by pruning sections not needed at your site.

See the section "Topics Pruned From the Documentation Database" in *User's Guide to Symbolics Computers*.

21.6 Running the Ephemeral-object Garbage Collector Diagnostic

To run this diagnostic, do the following:

- If the diagnostic has not yet been loaded from the documentation tape, use **dis:load-distribution-tape** to load this system. The name of the system is EGC-DIAG.
- Copy the diagnostic from the SYS host to your local machine, using the following form:

```
(si:install-microcode "tmc5-egc-diag-mic.mic")
```

When you are asked about updating your boot file, answer NO.

- Halt your machine, using the Halt Machine command.
- Type the following commands to the FEP:

```
Fep>Clear Machine
Fep>Load Microcode FEP:>TMC5-EGC-DIAG-MIC.MIC
Fep>Load World FEP:> world-load-file-name
Fep>Start
```

world-load-file-name is the name of the world load. The diagnostic can be run with either Release 5.3 or Release 6.0.

- The diagnostic will print out any errors it discovers, and then, finally, will report the total number of errors in the total test. The final line printed is the total.
 - If this number is 0, it indicates that no errors were found in the hardware being tested and the ephemeral-object garbage collector can be used on this machine.
 - If this number is not zero, you should call your local Customer Service Engineer and have the DP and FEP boards checked and possibly replaced. In the meantime, it is important that you do not run the ephemeral-object garbage collector. The ephemeral-object garbage collector is disabled by default.
 - In some cases, the diagnostic finds errors that make it impossible for the diagnostic to continue. When this happens, Lisp halts itself and displays the message "Lisp halted itself" and you are returned to the FEP. Use the Show Status command:

```
FEP>show status
```

Several lines of information are displayed; within this text will be statements phrases about GC parity errors.

This indicates that you have hardware problems. You should call your local Customer Service Engineer and have the DP and FEP boards checked and possibly replaced. In the meantime, it is important that you do not run the ephemeral-object garbage collector.

21.7 Using Disk Restore

The Disk Restore command restores the file system or files from cartridge tapes to disk. This command is used to install a new release world, that is, a microcode file and world load file.

The Disk Restore command displays three questions:

1. *Have you done a Set Disk-type for each disk unit that does not have valid label block on it?*

The disk type must be known before the first reference to it, in case the label block is not yet written.

Answer Yes to this question.

2. *Is there a microcode at the beginning that should be skipped?*

This is asking about a possible stream-format microcode at the beginning of the tape. Answer No to this question.

3. *Do you want to restore it?*

This question is asked for each file that is on the tape. Answering No skips the current tape restore section and searches for the next one. Answering Yes restores the file to disk.

21.7.1 Using Disk Restore During Installation

Newly distributed Symbolics release tapes are written in file restore format. Data recorded in file restore format must be a file. The header, which is displayed after the question about skipping microcode at the beginning, includes the name of the source file, its length, its creation date, its author, and any comments supplied by the author of the tape. You are asked for a destination pathname (FEP directory) for the data; the default disk unit is assumed unless another is specified. Both a file system and the specified destination file must already exist on the disk unit, and the destination file must be large enough to hold the data on tape.

Large files (for example, worlds not compacted by garbage collection) cross tape boundaries. But since all block numbers are relative to the beginning of the file, the

second reel of tape is logically continuous with the first, and file restoration proceeds as for single-reel files.

Note: The first machine on which you install a new release must have its own tape drive. Check that the correctly labelled tape is in this machine's tape drive.

21.7.2 Release 6.0 Microcode Types

Note: for a 3600 with IO Rev 6 (IO6), use the corresponding 3670 microcode.

3600	TMC5-MIC.MIC.319
3600, FPA	TMC5-FPA-MIC.MIC.319
3600, IFU	IFU-MIC.MIC.319
3600, IFU, FPA	IFU-FPA-MIC.MIC.319
3640	TMC5-IO4-ST506-MIC.MIC.319
3640, FPA	TMC5-IO4-ST506-FPA-MIC.MIC.319
3640, IFU	IFU-IO4-ST506-MIC.MIC.319
3640, IFU, FPA	IFU-IO4-ST506-FPA-MIC.MIC.319
3670	TMC5-IO4-MIC.MIC.319
3670, FPA	TMC5-IO4-FPA-MIC.MIC.319
3670, IFU	IFU-IO4-MIC.MIC.319
3670, IFU, FPA	IFU-IO4-FPA-MIC.MIC.319

21.8 Set Site Command

Set Site *site name*

Starts a dialogue to set the current site to be *site name*. This command is used to configure the software and identify your machine before you use a new world load. It should be the first thing you type to your machine after booting the new software.

When a new world is booted for the first time, the herald gives the machine name as *DIS-LOCAL-HOST*. You are prompted in the course of the Set Site dialogue for a name for the machine.

You need the following information to use the Set Site command:

Site name	What you call the location of your machines. This might be the name of your company, or, if you are more whimsically minded, it might be related to the machine names you have chosen. In the sample dialogues, we have chosen the site name Downunder.
-----------	---

Name of the local host

The name of the lisp machine you are configuring. See the section "Why Do You Name Machines and Printers?" in *User's Guide to Symbolics Computers*. In the sample dialogues, we have chosen machine names Koala and Kangaroo.

Name of the namespace server

The name of the machine where the namespace database is stored.

Chaosnet address of the namespace server

The octal number which identifies the location of the namespace server on the network. You can use Show Host *machine name* or (**hostat "machine name"**) to find this number.

If you are configuring a new site, you also need:

SYS host The machine where the sources are to be stored.

Host for bug reports

The machine to which bug reports are to be sent.

SYS:SITE; directory translation

The physical pathname that SYS:SITE; translates to on the SYS host. See the section "What is a Logical Pathname?" in *User's Guide to Symbolics Computers*. In the sample dialogues this is

```
koala:>sys>site>
```

System account The user-id that the system uses when a server logs into a machine. In the sample dialogues we have chosen Wombat.

21.8.1 Configuring a Namespace Server At an Established Site

Use this dialogue if you are installing new software on the machine at your site that has been the namespace server. What you type is underlined. RETURN means press the key labeled return.

Command: Set Site (a site name) downunder

Define a new site named DOWNUNDER (as opposed to looking for an existing definition of DOWNUNDER on disk)? (Y or N) No

You answer no because your site is already defined.

What host is the namespace server for DOWNUNDER (default: local): RETURN
 Where is the descriptor file for DOWNUNDER (default
 local:>sys>site>downunder-namespace.text) : RETURN
 Assuming that KOALA is the real pathname host name of the local host.
 Warning: the local host, DIS-LOCAL-HOST, was not known to have that
 name.
 [13:15:16 Namespace lock DIS-LOCAL-HOST: Reloading namespace DOWNUNDER.
 Recent servers contacted are DIS-LOCAL-HOST.]
 The local host is now KOALA.
 Command:

You are now ready to use Save World to save the configured Lisp world. You can then use Copy World to distribute the new world to the other machines at your site.

21.8.2 Configuring a Namespace Server At a New Site Where the Sys Host is a Lisp Machine

Use this dialogue if you are installing your site's first Symbolics 3600-family computer. What you type is underlined. RETURN means press the key labeled return.

Command: Set Site (a site name) downunder
 Define a new site named DOWNUNDER (as opposed to looking for an existing
 definition of DOWNUNDER on disk)? (Y or N) Yes

You answer yes because your site is being defined.

What host is the namespace server for DOWNUNDER (default: local): RETURN

The machine you are installing the software on (the *local host*) is the namespace server.

What host is to be the SYS host for DOWNUNDER (default: local): RETURN
 What host is to be used for bug reports for DOWNUNDER? (default: local): RETURN
 What is the real name of the local host? Koala

Here is where you name your machine.

What directory on KOALA corresponds to SYS:SITE; (default: >sys>site) Koala:>sys>site>
 What account should be used for the system to login to Koala? (default: lisp) wombat
 What is the local timezone (without daylight savings time) (default: EST) RETURN
 Is DOWNUNDER a standalone site (there are no servers to respond to a
 Who-am-I request)? (Y or N) no
 Should changes to the database be recorded so as to allow incremental
 updating (only of help to sites with a very large number of hosts) (Y or N) Yes
 [13:15:16 Namespace lock DIS-LOCAL-HOST: Reloading namespace DOWNUNDER.
 Recent servers contacted are DIS-LOCAL-HOST.]
 The local host is now Koala.
 Command:

You are now ready to use Save World to save the new Lisp world. Then you can edit your namespace database to enter the names of the other machines at your site and use Copy World to move the new world to them. See the section "Updating the Namespace Database" in *User's Guide to Symbolics Computers*.

21.8.3 Configuring a Non-namespace Host At an Established Site

Use this dialogue if you are installing new software on a machine at your site that is not the namespace server. What you type is underlined. RETURN means press the key labeled return.

Command: Set Site (a site name) downunder

Define a new site named DOWNUNDER (as opposed to looking for an existing definition of DOWNUNDER on disk)? (Y or N) No

You answer no because your site is already defined.

What host is the namespace server for DOWNUNDER (default: local): Koala

Chaosnet address for KOALA: 401

Host responds as KOALA, ok? (Y or N) Yes.

The local host is now KANGAROO

Command:

You are now ready to use Save World to save the configured Lisp world and reboot the machine.

21.9 Lgp Example

If you have the Fonts Widths File entry in the namespace object pointing to SYS: PRESS-FONTS; you must change that entry because this directory no longer exists on the tape being distributed. The file in question lives in SYS: FONTS; LGP-1; (since it is specific to the LGP-1). We recommend that you change the namespace entry for the LGP to point to

SYS: FONTS; LGP-1; FONTS.WIDTHS

and if you are keeping a separate Release 5.2 hierarchy with separate translations, copy the file from the Release 5 SYS:PRESS-FONTS; to the Release 5

SYS:FONTS;LGP-1;

21.10 Distributing the New World and Microcode to Other Machines

To distribute the new Release 6.0 world and the accompanying microcode, use the Copy World Command to copy the world to the specified machine. In the following example, the world Rel6.load.1 would be copied to the host Puffin.

Copy World Rel6.load.1 Puffin

For more information: See the section "Copy World Command" in *User's Guide to Symbolics Computers*.

To distribute the new microcode, use the **si:install-microcode** function on the machine that will receive the new microcode.

(si:install-microcode 319.)

If you have a machine that is running Release 5, use **si:receive-band** to copy the world from another machine to your machine. See the function **si:receive-band**, page 50.

21.11 Software Installation Terminology

This section describes some terms from the viewpoint of a site upgrading to the newest release and notes some general restrictions.

namespace server Every site has associated with it a *namespace*. A namespace is a context in which names of users, hosts, and printers are resolved into their representative objects. A *namespace server* maintains these mappings of names into objects. These mappings are stored in files that make up the *namespace database*. A namespace server also processes queries and updates about the mappings and the objects to which those mappings refer.

A namespace has a single *primary namespace server*. This server is responsible for maintaining the files that make up the namespace database, and for controlling the access of users and secondary namespace servers to this information.

A namespace can have one or more *secondary namespace servers*. The secondary namespace servers answer queries when their information is up-to-date and can ask the primary server for up-to-date information. Secondary namespace servers are useful only for large sites (more than 25 machines).

All hosts chosen as namespace servers must be Symbolics machines.

namespace database

The namespace database consists of a number of files, containing information about objects of various classes. You can create, modify, or delete objects in the namespace database by using the namespace editor. To run the namespace editor, use [Namespace] in the System menu, the Edit Namespace Object command, or the **tv:edit-namespace-object** function with no arguments. These invoke a window-based editor for the database. The namespace database usually resides in files in the sys: site; logical directory.

The default location for its namespace database is on the SYS host. The default name for the file that contains the names of the other files making up the database (the *namespace descriptor file*) would be as follows, assuming that SITENAME is the registered name of the site:

sys:site;SITENAME-namespace.text

Other files constituting the database would have names such as:

sys:site;SITENAME-namespace-log.text
 sys:site;SITENAME-namespace-changes.text
 sys:site;SITENAME-hosts.text
 sys:site;SITENAME-users.text
 sys:site;SITENAME-others.text

The Edit Namespace Object command or the **tv:edit-namespace-object** function can be run on any host at the site. To update a database, both the primary namespace server and the host on which the namespace database resides must be available.

objects and classes The namespace database contains *objects* of several different *classes*. The set of classes in the database is as follows:

namespace
 site
 network
 host
 printer
 user

For example, suppose you have a user CHARLEY logged in to 3600-family machine TUNA at site FISHERY.

CHARLEY is an object of class user.
 TUNA is an object of class host.
 FISHERY is an object of class site.
 FISHERY is an object of class namespace.

namespace object Namespace objects are named by an identifier that is qualified by the namespace name:

FISHERY|TUNA

TUNA happens to be an object of class host and in the FISHERY namespace.

standalone site A site is said to be *standalone* if it has no way of accessing a namespace server at boot time. The most common standalone site is a single Symbolics machine that is not connected to any network. However, because only Symbolics machines can provide namespace service, a site is also considered standalone if it consists of one Symbolics machine and any number of non-Symbolics-machine hosts. An organization with several Symbolics machines, none of which is on a network, constitutes several independent standalone sites; that is, each machine appears as the single Symbolics machine host at a standalone site.

who-am-i The *who-am-i* request is part of the procedure Symbolics machines use at boot time to determine if the saved Lisp environment matches the current site. If the information returned in response to the *who-am-i* request differs from the information that was saved away in the booted world load, the machine asks the user to confirm the machine's identity. If the user replies negatively, the machine disassociates itself from the old site and waits for the user to adjust it to the new site.

In Release 6.0, the Symbolics machine always broadcasts a *who-am-i* request over the local network at boot time. Any other machine receiving this request would then return a response containing the just-booted machine's identity.

dis:load-distribution-tape

Function

Invokes the distribution loader. It is used to load files from a distribution tape into places specified by logical pathnames, creating directories as needed. Invoking **dis:load-distribution-tape** pops up a menu that shows all the files that are on the tape. You then have a choice of loading or skipping each of the files. The tape should be inserted in the tape drive before invoking the function.

To use **dis:load-distribution-tape**, the machine must have access to a tape drive; it can either be on the local host or on another machine on the network. If the files are to be loaded onto another machine, that other machine must use the same SYS host as the machine with the tape drive.

Distribution tapes are used to distribute transportable systems and libraries, defined by system declarations and pseudosystem declarations, from one site to another.

21.12 Release 5 and Release 6 Directory Information

This information contains additional information about the Release 5 directories and the Release 6 directories.

21.12.1 Description of Release 5 Directories

Release 5 can be divided into the following parts. We suggest you consider directories for deletion by group, in the order given here.

Sources

The Release 5 system sources are unlikely to be needed for reference if no new code is being developed for Release 5. Also, Release 5 demo software is included in this category.

sys: cc;
sys: cube;
sys: debugger;
sys: demo;
sys: distribution;
sys: io;
sys: iol;
sys: l-bin;
sys: l-compiler;
sys: lmfs;
sys: lmsym;
sys: network;
sys: sys;
sys: sys2;
sys: window;
sys: ubin;
sys: ucadr;
sys: zwei;

Documentation This is the online documentation distributed with Release 5, in a format suitable for **viewf** or reading into an editor buffer. In Release 6, a completely new set of online documentation is distributed, in a format suitable for reading with the Document Examiner. The Release 5 online documentation offers little advantage over the printed documentation, so it is a good candidate for deletion. This is especially true if doing so makes room for the Release 6 online documentation, which is much more useful.

sys:doc;

Examples This category includes the examples distributed with Release 5. It will only be needed for Release 5 development. A new examples directory is included with Release 6.

sys: examples;

Other This category includes directories which are needed for backup for critical things distributed with Release 5. These don't take much space, but you can delete them if you are careful not to delete the copies from your FEP filesystems.

sys: l-fep;
sys: l-ucode;

Patch directories

This category includes directories which contain patch files. All

except for `sys: patch`; also contain sources, but you should not delete the patches in these directories until all Release 5 worlds either are no longer needed, or have been upgraded to 5.3 and disk-saved.

sys: hardcopy;
sys: lmfs-patch;
sys: lmtape;
sys: patch;
sys: zmail;

Run-time files In this category are the directories with files needed at run-time by people using Release 6. For example, printer fonts are needed as long as the machine with the print spooler is running Release 5. Some programs may need fonts from the `SYS: FONTS; TV;` directory. You should consider the needs of users at your site in considering whether to delete these directories.

sys: fonts; tv;
sys: fonts; lgp-1;
sys: press-fonts;

Layered products

Included in this category are any layered products and the print system.

sys: print;
<layered product directories>

Site directory The site directory should *not* be deleted. It contains many files which are needed by Release 6, including namespace files, translation files, and system location files.

sys: site;

21.12.2 Release 6 Sys Host Size

Tables 5 and 6 provide additional information about the sizes of individual directories in Release 6.

Both may be used as an aid in splitting the SYS host across multiple physical hosts.

To convert to units of VMS or Unix blocks, multiply LMFS record sizes by 8. The result usually slightly overstates the disk space required.

Table 6 can be used as a guide to how much space can be saved by pruning the documentation database.

Directory

LMFS Records

sys: **;	6652
sys: cl; **;	5
sys: debugger; **;	94
sys: dialnet; **;	28
sys: doc; **;	2060
sys: examples; **;	150
sys: fonts; **;	991
sys: fonts; lgp-1; **;	636
sys: fonts; nlgpl; **;	21
sys: fonts; tv; **;	331
sys: hardcopy; **;	13
sys: io; **;	309
sys: iol; **;	89
sys: l-fep; **;	36
sys: l-ucode; **;	1240
sys: lmfs; **;	33
sys: lmsym; **;	31
sys: lmtape; **;	80
sys: mailer; **;	53
sys: network; **;	234
sys: patch; **;	77
sys: print; **;	38
sys: sage; **;	7
sys: sys; **;	102
sys: sys2; **;	256
sys: window; **;	315
sys: zmail; **;	21
sys: zwei; **;	361

Table 5. Size of Release 6 Directories in LMFS records

Directory	LMFS Records
sys: doc; **;	2060
sys: doc; arr; **;	59

sys: doc; audio; **;	23
sys: doc; cl; **;	36
sys: doc; comp; **;	20
sys: doc; cond; **;	67
sys: doc; conv; **;	6
sys: doc; cp; **;	45
sys: doc; debug; **;	32
sys: doc; defs; **;	27
sys: doc; eval; **;	26
sys: doc; fed; **;	26
sys: doc; file; **;	70
sys: doc; flav; **;	50
sys: doc; flow; **;	42
sys: doc; fsed; **;	7
sys: doc; func; **;	33
sys: doc; hard; **;	7
sys: doc; ined; **;	8
sys: doc; init; **;	5
sys: doc; int; **;	61
sys: doc; intstr; **;	11
sys: doc; lms; **;	14
sys: doc; mac; **;	24
sys: doc; maint; **;	49
sys: doc; menus; **;	68

Table 6. Size of documentation directories in LMFS records

sys: doc; miscf; **;	14
sys: doc; misct; **;	4
sys: doc; miscu; **;	5
sys: doc; miscui; **;	29
sys: doc; netio; **;	84
sys: doc; nota; **;	11
sys: doc; patch; **;	14
sys: doc; pkg; **;	55
sys: doc; prim; **;	110
sys: doc; proc; **;	18

sys: doc; prot; **;	118
sys: doc; rn6; **;	51
sys: doc; sage; **;	15
sys: doc; scroll; **;	10
sys: doc; site; **;	35
sys: doc; stor; **;	29
sys: doc; str; **;	83
sys: doc; tape; **;	21
sys: doc; tools; **;	102
sys: doc; user; **;	29
sys: doc; windoc; **;	122
sys: doc; zmacs; **;	157
sys: doc; zmailc; **;	37
sys: doc; zmailt; **;	39

Table 6, concluded

21.12.3 More Information About Release 6 Directories

Here is more information about the Release 6 Directories, broken down by type of use and showing the systems and directories which support that usage.

Note that some directories appear in more than one category. The sizes are all in LMFS records and appear here for convenience only. When a directory appears in more than one category, the record count is the total for all categories of use, not for the individual category.

- Legal notices.

Documentation Font Notice Latest files.

Trade Secret Notice Latest files.

sys: site; *Size varies.*

sys: fonts; nlgp1; 21 records.

- Microcode and FEP binaries.

Microcode Auxiliaries Rel 6 0 Latest files.

Egc Diag Latest files.

Fep Debug Tools Latest files.

sys: l-ucode; 1240 records.

sys: l-fep; 36 records.

- Additional screen fonts not preloaded. These should always be available; some software will load from this directory automatically.

Tv Fonts 6 0 Latest files.

sys: fonts; tv; 331 records.

- The on-line documentation database. Highly recommended for quick on-line lookup of Release 6 documentation. See the section "Pruning the Documentation Database", page 136.

*Documentation Database 3 System build objects, patches 3.1 to 3.0.
Documentation Fonts Latest files.*

sys: doc; 2060 records.

sys: fonts; nlGPL; 21 records.

- Examples. These are useful examples of programming various parts of the system, including the mouse, the window system, etc.

Examples Rel 6 0 Latest files.

Graphic Lisp 7 Latest sources, system build objects, patches 7.1 to 7.0.

Zmail Examples Latest files.

sys: examples; 150 records.

sys: zmail; 21 records.

- System sources. This is a selected set of useful sources for perusal with the `m-` command.

System 271 Latest sources, patches 271.1 to 271.0.

Lmfs Tools 6 0 Latest files.

Tape 41 Latest sources, patches 41.1 to 41.0.

sys: debugger; 94 records.

sys: io; 309 records.

sys: iol; 89 records.

sys: lmsym; 31 records.

sys: lmtape; 80 records.

sys: network; 234 records.

sys: sys; 102 records.

sys: sys2; 256 records.

sys: window; 315 records.

sys: zwei; 361 records.

- Patch directories. These need to be available when applying patch releases. Also, if the patch directories are not loaded, the Load Patches command will print a warning. This might be a nuisance when loading patches for an application system, unless the system is explicitly named in the Load Patches command.

System 271 Latest sources, patches 271.1 to 271.0.
COMMON-LISP 4 Patches 4.1 to 4.0, PATCHES ONLY.
Zmail 107 Patches 107.1 to 107.0, PATCHES ONLY.
Hardcopy 46 Patches 46.1 to 46.0, PATCHES ONLY.
LMFS 57 Patches 57.1 to 57.0, PATCHES ONLY.
Tape 41 Latest sources, patches 41.1 to 41.0.

sys: patch;	77 records.
sys: cl;	5 records.
sys: zmail;	21 records.
sys: hardcopy	13 records.
sys: lmfs;	33 records.
sys: lmtape;	80 records.

- Printer software. Provides spooling and LGP support. **sys: print;** is needed when loading the print spooler or applying patch releases, but not during operation, if a world is saved with the print spooler on the spooling host. The font directories are needed during operation.

Print 38 Latest sources, system build objects, patches 38.1 to 38.0.
Hardcopy Fonts 6 0 Latest files.
Documentation Font Notice Latest files.
Documentation Fonts Latest files.

sys: print;	38 records.
sys: fonts; lgp-1;	636 records.
sys: fonts; nlgpl;	21 records.

- Store and forward mailer. Provides mailing lists, forwarding, dialnet. **sys: mailer;** is needed when loading the mailer or applying patch releases, but not during operation, if a world is saved with the mailer on the mail server host.

Mailer 42 System build objects, patches 42.1 to 42.3.
Dialnet Registry 6 0 Latest files.

sys: mailer;	53 records.
sys: site;	Size varies.

- Patch Release 5.3. This is not a part of Release 6, but is needed for Release 5.

System 242 Patches 242.354 to 242.358, PATCHES ONLY.

Zmail 83 Patches 83.60 to 83.61, PATCHES ONLY.

Hardcopy 20 Patches 20.9 to 20.10, PATCHES ONLY.

Various Release 5 directories

21.13 Field Software Support Offices

Contact your local Field Software Support Office and ask for Software Support if you need assistance while installation Symbolics software on your Symbolics machine.

Field Software Support Offices

Symbolics Austin
Austin, Tx.
(512) 346-4954

Symbolics Cambridge
Cambridge, Mass.
(617) 577-7600

Symbolics Los Angeles
Los Angeles, Ca.
(213) 473-6583

Symbolics Palo Alto
Palo Alto, Ca.
(415) 494-8081

Symbolics Washington, D.C.
Falls Church, Va.
(703) 790-0303

Index

A

Tape

UNIX
Installing an
Installing an
Registering an
Syntax and Base file
Namespace
Namespace
Namespace
Namespace

A

Access 17
Adding a New Machine to an Existing Site 93
Adding a Partition to LMFS 35
Adding patches to Lisp image 46
Allocating Extra Paging Space 44
ASCII print server 67
ASCII Print Server on UNIX 67
ASCII Print Server on VMS 68
ASCII Print Server on VMS 76
attributes 131
Attributes for a New Lisp Host 79
Attributes for a New UNIX Host 80
Attributes for a New VMS Host 80
Attributes for New Hosts 79

A

B

Finding
Verifying a

Syntax and

The Dumper Choice
Free

B

Backup Copies of Files 12
Backup dump tape directory 10
backup tape 10
Back up the Files on Your Machine 134
Bands 16
Base file attributes 131
Binary files 31
Booting a World 59, 134
Boxes 21
buffer 37

B

C

tape:
tape:
tape:

The

The
The

Installation
si:
The Dumper
Database
Loading System Sources
Running the Ephemeral-object Garbage
SELECT F
Set Site
tv:edit-namespace-object
Edit Namespace Object
File System Maintenance
si:

C

carry-dump function 31
carry-list function 34
carry-load function 32
Carry-tape System 30
Carry-tape Dumper 30
Carry-tape file groups 32
Carry-tape Lister 34
Carry-tape Loader 32
CC symbol table 54
chaos:my-address variable 83
Character files 31
Checklist 106
checksum-band function 52
Choice Boxes 21
classes 143
and Documentation and Ephemeral-object Garbage
Collector Diagnostic 135
Collector Diagnostic 137
command 35
Command 139
command 143
Commands 39
compare-band function 53
Conditionalizing the translation file 131
Configuring a Namespace Server At an Established
Site 140

C

Installing and Listing of tape
 Distribution Finding Backup
tape:
 Save
 3640:

Configuring a Namespace Server At a New Site
 Where the Sys Host is a Lisp Machine 141
 Configuring a Non-namespace Host At an Established Site 142
 Configuring Large Sites for Multiple Mail Servers 64
 Configuring the Mailer 61
 contents 18
 Copier 15
 Copier 18
 Copies of Files 12
copy-tape function 18
 Copying distribution tape 18
 Copying Worlds to Machines with Large Disks 93
 Copying Worlds to Machines with Small Disks 94
 Creating mailing lists 61
 Creating More Room on the Local Disk 43
 current Lisp world 46
 Customizing and Saving the World 55

D**D****D**

Pruning the Documentation
 Hardware/microcode
dis:
si:
 Namespace
 Loading System Sources and Documentation and Ephemeral-object Garbage Collector Diagnostic 135
 Running the Ephemeral-object Garbage Collector Diagnostic 137
 Description of Release 5 Directories 145
 >dump-maps directory 6
 >repatriations directory 38
 Backup dump tape directory 10
 Sys:distribution; directory 18, 19
 Tape directory file 6
 Release 5 and Release 6 Directory Information 145
 dis:dfppseudo-system special form 16, 25
 dis:load-distribution-tape function 145
 Creating More Room on the Local Disk 43
 Installing Release 6.0 on a 3640 with One 140-mbyte Disk 103
 disk-save function 46
 disk-saves 48
 Using Disk Restore 138
 Using Disk Restore During Installation 138
 Configuring Preinstalled Worlds on Machines with Large Disks 95
 Configuring Preinstalled Worlds on Machines with Small Disks 96
 Disks 93
 Disks 94
 Disk Space on the 3640 55
 distributing FEP files 15
 Distributing the New World and Microcode to Other Machines 142
 Distributing transportable systems and libraries 15
 Distribution Copier 18
 Distribution Dumper 19
 Distribution Loader 17

Sys:
 Copying
 Description of
 Release 6.0
 Loading System Sources and
 Pruning the
 Tape
 Registering a Tape
 Selecting system to be
 Distribution
 Giving the Go-ahead to the
 The Carry-tape
 The
 The
 3600 FEP files selected for
 Problems During
 Systems selected for
 Performing
 Backup
 Distribution Mapper 18
 distribution; directory 18, 19
 distribution tape 18
 Distribution Tapes 16
 Distribution Tapes 101
 Distribution Tape Subsystem 15
 Documentation and Ephemeral-object Garbage
 Collector Diagnostic 135
 Documentation Database 136
 Drive ID 6
 Drive in the Namespace 72
 Dump map 6
 dumped 20
 Dumper 15, 25
 Dumper 19
 Dumper 23
 Dumper 30
 Dumper Choice Boxes 21
 Dumper Systems List 20
 dumping 20
 Dumping 23
 dumping 20
 Dumping, Reloading, and Retrieving 5
 Dumping individual files 15
 Dumping World Loads 20
 Dumps 6
 dump tape directory 10

E

Loading System Sources and Documentation and
 Running the
 Lgp
 Sys:
 Tape Spec Merging

E

Edit Namespace Object 73
edit-namespace-object Edit Namespace Object
 command 143
 Ephemeral-object Garbage Collector Diagnostic 135
 Ephemeral-object Garbage Collector Diagnostic 137
 Example 142
 examples 135
 Examples 30

E**F**

SELECT
 Transporting and distributing
 3600
 >fspt.fspt
 Conditionalizing the translation
 Lmfs.file
 Namespace descriptor
 Tape directory
 Updating the Sys Translations
 Syntax and Base
 Carry-tape
 Binary
 Character
 Dumping individual
 FEP
 Finding Backup Copies of
 Repatriating
 Transporting and distributing FEP
 Transferring source
 Snapshot
 Back up the

F

F command 35
 FEP files 16
 FEP files 15
 FEP files selected for dumping 20
 Field Software Support Offices 153
 file 35
 file 131
 file 35
 file 143
 file 6
 File 131
 file attributes 131
 file groups 32
 files 31
 files 31
 files 15
 files 16
 Files 12
 files 38
 files 15
 files between hosts 15
 files on a local LMFS 15
 Files on Your Machine 134

F

3600 FEP	files selected for dumping 20
Write	files to tape 31
Manipulating the local	file system 39
	File System Maintenance Commands 39
	File System Maintenance Program 37
	File system partition table 35
System	file types 20
lmfs:	find-backup-copies function 12
	Finding Backup Copies of Files 12
	Forwarding tables 64
	Free record map 37
	Free Records: File System Maintenance Program 37
	FSPT 35
dis:load-distribution-tape	function 145
disk-save	function 46
lmfs:find-backup-copies	function 12
load-and-save-patches	function 46
net:set-site	function 83
print-herald	function 85
print-disk-label	function 45
si:checksum-band	function 52
si:compare-band	function 53
si:describe-system-versions	function 85
si:get-system-version	function 85
si:install-microcode	function 49
si:login-to-sys-host	function 83
si:print-login-history	function 48
si:print-system-status-warning	function 85
si:receive-band	function 50
si:transmit-band	function 51
tape:carry-dump	function 31
tape:carry-list	function 34
tape:carry-load	function 32
tape:copy-tape	function 18
tape:write-fep-files-to-tape	function 20
Herald	Functions and Variables 85
Site Related	Functions and Variables 83

G**G****G**

	Loading System Sources and Documentation and Ephemeral-object
	Garbage Collector Diagnostic 135
Running the Ephemeral-object	Garbage Collector Diagnostic 137
si:	get-system-version function 85
	Graphic-lisp system 135
Carry-tape file	groups 32

H**H****H**

Registering Users and	Hardware/microcode debugger 54
Namespace Attributes for a New LispM	Hardware in the Site's Namespace 71
Namespace Attributes for a New UNIX	Herald Functions and Variables 85
Namespace Attributes for a New VMS	Host 79
Preparing the Sys	Host 80
Configuring a Non-namespace	Host 80
	Host 121
Configuring a Namespace	Host At an Established Site 142
	Configuring a Namespace Server At a New Site Where the Sys
	Host is a Lisp Machine 141
Namespace Attributes for New	Hosts 79
Registering	Hosts 72
Splitting Logical Hosts Across Physical	Hosts 131
Transferring source files between	hosts 15
Splitting Logical	Hosts Across Physical Hosts 131

Release 6 Sys Host Size 147

<p> Tape Drive Adding patches to Lisp Dumping si: Mailer Introduction to the Software Software Expanded Release 6.0 Related Software Procedures for Expanded Release 6.0 Installation </p>	<p> ID 6 image 46 individual files 15 install-microcode function 49 installation 61 Installation Checklist 106 Installation Guide 89 Installation Guide 87 Installation Instructions 108 Installation Procedures 121 Installation Terminology 143 Installing microcode 49 Installing Printers 67 Installing an ASCII Print Server on UNIX 67 Installing an ASCII Print Server on VMS 68 Installing and Configuring the Mailer 61 Installing New Microcode 54 Installing New Software 49 Installing Release 5.3 99 Installing Release 6.0 101 Installing Release 6.0 104 Installing Release 6.0 on a 3640 with One 140-mbtype Disk 103 Instructions 108 Instructions for Managing Disk Space on the 3640 55 </p>
--	---

<p> Manipulating the Distributing transportable systems and Adding patches to Namespace Attributes for a New Save current Systems The Dumper Systems The Carry-tape Creating mailing Adding a Partition to Snapshot files on a local World dis: Distribution The Carry-tape Dumping World Saving New Versions of World Creating More Room on the Manipulating the Snapshot files on a Splitting </p>	<p> Label 45 Lgp Example 142 libraries 15 Lisp image 46 Lispm Host 79 Lisp world 46 List 20, 21 List 20 Lister 34 Listing of tape contents 18 lists 61 LMFS 35 LMFS 15 Lmfs.file file 35 lmfs:find-backup-copies function 12 LMFS Multiple Partitions 35 load-and-save-patches function 46 load disk-saves 48 load-distribution-tape function 145 Loader 15, 25 Loader 17 Loader 32 Loading System Sources and Documentation and Ephemeral-object Garbage Collector Diagnostic 135 Loads 20 Loads 45 Local Disk 43 local file system 39 local LMFS 15 Logical Hosts Across Physical Hosts 131 </p>
---	---

si: **login-history** variable 48
 si: **login-to-sys-host** function 83
 Lost records 37

M

Source
 Distributing the New World and Microcode to Other
 Target
 Configuring Preinstalled Worlds on
 Copying Worlds to
 Configuring Preinstalled Worlds on
 Copying Worlds to
 Adding a New
 Installing and Configuring the
 Creating
 Configuring Large Sites for Multiple
 File System
 File System
 Free Records: File System
 Instructions for
 Dump
 Free record
 Distribution
 Tape Spec
 Installing
 Installing New
 Distributing the New World and
 Release 6.0
 Creating
 Configuring Large Sites for
 LMFS
chaos:

M

machine 49
 Machines 142
 machines 49
 Machines with Large Disks 95
 Machines with Large Disks 93
 Machines with Small Disks 96
 Machines with Small Disks 94
 Machine to an Existing Site 93
 Mailer 61
 Mailer installation 61
 mailing lists 61
 Mail Servers 64
 Maintenance Commands 39
 Maintenance Program 37
 Maintenance Program 37
 Managing Disk Space on the 3640 55
 Manipulating the Label 45
 Manipulating the local file system 39
 map 6
 map 37
 Mapper 15
 Mapper 18
 Merging Examples 30
 microcode 49
 Microcode 54
 Microcode to Other Machines 142
 Microcode Types 139
 More Room on the Local Disk 43
 Multiple Mail Servers 64
 Multiple Partitions 35
my-address variable 83

M**N**

Registering a Tape Drive in the
 Registering Users and Hardware in the Site's
 Edit
tv:edit-namespace-object Edit
 Configuring a
 Configuring a

N

Namespace 72
 Namespace 71
 Namespace server 143
 Namespace Attributes for a New Lispm Host 79
 Namespace Attributes for a New UNIX Host 80
 Namespace Attributes for a New VMS Host 80
 Namespace Attributes for New Hosts 79
 Namespace descriptor file 143
 Namespace Object 73
 Namespace Object command 143
 Namespace Server At an Established Site 140
 Namespace Server At a New Site Where the Sys
 Host is a Lisp Machine 141
net:set-site function 83
 Newest sources 20

N

O

Edit Namespace
 Database
 Field Software Support
 Installing Release 6.0 on a 3640 with
 Writing systems
 Introduction to Site
 Site

O

Object 73
 Objects 20
 objects 143
 Offices 153
 One 140-mbtype Disk 103
 onto tape 19
 Operations 3
 Operations 1
 Orphans 38

O

P

Allocating Extra
 Tape Spec
 LMFS Multiple
 File system
 Adding a
 Adding
 Splitting Logical Hosts Across
 System Source
 Configuring
 Configuring
 Release 6.0
 Registering a
 Installing
 si:
 UNIX ASCII
 Installing an ASCII
 Installing an ASCII
 Registering an ASCII
 si:
 Related Installation
 File System Maintenance
 Free Records: File System Maintenance
 Reloader
 Tape Spec
 Defining a

P

Paging Space 44
 Parameters 26
 Partitions 16
 Partitions 35
 partition table 35
 Partition to LMFS 35
 Patches 20
 patches to Lisp image 46
 Performing Dumps 6
 Physical Hosts 131
 Policy 17
 Preinstalled Worlds on Machines with Large
 Disks 95
 Preinstalled Worlds on Machines with Small
 Disks 96
 Preparing the Sys Host 121
 Prerequisites 103
print-herald function 85
print-disk-label function 45
 Printer 73
 Printers 67
print-login-history function 48
 print server 67
 Print Server on UNIX 67
 Print Server on VMS 68
 Print Server on VMS 76
print-system-status-warning function 85
 Problems During Dumping 23
 Procedures 121
 Procedures for Installing Release 6.0 104
 Program 37
 Program 37
 program 10
 Prompting 25
 Pruning the Documentation Database 136
 Pseudosystem 25
 Pseudosystems 16

P

R

si:
 Free
 Lost
 Free

R

receive-band function 50
 record map 37
 records 37
 Records: File System Maintenance Program 37
 Records in use 37
 Reel tape spec parameter 26

R

Registering Hosts 72
 Registering Users 71
 Registering an ASCII Print Server on VMS 76
 Registering a Printer 73
 Registering a Tape Drive in the Namespace 72
 Registering Users and Hardware in the Site's
 Namespace 71
 Delete Unneeded Parts of
 Installing
 Release 5 131
 Release 5.3 99
 Release 5 and Release 6 Directory Information 145
 Description of
 Installing
 Release 5 Directories 145
 Release 6.0 101
 Procedures for Installing
 Release 6.0 104
 Release 6.0 Distribution Tapes 101
 Introduction to
 Expanded
 Release 6.0 Installation 101
 Release 6.0 Installation Instructions 108
 Release 6.0 Microcode Types 139
 Installing
 Release 6.0 on a 3640 with One 140-mbyte Disk 103
 Release 6.0 Prerequisites 103
 Release 5 and
 Release 6 Directory Information 145
 Release 6 Sys Host Size 147
 Reloader program 10
 Dumping,
 Reloading, and Retrieving 5
 Reloading and Retrieving 10
 Repatriating files 38
who-am-I
 request 143
 Using Disk
 Restore 138
 Using Disk
 Restore During Installation 138
 Dumping, Reloading, and
 Retrieving 5
 Reloading and
 Retrieving 10
 Creating More
 Room on the Local Disk 43

S

S
 Salvager 37, 38
 Save World 73
 Save current Lisp world 46
 Saving New Versions of World Loads 45
 Saving Subsequent Worlds 57
 Saving the World 55
 SELECT F command 35
 server 143
 server 67
 Server At an Established Site 140
 Server At a New Site Where the Sys Host is a Lisp
 Machine 141
 Server on UNIX 67
 Server on VMS 68
 Server on VMS 76
 Servers 64
set-site function 83
set-logical-pathname-host during installation 131
 Set Site Command 139
 Setting up a New Site 91
si:checksum-band function 52
si:compare-band function 53
si:describe-system-versions function 85
si:get-system-version function 85
si:install-microcode function 49
si:login-history variable 48
si:login-to-sys-host function 83
si:print-login-history function 48
si:print-system-status-warning function 85
si:receive-band function 50
si:site-initialization-list variable 83

3640:
 3640: Customizing and
 Namespace
 UNIX ASCII print
 Configuring a Namespace
 Configuring a Namespace
 Installing an ASCII Print
 Installing an ASCII Print
 Registering an ASCII Print
 Configuring Large Sites for Multiple Mail
net:
 Using **fs:**

S

File	System Maintenance Program	37
Free Records: File	System Maintenance Program	37
File	system partition table	35
	Systems	16
	Systems List	20, 21
Distributing transportable	systems and libraries	15
The Dumper	Systems List	20
Writing	systems onto tape	19
	System Source Policy	17
Loading	System Sources and Documentation and Ephemeral- object Garbage Collector Diagnostic	135
	Systems selected for dumping	20
Selecting	system to be dumped	20
Updating the	Sys Translations File	131

T

	CC symbol	table	54
File system partition		table	35
Forwarding		tables	64
Copying distribution		tape	18
Verifying a backup		tape	10
Write files to		tape	31
Writing systems onto		tape	19
		Tape Access	17
		Tape Specs	25
		tape:carry-dump function	31
		tape:carry-list function	34
		tape:carry-load function	32
		tape:copy-tape function	18
		tape:write-fep-files-to-tape function	20
		(tape:write-fep-files-to-tape)	20
		tape contents	18
		tape directory	10
		Tape directory file	6
		Tape Drive ID	6
		Tape Drive in the Namespace	72
		Tapes	16
		Tapes	101
		Tape Spec Merging Examples	30
		Tape Spec Parameters	26
		Tape Spec Prompting	25
		Tape Spec Syntax	28
		Tape Subsystem	15
		Tape System	15
		Target machines	49
		Terminology	143
		Transferring source files between hosts	15
		translation file	131
		Translations File	131
		transmit-band function	51
		transportable systems and libraries	15
		Transporting and distributing FEP files	15
		tv:edit-namespace-object Edit Namespace Object command	143
Release 6.0 Microcode		Types	139
System file		types	20

T

T

U

Installing an ASCII Print Server on
 Namespace Attributes for a New
 Delete
 Setting
 Back
 Registering
 Registering

U

UNIX 67
 UNIX ASCII print server 67
 UNIX Host 80
 Unneeded Parts of Release 5 131
 up a New Site 91
 Updating Software 46
 Updating the Sys Translations File 131
 up the Files on Your Machine 134
 Users 71
 Users and Hardware In the Site's Namespace 71

U

V

chaos:my-address
sl:login-history
sl:site-initialization-list
sl:system-additional-info
 Herald Functions and
 Site Related Functions and
 Saving New
 Installing an ASCII Print Server on
 Registering an ASCII Print Server on
 Namespace Attributes for a New

V

variable 83
 variable 48
 variable 83
 variable 85
 Variables 85
 Variables 83
 Verifying a backup tape 10
 Versions of World Loads 45
 VMS 68
 VMS 76
 VMS Host 80

V

W

3640: Customizing and Saving the
 Booting a
 Save
 Save current Lisp
 Distributing the New
 Dumping
 Saving New Versions of
 3640: Saving Subsequent
 Configuring Preinstalled
 Configuring Preinstalled
 Copying
 Copying
tape:
 (tape:

W

who-am-I request 143
 World 55
 World 59, 134
 World 73
 world 46
 World and Microcode to Other Machines 142
 World load disk-saves 48
 World Loads 20
 World Loads 45
 Worlds 57
 Worlds on Machines with Large Disks 95
 Worlds on Machines with Small Disks 96
 Worlds to Machines with Large Disks 93
 Worlds to Machines with Small Disks 94
write-fep-files-to-tape function 20
 write-fep-files-to-tape) 20
 Write files to tape 31
 Writing systems onto tape 19

W

